Department of Adaptive Systems

ÚTIA, AV ČR

P.O.Box 18, 182 08 Prague 8,Czech Republic

RESEARCH REPORT

M. Kárný, J. Böhm, T. V. Guy, L. Jirsa, I. Nagy, P. Nedoma, L. Tesař, M. Tichý

Probabilistic Advisory System

theory, algorithms and software

Version 1.5 May 26, 2003

GAČR 102/03/0049, AVČR S1075351

ÚTIA AV ČR, P. O. Box 18, 182 08 Prague, Czech Republic
http://utia.cas.cz, Fax: (+420) 26605 2068, E-mail: school@utia.cas.cz

Contents

Chapter 1

Introduction

This work summarizes theoretical, algorithmic and software basis that has roots in the project of European Commission within the programme Future and Emerging Technologies IST-1999-12058, ProDaCTool and it is further developed and modified within projects GAČR 102/03/0049, ACČR S1075351. The text has arisen as a common communication basis of the research team. It may, however, serve to experts dealing with applied decision making or those searching both for ready particular solutions as well as for open and challenging problems.

The research was motivated by the urgent need for industrial producers to minimize the down time caused by process malfunction and improper maintenance. The potential of the elaborated methodology is much broader. The developed probabilistic mining of information hidden within acquired data can be used in other applications (medical, economical etc). This possibility should be kept in mind when interpreting the used terminology as we stick to terms usual in maintenance of technological processes.

1.1 Operator control

Automation, which is reliable and effective in a production line, can greatly improve the quality of production as well as decrease production costs. This is because not only automation addresses the ever increasingly complex problems met in production, but also possibility to exploit the greater power of contemporary computers. In spite of this, the optimal global solutions are often not feasible. The complexity of real-life problems calls for a hierarchical solution. The lowest level is typically low dimensional, but counteract highly dynamical phenomena. Often, the solutions of the low-dimensional problems are used repeatedly. Thus, it is worthwhile to spend a substantial amount of energy on a careful modelling and control (decision-making) design. At the upper levels of a hierarchical solution,

- the problems being addressed become multi-variate,
- (almost) static relationships dominate,
- detailed modelling is too expensive to be performed.

The project aims to provide computer support at the higher level, where supervision is carried out by an operator whose performance greatly influences the quality of resulting products. Indeed, the attained quality is strongly dependent on the operator's experience accumulated during years of practice as well as on his specific personal performance. Unfortunately, the operator is often left without guidelines on how to treat complex situations faced during operating. A support is therefore needed not only to warn the operator against potential undesirable modes of the controlled process, but also to direct operations to the desirable state.

In addition, the operators may well be knowledgeable about the controlled processes, but this may not necessarily apply when the processes are formulated in a mathematical language. Thus, operator support has to be constructed in a comprehensible manner. Therefore a dominant part of the information flow towards the operator needs to be simply, mostly graphically, expressed.

1.2 State of operator support motivating the research

Here, we outline the state of the art in statistical process control as it looked at the beginning of the ProDaCTool project. It served us as a departure point but during a few years of the solution it has become obsolete. For this reason, a special Chapter 15 is included reflecting the better current situation. Moreover, as one of concluding chapters of this text, it allows us to compare our advisory system with its competitors in a clear way.

Attempts towards meeting industrial technological needs have ranged from simple expert systems based on acquiring and storing the knowledge and experiences of the operators [1] to a more systematic approach such as the mining of process data [2]. With the construction of expert systems often experiencing bottleneck problems [3], the alternative methods of data mining become more attractive.

In the industrial sector, *statistical process control* (*SPC*) is widely used whereby control charts such as $\overline{X}$, Range and Cumsum charts are independently computed for each individual process variable. However, the reliability of these charts have been criticized as dependent upon the sample size on which computation is based [4]. Furthermore, on a screen where these charts are simultaneously displayed, any undesirable deviation in the variable trends is mostly detected through visual comparisons between the variables. Clearly this approach ignores any existing "correlation" between variables. The correlations are complex and important primary factors for understanding the managed process. Practical attempts being made towards multi-variate analysis show at most two-dimensional "static" scatter-grams. In a communication with an SPC software manufacturer, it was revealed that multi-variable SPC analysis is yet to be realized: the manufacturers still have to be persuaded to believe that it can work reliably and with a benefit.

1.3 Dynamic cluster analysis

In this project, *probabilistic cluster analysis* is exploited for data mining and operator's guiding. The developed clustering attempts to capture *dynamic properties* of a good operating. This makes it rather difficult and specific.

Statically, each multi-variate data record contains a combination of sensor readings that have been taken at the same time instant, describing thus their static relationships. Given identical conditions of machine running, similar readings of the sensors can be expected. By collecting such informative data over a period of time, clusters of these data points can be expected to exist. Their descriptive power increases substantially if at least short-term history of is clustered together with respective data records, if dynamic clusters are searched for.

Often, both static and dynamic clusters can be assumed to emerge from a mixture normal multivariate distribution. This observation has led to the study of normal mixtures [5] and probability density estimation [6] [7], to name but a few. The established method of clustering differ widely across the diverse applications that utilize it, both in interpretation terms and algorithm design [8, 9, 10].

The probabilistic interpretation of clusters, we employ, provides a rich set of formal and algorithmic tools that help us to keep the overall solution internally consistent. It supports the considered final technical aim. The existence of conditioning as the basic tool for guiding a human being through a high-dimensional data space would be sufficient reason for our preference to this type problem formalization and solution. We can moreover rely on a well established methodology that covers in a consistent way both learning and design parts of the constructed advisory system.

Orientation towards dynamic modelling and optimized guiding are dominant features that make our approach unique while keeping a significant overlap with achievements of our predecessors and competitors, see Chapter 15.

1.4 Layout and style

The project has ambitions to create a *generic software tool for guiding operators*. This determines layout and style of this text.

First, the text fills the chosen Bayesian framework by all elements needed for the final implementation and use of the advising system. This is done in preparatory **Chapters 2, 3** and **4**. They respectively motivate and describe the underlying Bayesian paradigm as well as related established learning and design techniques.

Chapter 5 provides a specific problem formulation with extensions necessary for applicability of the Bayesian decision making.

Practical feasibility of the proposed solution, i.e. way out of "curse of dimensionality", is the main driving force determining the rest of the text. General techniques describing learning part of the advisory system form the content of **Chapter 6. Chapters 8** and **10** are its images for normal and Markov chain mixtures, respectively.

General solution of the design part of the advisory system is presented in **Chapter 7**. Its normal and Markov chain counterparts are in **Chapters 9** and **11**.

The above basis is complemented by several Chapters dealing with the following specific aspects. **Chapter 12** presents a solution of the vital problem of learning initialization. At the same time, it improves substantially so called *mean tracking (MT)* algorithm [11] that was at the birth of the project. **Chapter 13** complements information on treatment of mixed mixtures that blend model and data of different types. **Chapter 14** describes overall structure of the adopted software solution. It serves as a bridge to the software documentation [12] and justifies various adopted restriction on the used tool. It also refers briefly on parallel implementation of the learning part of the advisory system. **Chapter 15** provides updated review of the state of the art in the addressed area as well as basic information on potential competitors of our advisory system. It also informs on applications we are working on. Among others, they confirm that the created tool is indeed of a generic nature. **Chapter 16** summarizes status of the project achieved and names some of problems to be addressed in future.

Style of the presentation is driven by the communication purpose of the text. It is structured so that developers of particular algorithms and their implementations find all necessary information in a small sub-part of this extensive text. It motivated mathematical-text-like structuring and repetition of sub-parts.

Moreover, real implementation requires solution of all minor steps including defaults on various optional parameters. It makes us to care about them too in spite of the fact they make presentation a bit unbalanced. The care about details helps us also to decrease the amount of manually tuned knobs. This aspect is often overlooked in academic solutions of complex problems and often inhibits a practical use of, otherwise sophisticated, solutions. Simply, it can hardly be expected that some user will be able and willing to tune tens or more such knobs.

Last but not least, the presented solution has many directions and aspects that remain to be solved, inspected or improved. The most important ones are explicitly formulated throughout the text as open problems. Consequently, they will not be forgotten and moreover they may serve as a basis of many smaller (typically postgraduate) research projects.

1.5 Acknowledgements

This text report results put together within three-years of intensive work. Without firm background, we could build upon it would be impossible. Thus, it is fair to name our most important predecessors first in this short and incomplete paragraph. We were definitely deeply influenced by

Václav Peterka, Alena Halousková and Rudolf Kulhavý.

Naturally, any complex project relies on external support of various institutions and persons who are directly or indirectly involved. We want confirm that we are aware that without such support the described research would be impossible or its outcomes would be much weaker.

Supporting institutions

European Commission sponsored the project ProDaCTool IST-99-12058.

GA ČR sponsored the projects RECiAS 102/99/1564 and further development within the project GAČR 102/03/0049.

AV ČR supported applications of the advisory system by the grant ACČR S1075351.

ÚTIA AV ČR, University of Reading, Trinity College Dublin provided institutional support to the academic ProDaCTool partners and ÚTIA AV ČR supports the further development.

Compureg Plzeň s.r.o., Kovohutě Rokycany a.s., Department of Nuclear Medicine, FNM, Eltodo, a.s. participated on the ProDaCTool project as application partners. They invested their skills, energy, money and served us as strong practical-feedback providers.

People

EC scientific officer and reviewers Pekka Karp, Francois Arlabosse, Kauko Leiviskä, Colin Piddington, Åsbjorn Rolstadas gave us strong and useful feedback during the solution of ProDaCTool.

ProDaCTool team members (except authors) who contributed informally to this text are

University of Reading Kevin Warwick, Dean Parry, Matthew Maisey, Rak Pattel, Kate Darby, Matthew Taylor

Trinity College Dublin Anthony Quinn, Václav Šmídl

ÚTIA AV ČR Josef Andrýsek, Jan Kracík, Bohumil Kovář, Markéta Valečková

Compureg Plzeň, s.r.o. František Jirkovský, Pavel Ettler, Ivan Puchr

Kovohutě Rokycany, a.s. Václav Nejdl, František Šulc

Department of Nuclear Medicine, University Hospital Motol Jindřiška Heřmanská, Nenad Medič, Petr Vlček, Jaroslav Zimák

ProDaCTool project co-founders Elmawatti L. Sutanto and Jiří Kadlec who started the reported research with us.

Experts that helped us in specific problems are Jindřich Bůcha, Jiří Grim, Dani Juričič, Andrej Rakar, Jan Šindelář, Igor Vajda.

Technical support was made by Jiří Fidler, Jaroslava Hradcová, Roman Kytka.

Many thanks to everyone ...

Chapter 2

Underlying theory

At a general level, the presented decision-making theory should help the decision maker – typically, human being – to select one of the available options (decisions). These options relate to a part of the real world, to a system. They concern with descriptions of the system and/or its influencing.

This chapter summarizes the design principles and tools exploited later on in our particular task, i.e. operator support. It starts with conventions and notions used throughout, Sections 2.1, 2.2. The considered framework covers a broad range of problems. Inevitably, the adopted symbols and notions have more specific meanings in specific application fields. The reader is kindly asked to be patient especially in this respect.

The adopted principle of the optimal decision-making under uncertainty, inspected in Section 2.3, implies the following important conclusion: incomplete knowledge and randomness have the same operational consequences for decision-making and should be treated in the same way labelled as *Bayesian decision making*. In the same section, the design of optimal decision rules is solved. In Section 2.4 the design of the optimal strategies is derived. These designs work with models that are usually obtained through Bayesian learning described in Section 2.5.

A similarly formulated control design, that is a specific decision-making problem, is presented in [13]. For a detailed explanation of Bayesian learning see [14]. This chapter is a tailored version of a chapter in [15] where our view on decision making is summarized.

2.1 General conventions

The conventions presented here are mostly respected in this work. If some exception is necessary it is named just at the place of its validity. If some verbal notions are introduced within bodies of Propositions, Remarks etc., then they are *emphasized by the print* that differs from that of the surrounding text. Moreover, they appear in Index. Sometimes, important parts of sentences are stressed by underlining them.

f *is the letter reserved for probability (density) functions (p(d)f).*

The meaning of the p(d)f is given through the name of its argument.

x^* *denotes the range of* x, $x \in x^*$.

$\mathring{x}$ *denotes the number of members in the countable set* x^* or *number of entries in the vector* x.

$\equiv$ *means the equality by definition.*

x_t *is a quantity* x *at the discrete time labelled by* $t \in t^* \equiv \{1, \ldots, \mathring{t}\}$.

$\mathring{t} \leq \infty$ *is called (decision, learning, prediction, control, advising) horizon.*

$x_{i;t}$ *is an* i*-th entry of the array* x *at time* t.

The semicolon in the subscript indicates that the symbol following is time index.

$x(k\_l)$ *denotes the sequence* x_t *with* t *between time moments* $k \leq l$, i.e. $x(k\_l) \equiv x_k, \ldots, x_l$.

$x(t) \equiv x(1\_t)$.

$x(t)$ *is an empty sequence and reflects just the prior information if* $t < 1$.

$\mathrm{supp}\,[\,f(x)]$ *is the support of the pdf* $f : x^* \to [0, \infty]$, *i.e. the subset of* x^* *on which* $f(x) > 0$.

2.2 Basic notions and notations

Below, if need be, a brief characterization of the introduced notion is complemented by explanatory comments (printed differently).

Quantity *is a mapping with a numerical range, i.e. a subset of the multi-variate, real-valued space.*

The domain and form of the quantity are mostly unused and unspecified. The introduced notion corresponds with *random variable* used in probability theory. The use of the alternative term should stress that probability is not our primary aim but the tool we adopted for our main goal, i.e. decision making under uncertainty. Moreover, we stress our orientation on numerical values that arise mostly by measuring physical quantities.

Realization *is a value of the quantity for its fixed argument.*

Often, the quantity and its realization are not distinguished, as usual. The proper meaning is determined by the context.

System *is the part of the world that is of interest for a decision maker who should either describe or influence it.*

Decision maker might be a person, group of persons or mechanisms.

The system is specified with respect to the aim that the decision maker wants to reach and with respect to the tools he/she has available. In other words, the boundaries (that can be penetrated) of the system are implied by the decision task.

Decision $a \in a^*$ *is the value of a quantity that can be directly chosen by the decision maker for reaching his/her aims.*

A decision task makes sense only when the decision maker faces real option, i.e. when $\mathring{a} > 1$.

(Decision) experience $\mathcal{P}_a \in \mathcal{P}_a^*$ *is knowledge about the system available to the decision maker for the selecting the decision* $a \in a^*$.

For example, if just data values D are available for constructing the estimate $\hat{\Theta}$ of an unknown quantity $\Theta \in \Theta^*$ then the experience is $\mathcal{P}_{\hat{\Theta}} \equiv D$. Often, experience includes the past data observed.

Note that the symbol $\mathcal{P}_{a^*}$ would be more adequate than $\mathcal{P}_a$. Its use would, however, made the notation even more cumbersome.

(Decision) ignorance $\mathcal{F}_a \in \mathcal{F}_a^*$ *is knowledge about the system unavailable to the decision maker for the choice of the value of* a.

The unknown value of the quantity Θ to be estimated is a part of the ignorance $\mathcal{F}_{\hat{\Theta}}$ of its estimate $\hat{\Theta}$. Often, ignorance contains future, still unobserved data.

(System) behavior $\mathcal{Q}^*$ *consists of all possible realizations (of trajectories)* $\mathcal{Q}$, *i.e. values of all quantities within the time span determined by the horizon of interest that are related to the system and considered by the decision maker.*

The realization $\mathcal{Q}$ can be split with respect to any decision $a \in a^*$ into the relevant experience $\mathcal{P}_a$ and ignorance $\mathcal{F}_a$. Formally, $\mathcal{Q} = (\mathcal{P}_a, a, \mathcal{F}_a)$.

Note that a single realization $\mathcal{Q}$ can be split differently with respect to a pair of decisions, say a, $\tilde{a}$ with different experience $\mathcal{P}_a \neq \mathcal{P}_{\tilde{a}}$ and different ignorance $\mathcal{F}_a \neq \mathcal{F}_{\tilde{a}}$. $\mathcal{Q} = (\mathcal{P}_a, a, \mathcal{F}_a) = (\mathcal{P}_{\tilde{a}}, \tilde{a}, \mathcal{F}_{\tilde{a}})$.

(System) input $u \in u^*$ *is a decision, which is supposed to influence the ignorance part* $\mathcal{F}_u$ *of the (system) behavior.*

A manipulated valve position influencing a fluid flow is the system input. A point estimate $\hat{\Theta}$ of an unknown (realization of) quantity Θ is an instance of the decision that is not the system input. It is chosen with the aim to describe the system and exhibits no direct influence on it.

(System) output $y \in y^*$ *is an observable quantity that provides the decision maker information about the (system) behavior.*

A pressure measured in an isolated heated system is an instance of the output. A pressure applied on a closed system is an instance of the input.

To be or not to be input or output is a relative property.

Innovation $\Delta_{a_t} \in \Delta^*_{a_t}$ *contains quantities that are not included in the experience* $\mathcal{P}_{a_t}$ *but are included in* $\mathcal{P}_{a_{t+1}} \setminus a_t$.

Often, $\Delta_{a_t} = y_t =$ the system output at time t.

Decision rule $\mathcal{R} : \mathcal{Q}^* \to a^*$ *is a mapping that assigns a decision* $a \in a^*$ *to the behavior* $\mathcal{Q} \in \mathcal{Q}^*$.

Causal decision rule $\mathcal{R} : \mathcal{P}^*_a \to a^*$ *is a mapping that assigns the decision* $a \in a^*$ *to its experience* $\mathcal{P}_a \in \mathcal{P}^*_a$.

In other words, the decision a made by a causal decision rule is uninfluenced by the related ignorance $\mathcal{F}_a$.

Mostly, we deal with the causal decision rules so that the term "causal" is often dropped.

Estimator is an instance of the causal decision rule $\mathcal{R} : \mathcal{P}^*_{\hat{\Theta}} \to \hat{\Theta}^*$ that assigns an estimate $\hat{\Theta}$ of the unknown quantity $\Theta \in \Theta^*$ to the available experience $\mathcal{P}_{\hat{\Theta}}$.

Strategy *is a sequence of decision rules* $\{\mathcal{R}_t : \mathcal{Q}^* \to a^*_t\}_{t \in t^*}$.

Causal strategy $\{\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a^*_t\}_{t \in t^*}$ *is a sequence made of causal decision rules.*

Again, we deal mostly with causal strategies so that the term "causal" is often dropped.

Controller *is a causal strategy assigning inputs* u_t *to experience* $\mathcal{P}_{u_t}$, $\forall t \in t^*$.

The proportional controller given by the constant p is an example of the causal control strategy $\{y^*_{t-1} \to u^*_t : u_t = p y_{t-1}\}_{t \in t^*}$.

Design *selects the decision rule or strategy.*

The design selecting a single rule is called *static design.* The choice of the strategy is called *dynamic design.*

The person (group) who makes the selection is *designer.* Authors and readers of this text are supposed to be designers. In that sense, the term *we* used within the text should mostly be read: *we designers.*

Agreement 2.2.1 (Interface between the design and reality) *Physical connections of the design domain to the real world (sensors, transmission lines, actuators etc.) are here always taken as a part of the system. Consequently, all quantities and mappings considered can be and are taken as mathematical entities living in an abstract calculating machine.*

Uncertain behavior *(related to static design) is a behavior whose realizations* $\mathcal{Q}$ *can be decomposed into*

- $\mathcal{Q}_{\mathcal{R}} \equiv$ *the part that is unambiguously determined by the considered decision rule* $\mathcal{R} \in \mathcal{R}^*$,
- uncertainty $\Upsilon \equiv$ *the part of the behavior that belongs to the ignorance* $\mathcal{F}_{\mathcal{R}(\mathcal{P})}$ *of decisions* $\mathcal{R}(\mathcal{P})$ *generated by the admissible rules* $\mathcal{R} \in \mathcal{R}^*$ *and uninfluenced by them, even indirectly.*

With an abuse of notation, we write the corresponding decomposition of realization $\mathcal{Q} = (\mathcal{Q}_{\mathcal{R}}, \Upsilon) = (\cdot, uncertainty)$.

Uncertain behavior related to dynamic design is encountered if any of its rules faces uncertainty.

Incomplete knowledge of (the realization of) a considered quantity $\Theta \in \Theta^*$ makes the behavior uncertain.

External unobserved noise influencing the system makes its behavior uncertain.

Uncertainty expresses both incomplete knowledge and randomness.

Uncertain behavior in the dynamic design is defined in a similar way by replacing rules with strategies. The related cumbersome notation is avoided in our presentation.

Decision-making *means design and application of a decision rule (strategy).*

Admissible strategy *is a strategy* $\{\mathcal{R}_t\}_{t\in t^*}$ *that*

- is *causal* $\Leftrightarrow$ $\{\mathcal{R}_t\}_{t\in t^*} \equiv \{\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a^*_t\}_{t\in t^*}$ *and*
- meets *technological restrictions* $\Leftrightarrow$ *the ranges of its decision rules are in pre-specified subsets of respective decision sets.*

Loss function, $\mathcal{Z} : \mathcal{Q}^* \to [0, \infty]$, *quantifies the degree of achievement of the design aim.*

The loss function measures quality of the realizations $\mathcal{Q}$ and indirectly (partially) orders admissible strategies influencing the behavior.

The smaller the value of $\mathcal{Z}(\mathcal{Q})$ the better.

Negative value of profit might be a loss function in economic applications.

Quadratic deviation $(\hat{\Theta} - \Theta)'(\hat{\Theta} - \Theta)$ ($'$ is transposition) of the estimate $\hat{\Theta}$ and of an unknown vector Θ is a wide spread loss function used for ranking decision rules generating point estimates.

The important case of multi-valued loss function [16] is out of the scope of this text but the approach discussed in Section 2.3 could be relatively simply extended to it by embedding the index of the "decisive" entry of the loss function into uncertainty.

"Expected" loss $\tilde{\mathcal{E}}(\mathcal{Z}) \equiv \tilde{\mathcal{E}}_{\{\mathcal{R}\}}(\mathcal{Z})$ *assigns to the considered loss function* $\mathcal{Z}$ *and strategy* $\{\mathcal{R}\}$ *a value in* $[0, \infty]$ *that is independent of the uncertainty.*

The quotation marks as well as the sign ˜ serve us in the discussion that shows that, under widely acceptable conditions, we have to deal with expectation in mathematical sense. Then, they are unused any more.

Optimal design *selects an admissible strategy that leads to the smallest value of the "expected" loss function.*

Practically admissible strategy *is an admissible strategy that respects restrictions limiting complexity of the decision-making.*

The complexity is considered with respect to the computational resources available at the design and application stages. The majority of discussed problems in which the complexity restriction play a role are computationally hard in terms of computer sciences. An intuitive understanding of the computational complexity is sufficient to our purposes.

Practically optimal design *selects a practically admissible strategy giving the smallest values of the "expected" loss.*

The presented optimal design provides optimal admissible strategies and can be simply adapted to provide strategies of a pre-specified complexity by optimizing over a set of simple decision rules, for instance, over proportional controllers only.

Operational formal tools for practically optimal design are not available. It is not known how to make the optimal design of a pre-specified complexity.

We never know whether the selection of the constant determining proportional controller made with use of, say, ten algebraic operations is really the best one possible among all selections that are allowed to perform ten algebraic operations.

This is the *main barrier of the applicability* of the theory describing the optimal design. The optimal design becomes a practical tool by employing sound engineering heuristics. The practical optimum is not guaranteed. This fact is stressed by using the term *suboptimal design*.

2.3 Decision-making under uncertainty

Here, we describe a general way how to understand and face uncertainty that causes incomplete ordering of strategies. In order to avoid a cumbersome notation, we formulate the adopted design principle, related requirements and their consequences for the static design, i.e. the design of a single not necessarily causal decision rule. The obtained results are also valid for the dynamic design, i.e. the choice of decision strategies.

Agreement 2.3.1 (Uncertainty in decision-making) *Decision-making under uncertainty arises if the optimal decision-making is to be performed and*

- *at least a pair of different decisions can be made,* $\mathring{a} > 1$*,*
- *the considered loss function* $\mathcal{Z}(\mathcal{Q}) \equiv \mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon)$ *depends on non-void set* Υ^* *of uncertainties.*

Under uncertainty, the function $\mathcal{Z}_{\mathcal{R}}(\Upsilon) \equiv \mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon)$ *of uncertainty* Υ*, i.e. the part of realization in ignorance uninfluenced by the rule* $\mathcal{R}$*, is assigned to each considered decision rule* $\mathcal{R} \in \mathcal{R}^*$*. The set of such functions is denoted* $\mathcal{Z}_{\mathcal{R}^*}$

$$\mathcal{Z}_{\mathcal{R}^*} \equiv \{\mathcal{Z}_{\mathcal{R}} : \Upsilon^* \to [0, \infty],\ \mathcal{Z}_{\mathcal{R}}(\Upsilon) \equiv \mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon)\}_{\mathcal{R} \in \mathcal{R}^*}. \tag{2.1}$$

Under uncertainty, the loss function is insufficient for a *complete ordering* (comparing) of admissible rules, and consequently strategies, in spite of the fact that its values are in the fully ordered set. Each $\mathcal{R}$ is assigned a function in the set (2.1) and not a single number.

For instance, consider a pair of estimates $\hat{\Theta}_1 \neq \hat{\Theta}_2$ of an unknown scalar quantity $\Theta \in \Theta^* \equiv (-\infty, \infty)$. They cannot be compared with the help of the quadratic loss function $(\Theta - \hat{\Theta})^2$ as we do not know whether Θ is in that part of Θ^* where $(\Theta - \hat{\Theta}_1)^2 \leq (\Theta - \hat{\Theta}_2)^2$ or in its complement.

2.3.1 Ordering of decision rules

We rely on the following systematic choice of a good decision rule.

Agreement 2.3.2 (Design principle: "expectation"-minimization design)

- *A functional* $\tilde{\mathcal{E}}_{\mathcal{R}}$*, called "*expectation*", is selected by designer. It assigns to functions in (2.1) (determined by the considered loss function* $\mathcal{Z}$ *and indexed by inspected decision rules* $\mathcal{R} \in \mathcal{R}^*$*) a non-negative "*expected loss*"* $\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{Z}]$

$$\begin{aligned} \tilde{\mathcal{E}}_{\mathcal{R}} &: \mathcal{Z}_{\mathcal{R}^*} \to [0, \infty] \\ \mathcal{Z}_{\mathcal{R}^*} &\equiv \{\mathcal{Z}_{\mathcal{R}} : \Upsilon^* \to [0, \infty],\ \mathcal{Z}_{\mathcal{R}}(\Upsilon) \equiv \mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon)\}_{\mathcal{R} \in \mathcal{R}^*} \end{aligned} \tag{2.2}$$

- *The minimizer of* $\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon)] \equiv \tilde{\mathcal{E}}[\mathcal{Z}_{\mathcal{R}}]$ *found in* $\mathcal{R}^*$ *is taken as the optimal decision rule.*

The outcome of this design depends on the introduced "*expectation*" $\tilde{\mathcal{E}}_{\mathcal{R}}$. It has to avoid (at least) unequivocally bad rules. Such bad rules are identified here with dominated decision rules.

Agreement 2.3.3 (Dominated decision rules; strictly isotonic expectation) *Let a loss function* $\mathcal{Z}$ *measure the quality of the behavior. The decision rule* $\mathcal{R} : \mathcal{Q}^* \to a^*$ *is called* dominated *iff there is another decision rule* $\tilde{\mathcal{R}} : \mathcal{Q}^* \to a^*$ *such that for*

$$\mathcal{Z}_{\mathcal{R}} \geq \mathcal{Z}_{\tilde{\mathcal{R}}} \ \Leftrightarrow\ \mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon) \geq \mathcal{Z}(\mathcal{Q}_{\tilde{\mathcal{R}}}, \Upsilon),\ \forall \Upsilon \in \Upsilon^*. \tag{2.3}$$

The dominated decision rule is called strictly dominated *iff there is a non-trivial subset of* Υ^* *on which the inequality (2.3) is strict.*

The "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$ *(2.2) is said to be strictly isotonic if for a decision rule* $\mathcal{R}$ *strictly dominated by a decision rule* $\tilde{\mathcal{R}}$ *it holds*

$$\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}}, \Upsilon)] > \tilde{\mathcal{E}}_{\tilde{\mathcal{R}}}[\mathcal{Z}(\mathcal{Q}_{\tilde{\mathcal{R}}}, \Upsilon)]$$

Now we can specify the dominated decision rules as those to be surely avoided.

Requirement 2.3.1 (Inadmissibility of strictly dominated decision rules) *The considered "expectation"-minimization design, Agreement 2.3.2, must not lead to the strictly dominated decision rule.*

We want to select the "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$ so that the decision rules are ordered as objectively as possible. It means that the ordering should weakly depend on the inspected set of rules $\mathcal{R}^*$, on the loss function $\mathcal{Z}$ considered and on the "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$ selected by the designer.

Requirement 2.3.2 (Independence of $\mathcal{R}^*$) *The chosen "expectation" should fulfill Requirement 2.3.1 even if we restrict the set of possible decision rules* $\mathcal{R}^*$ *to any of its non-trivial subsets that contains at least two different rules while at least one of them gives a finite "expected" loss.*

Proposition 2.3.1 (Isotonic ordering) *Assume that there is a rule in* $\mathcal{R}^*$ *for which the "expected" loss is finite. Then, Requirements 2.3.1, 2.3.2 are fulfilled, i.e. the chosen rule is not dominated even if the set* $\mathcal{R}^*$ *is restricted, iff the "expectation" is strictly isotonic, see Agreement 2.3.3.*

Proof:

1. We prove by contradiction that – with strictly isotonic "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$ – the minimizer cannot be strictly dominated.

   Let $\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{Z}]$ be strictly isotonic on its domain $\mathcal{Z}_{\mathcal{R}^*}$ (2.2) and $\mathcal{R}_o \in \mathcal{R}^*$ be a minimizer of the "expected" loss. The minimizer gives necessarily a finite value of the corresponding $\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{Z}]$. Let $\mathcal{R}_d \in \mathcal{R}^*$ dominate it strictly. Then, because of the construction of $\mathcal{R}_o$, the strict dominance and strictly isotonic nature of $\tilde{\mathcal{E}}_{\mathcal{R}}$, we get the following contradictory inequality

   $$\tilde{\mathcal{E}}_{\mathcal{R}_o}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}_o}, \Upsilon)] \underbrace{\leq}_{minimum} \tilde{\mathcal{E}}_{\mathcal{R}_d}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}_d}, \Upsilon)] \underbrace{<}_{strictly\ isotonic} \tilde{\mathcal{E}}_{\mathcal{R}_o}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}_o}, \Upsilon)].$$

2. We prove by contradiction that use of an "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$ that is not strictly isotonic leads to violation of Requirement 2.3.1 when Requirement 2.3.2 holds.

   If $\tilde{\mathcal{E}}_{\mathcal{R}}[\mathcal{Z}]$ is not strictly isotonic on its domain $\mathcal{Z}_{\mathcal{R}^*}$ (2.2) then there is rule $\mathcal{R}_1 \in \mathcal{R}^*$ strictly dominated by a decision rule $\mathcal{R}_d \in \mathcal{R}^*$ such that

   $$\tilde{\mathcal{E}}_{\mathcal{R}_d}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}_d}, \Upsilon)] \geq \tilde{\mathcal{E}}_{\mathcal{R}_1}[\mathcal{Z}(\mathcal{Q}_{\mathcal{R}_1}, \Upsilon)].$$

   If we restrict the set of decision rules $\mathcal{R}^*$ to the pair of rules $\{\mathcal{R}_d, \mathcal{R}_1\}$ then $\mathcal{R}_1$ can always be taken as the optimal decision rule. Thus, under Requirement 2.3.2, Requirement 2.3.1 is not fulfilled with such an "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$.

□

Requirement 2.3.2 guarantees suitability of the "expectation" to a wide range of decision rules. The "full objectivity" of ordering of decision rules is approached if the "expectation" $\tilde{\mathcal{E}}_{\mathcal{R}}$ is made weakly dependent on the loss function considered. We have to adopt rather technical conditions on the set of such loss functions. Essentially, applicability to a very smooth functions and a restricted version of "linearity" of $\tilde{\mathcal{E}}_{\mathcal{R}}$ are required.

Requirement 2.3.3 (Independence of loss function) *Let us consider various loss functions $\mathcal{Z} \in \mathcal{Z}^*$. The "expectation" $\tilde{\mathcal{E}}$ acts on union of the sets $\mathcal{Z}_{\mathcal{R}^*}$ (2.1) of functions with a common uncertainty set Υ^*. The union is taken over the set of the loss functions*

$$\mathcal{Z}^*_{\mathcal{R}^*} \equiv \cup_{\mathcal{Z}\in\mathcal{Z}^*}\mathcal{Z}_{\mathcal{R}^*} \tag{2.4}$$

The set $\mathcal{Z}^*_{\mathcal{R}^*}$ is required to contain a subset of test loss functions *that are zero out of a compact non-empty subset Ω of $\mathcal{Q}^*$ and continuous on Ω.*

The "expectation" is assumed to be an isotonic, sequentially continuous and boundedly uniformly continuous functional on $\mathcal{Z}^*_{\mathcal{R}^*}$ that is, moreover, additive on loss functions with non-overlapping supports

$$\tilde{\mathcal{E}}[\mathcal{Z}_1 + \mathcal{Z}_2] = \tilde{\mathcal{E}}[\mathcal{Z}_1] + \tilde{\mathcal{E}}[\mathcal{Z}_2] \text{ if } \mathcal{Z}_1\mathcal{Z}_2 = 0,\ \mathcal{Z}_1,\, \mathcal{Z}_2 \in \mathcal{Z}^*_{\mathcal{R}^*}.$$

This technical Requirement allows us to get an integral representation of the "expectation" searched for. Its proof as well as definitions of the adopted non-common terms can be found in [17].

Proposition 2.3.2 (Integral form of "expectation") *Under Requirement 2.3.3, the "expectation" $\tilde{\mathcal{E}}$ has the form*

$$\tilde{\mathcal{E}}[\mathcal{Z}] = \int_\Omega \mathcal{U}(\mathcal{Z}(\mathcal{Q}), \mathcal{Q})\, \mu(d\mathcal{Q}), \quad \text{where} \tag{2.5}$$

μ is a finite regular non-negative Borel measure on Ω. The utility function *$\mathcal{U}$ satisfies $\mathcal{U}(0, \mathcal{Q}) = 0$. It is continuous in values of $\mathcal{Z}(\cdot)$ almost everywhere (a.e.) on Ω, bounded a.e. on Ω for each $\mathcal{Z}$ in the set of the test loss functions.*

Remark(s) 2.3.1

1. *The test loss functions are widely applicable and their consideration implies no practical restriction. The continuity requirements on $\tilde{\mathcal{E}}$ are also widely acceptable.*
2. *The linearity of $\tilde{\mathcal{E}}$ on functions with non-overlapping support seems to be sound. Any loss function $\mathcal{Z} \in \mathcal{Z}^*_{\mathcal{R}^*}$ can be written as $\mathcal{Z} = \mathcal{Z}\chi_\omega + \mathcal{Z}(1-\chi_\omega) \equiv \mathcal{Z}_1 + \mathcal{Z}_2$, $\mathcal{Z}_1\mathcal{Z}_2 = 0$ with χ_ω denoting an* indicator of a set $\omega \subset \Omega \subset \mathcal{Q}^*$. *The indicator χ equals 1 on ω and it is zero outside of it.*

   The loss "expected" on the set ω and its complement should sum to the loss "expected" on the whole set of arguments.
3. *The utility function $\mathcal{U}$ allows the designer to express his/her attitude to the design consequences: he/she might be risk aware, risk prone or risk indifferent [16].*
4. *The utility function $\mathcal{U}$ and the non-negative measure μ are universal for the whole set of test functions. $\mathcal{U}$ and μ are (almost) "objective", i.e. suitable for a wide range of decision tasks facing the same uncertainty.*

We formulate now our final objectivity oriented Requirement.

Requirement 2.3.4 (Indifference of the designer)

- *The designer is risk indifferent, i.e. $\mathcal{U}$ is the identity mapping.*
- *The "expectation" preserves any constant loss $\tilde{\mathcal{E}}[constant] = constant$.*
- *The involved measure μ has Radon-Nikodým derivative, [17], $f(\mathcal{Q})$ with respect to a dominating measure denoted $d\mathcal{Q}$.*

  In the treated cases, $d(\cdot)$ is either Lebesgue or counting measure.

Adopting Requirement 2.3.4, we get the basic representation Proposition that introduces objective "expectation".

Proposition 2.3.3 (Objective "expectation") *Under Requirement 2.3.4, the "expectation" $\tilde{\mathcal{E}}$ (2.5) is formally identical with a* mathematical expectation. *The Radon Nikodým's derivative f, [17], has all properties of the* joint probability (density) function *(*p(d)f*) on $\mathcal{Q}^*$.*

Proof: It is sufficient to observe that the preservation of constants implies that μ is a probabilistic measure, i.e. $\mu \geq 0,\ \mu(\mathcal{Q}^*) = 1$. □

Remark(s) 2.3.2

1. *The (mathematical) "expectation" is distinguished by dropping the sign ˜ as well as the quotation symbols " "*

$$\mathcal{E}[\mathcal{Z}] \equiv \int_{\mathcal{Q}^*} \mathcal{Z}(\mathcal{Q}) f(\mathcal{Q})\, d\mathcal{Q} \equiv \int \mathcal{Z}(\mathcal{Q}) f(\mathcal{Q})\, d\mathcal{Q}. \tag{2.6}$$

2. *The first item in Requirement 2.3.4 is our last step we made in the objectivity direction. It has clear meaning: we support objective, emotionally indifferent designers.*

3. *The last item in Requirement 2.3.4 is unnecessary but it covers all cases we are here interested in. It helps us to deal with simpler objects, namely, with probability density functions (pdf) or probability functions (pf).*

   Mostly, we make no formal distinction between these cases and we use notations related to pdfs. Only when necessary, we underline that we deal with a pf and write integrals as sums.

4. *The notion of* conditional expectation $\mathcal{E}[\bullet|*]$, *[17] is adopted further on and it is treated in a simplistic way as the integral* $\int \bullet(a) f(a|*)\, da$ *weighted by the* conditional pdf $f(a|*)$.

2.3.2 Calculus with pdfs

The joint pdf f on $\mathcal{Q} \equiv (\alpha, \beta, \gamma)$ is analysed/synthesised using several pdfs related to it. Let us recall meaning of pdfs derived from $f(\mathcal{Q})$.

Agreement 2.3.4 (Nomenclature of pdfs; Independence) *We work with*

Name	Meaning
joint pdf $f(\alpha, \beta|\gamma)$ *of* α, β conditioned on γ	*a pdf on* $(\alpha, \beta)^*$ *restricting* $f(\mathcal{Q})$ *on the cross-section of* $\mathcal{Q}^*$ *given by a fixed* γ
marginal pdf $f(\alpha|\gamma)$ *of* α *conditioned on* γ	*a pdf on* α^* *restricting* $f(\mathcal{Q})$ *on the cross-section of* $\mathcal{Q}^*$ *given by a fixed* γ *with no information on* β
marginal pdf $f(\beta|\alpha, \gamma)$ *of* β *conditioned on* α, γ	*a pdf on* β^* *restricting* $f(\mathcal{Q})$ *on the cross-section of* $\mathcal{Q}^*$ *given by a fixed* α, γ

The conditioning symbol $|$ *is dropped if just trivial conditions are considered.*
The pdf $f(\alpha, \beta)$ *is the* lower dimensional joint pdf *(with respect to the pdf* $f(\alpha, \beta, \gamma)$*),* $f(\beta)$ *is the* marginal pdf.
Quantities α *and* β *are* conditionally independent *under the condition* γ *iff*

$$f(\alpha, \beta|\gamma) = f(\alpha|\gamma) f(\beta|\gamma). \tag{2.7}$$

Our manipulations with the introduced pdfs rely on the following calculus.

Proposition 2.3.4 (Calculus with pdfs) *For generic* $(\alpha, \beta, \gamma) \in (\alpha, \beta, \gamma)^*$ *it holds:*

Non-negativity $f(\alpha, \beta|\gamma),\ f(\alpha|\beta, \gamma),\ f(\beta|\alpha, \gamma),\ f(\beta|\gamma) \geq 0$.

Normalization $\int f(\alpha, \beta|\gamma)\, d\alpha d\beta = \int f(\alpha|\beta, \gamma)\, d\alpha = \int f(\beta|\alpha, \gamma)\, d\beta = 1$.

Chain rule $f(\alpha, \beta|\gamma) = f(\alpha|\beta, \gamma) f(\beta|\gamma) = f(\beta|\alpha, \gamma) f(\alpha|\gamma)$.

Marginalization $f(\beta|\gamma) = \int f(\alpha, \beta|\gamma)\, d\alpha,\ f(\alpha|\gamma) = \int f(\alpha, \beta|\gamma)\, d\beta$.

Bayes rule

$$f(\beta|\alpha,\gamma) = \frac{f(\alpha|\beta,\gamma)f(\beta|\gamma)}{f(\alpha|\gamma)} = \frac{f(\alpha|\beta,\gamma)f(\beta|\gamma)}{\int f(\alpha|\beta,\gamma)f(\beta|\gamma)\,d\beta} \propto f(\alpha|\beta,\gamma)f(\beta|\gamma). \tag{2.8}$$

The proportion sign, $\propto$, *means that the factor independent of β and uniquely determined by the normalization is not explicitly written in the equality represented.*

Independence equivalents

$$f(\alpha,\beta|\gamma) = f(\alpha|\gamma)f(\beta|\gamma) \Leftrightarrow f(\alpha|\beta,\gamma) = f(\alpha|\gamma) \text{ or } f(\beta|\alpha,\gamma) = f(\beta|\gamma). \tag{2.9}$$

Proof: For motivation see [14], a more precise and more technical treatment exploits the measure theory [17]. Technically, an intermediate insight can be gained by considering loss functions dependent only on a part of $\mathcal{Q}$ or with some parts of $\mathcal{Q}$ "fixed by the condition", [13]. □

Remark(s) 2.3.3

1. *Duplicates in the presented formulas stress symmetry of many of them.*
2. *The technical fact that the identities, like (2.9), hold only almost everywhere, (a.e.) is mostly omitted in the subsequent explanations.*
3. *The Bayes rule (2.8) is a simple consequence of previous formulas. Its importance in this text justifies the explicit presentation in various forms.*
4. *The symmetric identity (2.9) says that β does not influence the description of α if α and β are conditionally independent for a given γ.*

Often, a pdf $f(\alpha)$ of a multi-variate variable $\alpha \in \alpha^*$ is given and mapped on a variable β by a mapping $T : \alpha^* \to \beta^* \equiv T(\alpha^*)$. We need to specify the corresponding pdf $f(\beta)$. Rules for determining the pdf $f(\beta)$ are implied by the obvious need to preserve the expectation.

Proposition 2.3.5 (Pdfs of transformed quantities) *Let the expectation $\mathcal{E}$, acting on functions $\mathcal{B} : \beta^* \to (-\infty,\infty)$, be specified by the pdf $f_T(\beta)$, i.e.*

$$\mathcal{E}_T[\mathcal{B}] = \int \mathcal{B}(\beta)f_T(\beta)\,d\beta.$$

This functional expresses the same expectation as $\mathcal{E}[\mathcal{B}] = \int \mathcal{B}(T(\alpha))f(\alpha)\,d\alpha$ iff

$$\int_{T(A)} f_T(T(\alpha))\,dT(\alpha) = \int_A f(\alpha)\,d\alpha \tag{2.10}$$

for all measurable sets $A \subset \alpha^*$.

Let α be a real vector , $\alpha \equiv [\alpha_1,\ldots,\alpha_{\mathring{\alpha}}]$ and $T = [T_1,\ldots,T_{\mathring{\alpha}}]$ bijection (one-to-one mapping) with finite continuous partial derivatives a.e. on α^*

$$J_{ij}(\alpha) \equiv \frac{\partial T_i(\alpha)}{\partial \alpha_j},\ i,j = 1,\ldots,\mathring{\alpha}, \tag{2.11}$$

for all entries T_i of T and entries α_j of α. Then,

$$f_T(T(\alpha))|J(\alpha)| = f(\alpha), \quad \text{where} \tag{2.12}$$

$|\cdot|$ *denotes* determinant *of the matrix in its argument.*

Proof: Proposition describes substitutions in multi-variate integrals, see [17, 18]. □

It is useful to summarize basic properties of expectation, which help us to simplify formal manipulations.

Proposition 2.3.6 (Basic properties of $\mathcal{E}$) *For arbitrary functions $\mathcal{Z}_1(\cdot)$, $\mathcal{Z}_2(\cdot)$ on which the conditional expectation $\mathcal{E}[\cdot|\gamma]$ is well defined, $\mathcal{E}[\cdot|\gamma]$ has the following properties.*

Isotonic nature of $\mathcal{E}[\cdot|\gamma]$: $\mathcal{Z}_1 \leq \mathcal{Z}_2, cf.\ (2.3), \Rightarrow \mathcal{E}[\mathcal{Z}_1|\gamma] \leq \mathcal{E}[\mathcal{Z}_2|\gamma]$.

Linearity of $\mathcal{E}[\cdot|\gamma]$: $\mathcal{E}[A(\gamma)\mathcal{Z}_1 + B(\gamma)\mathcal{Z}_2|\gamma] = A(\gamma)\mathcal{E}[\mathcal{Z}_1|\gamma] + B(\gamma)\mathcal{E}[\mathcal{Z}_2|\gamma]$ *for arbitrary coefficients A, B depending at most on the condition γ.*

Chain rule for expectation: $\mathcal{E}\left[\mathcal{E}[\cdot|\gamma,\zeta]|\gamma\right] = \mathcal{E}[\cdot|\gamma]$ *for an arbitrary additional condition ζ.*

Conditional covariance *of a vector α $\mathrm{cov}[\alpha|\gamma] \equiv \mathcal{E}\left[(\alpha - \mathcal{E}[\alpha|\gamma])(\alpha - \mathcal{E}[\alpha|\gamma])'|\gamma\right]$ is related to the non-central moments through the formula*

$$\mathrm{cov}[\alpha|\gamma] = \mathcal{E}[\alpha\alpha'|\gamma] - \mathcal{E}[\alpha|\gamma]\mathcal{E}[\alpha'|\gamma], \quad {}' \text{ is transposition.} \tag{2.13}$$

Jensen inequality *bounds expectation of a convex function $T_\gamma : \alpha^* \rightarrow (-\infty, \infty)$*

$$\mathcal{E}[T_\gamma(\alpha)|\gamma] \geq T_\gamma\left(\mathcal{E}[\alpha|\gamma]\right). \tag{2.14}$$

Proof: All statements can be verified by using the integral expression (2.6) of the expectation. Proof of Jensen inequality can be found e.g. in [19]. □

Remark(s) 2.3.4

1. *Proposition is formulated for the conditional expectation. The unconditional case is formally obtained by omitting the condition used.*
2. *Note that whenever the expectation is applied to a vector (matrix) function V it should be understood as the vector (matrix) of expectations*

$$\left[\mathcal{E}(V)\right]_i \equiv \mathcal{E}(V_i). \tag{2.15}$$

2.3.3 Basic decision-making lemma

The optimal selection of *admissible decision rules* relies on the following key proposition that converts minimization over mappings to an "ordinary" minimization.

Proposition 2.3.7 (Basic decision-making lemma) *The optimal* admissible decision rule ${}^{\lfloor o}\mathcal{R}$

$${}^{\lfloor o}\mathcal{R}(\mathcal{P}_a) \equiv {}^{\lfloor o}a(\mathcal{P}_a), \ \forall \mathcal{P}_a \in \mathcal{P}_a^*$$

minimizing the expected loss (2.6) can be constructed value-wise as follows.

To each $\mathcal{P}_a \in \mathcal{P}_a^*$, a minimizing argument ${}^{\lfloor o}a(\mathcal{P}_a)$ in

$$\min_{a \in a^*} \mathcal{E}[\mathcal{Z}(\mathcal{P}_a, a, \mathcal{F}_a)|a, \mathcal{P}_a] \tag{2.16}$$

is assigned as the value of the optimal decision rule corresponding to the considered argument. The reached minimum is

$$\min_{\{\mathcal{R}:\, \mathcal{P}_a^* \rightarrow a^*\}} \mathcal{E}[\mathcal{Z}(\mathcal{P}_a, a, \mathcal{F}_a)] = \mathcal{E}\left\{\min_{a \in a^*} \mathcal{E}[\mathcal{Z}(\mathcal{P}_a, a, \mathcal{F}_a)|a, \mathcal{P}_a]\right\}. \tag{2.17}$$

Proof: Let us fix an arbitrary $\mathcal{P}_a \in \mathcal{P}_a^*$. The definition of minimum implies

$$\mathcal{E}[\mathcal{Z}(\mathcal{P}_a, {}^{\lfloor o}\mathcal{R}(\mathcal{P}_a), \mathcal{F}_a)|\, {}^{\lfloor o}\mathcal{R}(\mathcal{P}_a), \mathcal{P}_a] \leq \mathcal{E}[\mathcal{Z}(\mathcal{P}_a, a, \mathcal{F}_a)|a, \mathcal{P}_a], \ \forall a \in a^*.$$

Let an admissible rule $\mathcal{R} : \mathcal{P}_a^* \rightarrow a^*$ assign a decision $a \in a^*$ to the considered $\mathcal{P}_a$. Then, the previous inequality can be written

$$\mathcal{E}[\mathcal{Z}(\mathcal{P}_a, {}^{\lfloor o}\mathcal{R}(\mathcal{P}_a), \mathcal{F}_a)|\, {}^{\lfloor o}\mathcal{R}(\mathcal{P}_a), \mathcal{P}_a] \leq \mathcal{E}[\mathcal{Z}(\mathcal{P}_a, \mathcal{R}(\mathcal{P}_a), \mathcal{F}_a)|\mathcal{R}(\mathcal{P}_a), \mathcal{P}_a].$$

We apply unconditional expectation $\mathcal{E}[\cdot]$ acting on functions of $\mathcal{P}_a$ to this inequality. Due to the isotonic nature of $\mathcal{E}$, the inequality is preserved. The chain rule for expectations, see Proposition 2.3.6, implies that on the left hand side of the resulting inequality the value of the unconditional expected loss for ${}^{\lfloor o}\mathcal{R}$ is found. ${}^{\lfloor o}\mathcal{R}$ assigns to each $\mathcal{P}_a \in \mathcal{P}_a^*$ the decision ${}^{\lfloor o}a(\mathcal{P}_a)$. On the right hand side of the discussed inequality, we get the unconditional expected loss corresponding to an arbitrarily chosen $\mathcal{R} : \mathcal{P}_a^* \to a^*$. □

Proposition and its proof imply no preferences if there are more absolutely minimizing arguments ${}^{\lfloor o}a(\mathcal{P}_a)$. We can use any of them or switch between them in a random manner whenever the pdf $f(a_t|\mathcal{P}_{a_t})$ has its support concentrated on them. This is an example where a *randomized causal strategy* may occur. We specify it formally as it is used later on more extensively.

Agreement 2.3.5 (Outer model of randomized decision strategy) *The pdf* $f(a|\mathcal{P}_a)$ *is called* outer model of the decision rule. *The collection of pdfs* $\{f(a_t|\mathcal{P}_{a_t})\}_{t\in t^*}$ *forms the* outer model of the decision strategy.

A decision rule $f(a|\mathcal{P}_a)$ *is called* randomized decision rule *if its support contains at least two different values of* a_t. *The strategy is called* randomized strategy *if some of its rules is randomized.*

Remark(s) 2.3.5

1. *We do not enter the technical game with* ε*-optimum: the existence of the various minimizing arguments is implicitly supposed.*
2. *It is worth repeating that the optimal decision rule is constructed value-wise. Formally, the minimization should be performed for all possible instances of experience* $\mathcal{P}_a \in \mathcal{P}_a^*$ *in order to get the decision rule.*

   Often, we are interested in the optimal decision for a given fixed, say observed, experience. Then, just a single minimization is necessary. This is typically the case of the estimation problem. This possibility makes the main distinction from the dynamic design, when optimal strategy, a sequence of decision rules, is searched for. In this case, discussed in next Section, the construction of decision rules is necessary. This makes the dynamic design substantially harder and mostly infeasible [20, 21].

2.4 Dynamic design

We are searching for the optimal admissible strategy. We consider the usual case, when its individual rules use a non-decreasing sequence of experience. The extending experience models increasing amount of data available for the decision-making.

2.4.1 Dynamic programming

The optimal admissible strategy can be found by using a stochastic version of celebrated *dynamic programming* [22]. It is nothing but a repetitive application of Proposition 2.3.7.

Proposition 2.4.1 (Stochastic dynamic programming) *The optimal causal strategy* ${}^{\lfloor o}\{\mathcal{R}_t : \mathcal{P}_{a_t}^* \to a_t^*\}_{t\in t^*} \in \{\mathcal{R}_t : \mathcal{P}_{a_t}^* \to a_t^*\}_{t\in t^*}$ *acting on non-decreasing sequence of experience* $\mathcal{P}_{a_t} \subset \mathcal{P}_{a_{t+1}}$ *and minimizing the expected loss function* $\mathcal{E}[\mathcal{Z}(\mathcal{Q})]$ *can be constructed in a value-wise way. For every* $t \in t^*$ *and each* $\mathcal{P}_{a_t} \in \mathcal{P}_{a_t}^*$, *it is sufficient to take a minimizing argument* ${}^{\lfloor o}a(\mathcal{P}_{a_t})$ *in*

$$\mathcal{V}(\mathcal{P}_{a_t}) = \min_{a_t \in a_t^*} \mathcal{E}[\mathcal{V}(\mathcal{P}_{a_{t+1}})|a_t, \mathcal{P}_{a_t}],\ t \in t^* \tag{2.18}$$

as the decision generated by the t*-th rule of the optimal strategy, i.e.* ${}^{\lfloor o}a(\mathcal{P}_{a_t}) = {}^{\lfloor o}\mathcal{R}_t(\mathcal{P}_{a_t})$.

The recursion (2.18) is performed in the backward manner against the course given by the increasing experience. It starts with

$$\mathcal{V}(\mathcal{P}_{a_{\mathring{t}+1}}) \equiv \mathcal{Z}(\mathcal{Q}). \tag{2.19}$$

The reached minimum has the value $\mathcal{E}[\mathcal{V}(\mathcal{P}_{a_1})] = \min_{\{\mathcal{R}_t : \mathcal{P}_{a_t}^* \to a_t^*\}_{t\in t^*}} \mathcal{E}[\mathcal{Z}(\mathcal{Q})]$.

Proof: Let us define $\mathcal{P}_{a_{\mathring{t}+1}} \equiv \mathcal{Q}$ in order to get a uniform notation It is legitimate as $a_{\mathring{t}+1}$ is not optimized. The definition of minimum and Proposition 2.3.7 imply

$$\min_{\{\mathcal{R}_t:\, \mathcal{P}^*_{a_t}\to a^*_t\}_{t\in t^*}} \mathcal{E}[\mathcal{Z}(\mathcal{P}_{a_{\mathring{t}+1}})] = \min_{\{\mathcal{R}_t:\, \mathcal{P}^*_{a_t}\to a^*_t\}_{t<\mathring{t}}} \left\{ \min_{\mathcal{R}_{\mathring{t}}:\, \mathcal{P}^*_{a_{\mathring{t}}}\to a^*_{\mathring{t}}} \mathcal{E}[\mathcal{Z}(\mathcal{P}_{a_{\mathring{t}+1}})] \right\} \underbrace{=}_{(2.17)}$$

$$= \min_{\{\mathcal{R}_t:\, \mathcal{P}^*_{a_t}\to a^*_t\}_{t<\mathring{t}}} \mathcal{E}\left[\min_{a_{\mathring{t}}\in a^*_{\mathring{t}}} \mathcal{E}[\mathcal{Z}(\mathcal{P}_{a_{\mathring{t}+1}})|a_{\mathring{t}}, \mathcal{P}_{a_{\mathring{t}}}] \right].$$

Denoting $\mathcal{V}(\mathcal{P}_{a_{\mathring{t}}}) \equiv \min_{a_{\mathring{t}}\in a^*_{\mathring{t}}} \mathcal{E}[\mathcal{Z}(\mathcal{P}_{a_{\mathring{t}+1}})|a_{\mathring{t}}, \mathcal{P}_{a_{\mathring{t}}}]$, we proved the first step of the recursion and specified the start (2.19). The following step becomes

$$\min_{\{\mathcal{R}_t:\, \mathcal{P}^*_{a_t}\to a^*_t\}_{t<\mathring{t}}} \mathcal{E}\left[\mathcal{V}(\mathcal{P}_{a_{\mathring{t}}})\right].$$

We face the identical situation as above with the horizon decreased by one. Thus, the procedure can be repeated till the initial rule $\mathcal{R}_1$ is constructed. □

The optimization relies on our ability to evaluate the expectations

$$\mathcal{E}[\mathcal{V}(\mathcal{P}_{a_{t+1}})|a_t, \mathcal{P}_{a_t}] = \int_{\Delta^*_{a_t}} \mathcal{V}(\mathcal{P}_{a_t}, a_t, \Delta_{a_t}) f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t})\, d\Delta_{a_t},\ \forall t \in t^*.$$

The introduced *innovation* Δ_{a_t} contains those observable quantities that cannot be used for the choice of a_t. They are not in $\mathcal{P}_{a_t}$ but they belong to $\mathcal{P}_{a_{t+1}} \setminus a_t$. The pdfs $\{f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t})\}_{t\in t^*}$ model the relationships of Δ_{a_t} to $\mathcal{P}_t$ and a_t.

Agreement 2.4.1 (Outer model of the system) *The collection of pdfs*

$$\{f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t})\}_{t\in t^*}, \tag{2.20}$$

required for the optimal design, is called outer model of the system.

Remark(s) 2.4.1

1. *The term outer model of the system is abbreviated to the model of the system or even to the model. The exact meaning is clear from the specific context.*
2. *Often, the innovation $\Delta_{a_t} = y_t$ = observable output of the system. The set-point to which the output should be driven by the chosen input u_t has to be included into Δ_{u_t} if its values are uncertain.*

The following agreement is used in a presentation of the most common version of dynamic programming.

Agreement 2.4.2 (Data driven design; internal quantities) *The design is called* data driven *if realization $\mathcal{Q}$ consists of optional and potentially observable quantities, i.e. with ignorance consisting of unobserved data only*

$$\mathcal{Q} \equiv (\Delta(\mathring{t}), a(\mathring{t})) \equiv (\mathcal{P}_a, a, \mathcal{F}_a). \tag{2.21}$$

Quantities in $\mathcal{Q}$ that are never observed directly are called internal quantities.

Proposition 2.4.2 (Dynamic programming for additive loss function) *Let us consider data driven design and search for the optimal admissible strategy $\{\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a^*_t\}_{t\in t^*}$ acting on a non-decreasing sequence of experience $\{\mathcal{P}_{a_t}\}_{t\in t^*}$.*

Then, the optimal strategy $\{ {}^{\lfloor o}\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a^*_t\}_{t\in t^*}$ minimizing the expected additive loss function

$$\mathcal{E}\left[\sum_{t\in t^*} z(\Delta(t), a(t))\right] \tag{2.22}$$

can be constructed value-wise. For all $t \in t^*$ and to each $\mathcal{P}_{a_t} \in \mathcal{P}^*_{a_t}$, a minimizing argument ${}^{\lfloor o}a(\mathcal{P}_{a_t}) = {}^{\lfloor o}\mathcal{R}_t(\mathcal{P}_{a_t})$ in

$$\mathcal{V}(\mathcal{P}_{a_t}) = \min_{a_t \in a_t^*} \mathcal{E}[z(\Delta(t), a(t)) + \mathcal{V}(\mathcal{P}_{a_{t+1}})|a_t, \mathcal{P}_{a_t}], \ t \in t^* \tag{2.23}$$

is assigned. The recursion (2.23) is performed in the backward manner against the course given by the increasing experience, starting from

$$\mathcal{V}(\mathcal{P}_{a_{\mathring{t}+1}}) \equiv 0. \tag{2.24}$$

The reached minimum has the value

$$\min_{\{\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a_t^*\}_{t \in t^*}} \mathcal{E}[\mathcal{Z}(\mathcal{P}_{a_{\mathring{t}+1}})] = \mathcal{E}[\mathcal{V}(\mathcal{P}_{a_1})].$$

Proof: It follows exactly the line of Proposition 2.4.1 with a modified definition of the function $\mathcal{V}(\cdot)$

$$\mathcal{V}(\mathcal{P}_{a_t}) \equiv \min_{\{\mathcal{R}_\tau : \mathcal{P}^*_{a_\tau} \to a_\tau^*\}_{\tau \geq t}} \mathcal{E}\left[\sum_{\tau \geq t} z(\Delta(\tau), a(\tau))|a_t = \mathcal{R}_t(\mathcal{P}_{a_t}), \mathcal{P}_{a_t}\right]. \tag{2.25}$$

□

Agreement 2.4.3 (Partial loss; Bellman function) *The function $z(\Delta(t), a(t))$ is called the* partial loss. *The function $\mathcal{V}(\cdot)$ occurring in dynamic programming is called* Bellman function. *Bellman function $\mathcal{V}(\cdot)$ in (2.25) is also called the* optimal loss-to-go.

2.4.2 Fully probabilistic design

A specific but rather general design is formulated and solved here. It is believed to form a bridge between optimal and practically optimal design.

Here, and in many other places, we need the notion of the *Kullback-Leibler distance* [23] that measures well proximity of a pair of pdfs.

Agreement 2.4.4 (Kullback-Leibler distance) *Let f, g be a pair of pdfs acting on a common set x^*. Then, the Kullback-Leibler distance $\mathcal{D}(f||g)$ is defined by the formula*

$$\mathcal{D}(f||g) \equiv \int_{x^*} f(x) \ln\left(\frac{f(x)}{g(x)}\right) dx. \tag{2.26}$$

For conciseness, the Kullback-Leibler distance is referred to as the KL distance.

Proposition 2.4.3 (Basic properties of KL distance)
Let f, g be a pair of pdfs acting on a same set. It holds

1. *$\mathcal{D}(f||g) \geq 0$,*
2. *$\mathcal{D}(f||g) = 0$ iff $f = g$ (a.e.),*
3. *$\mathcal{D}(f||g) = \infty$ iff on a set of a positive measure $f > 0$ and $g = 0$,*
4. *$\mathcal{D}(f||g) \not\equiv \mathcal{D}(g||f)$; the KL distance does not obey triangle inequality.*

Proof: See, for instance, [19] □

Now we are ready to formulate and solve the fully probabilistic design problem. A simple version is presented here considering the data-driven design, see Agreement 2.4.2. In this case, the joint pdf $f(\mathcal{Q}) \equiv f(\Delta(\mathring{t}), a(\mathring{t}))$ describing observable quantities of interest can be factorized by a repetitive use of the chain rule, see Proposition 2.3.4,

$$f(\Delta(\mathring{t}), a(\mathring{t})) = \prod_{t \in t^*} f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t}) f(a_t|\mathcal{P}_{a_t}). \tag{2.27}$$

The first factors $\{f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})\}_{t\in t^*}$ under the product sign describe possible reactions of the system on the decision a_t under the experience $\mathcal{P}_{a_t}$. These pdfs form the outer model of the system, see Agreement 2.4.1. Similarly, $\{f(a_t|\mathcal{P}_{a_t})\}_{t\in t^*}$ represent an outer model of a randomized decision strategy to be chosen, see Agreement 2.3.5. Looking at the joint pdf (2.27), it seems to be "natural" to formulate the design as an attempt to make this pdf as close as possible to some "ideal" joint pdf.

Agreement 2.4.5 (Fully probabilistic design) *The fully probabilistic, data driven, design specifies its target through an* ideal pdf

$$\lfloor^I f(\mathcal{Q}) = \prod_{t\in t^*} \lfloor^I f(\Delta_{a_t}|a_t,d(t-1))\, \lfloor^I f(a_t|d(t-1)).$$

The optimal admissible, possibly randomized, decision strategy is defined as a minimizer of the KL distance (2.26) of $f(d(\mathring{t})) = f(\Delta(\mathring{t}),a(\mathring{t}))$ and $\lfloor^I f(d(\mathring{t})) = \lfloor^I f(\Delta(\mathring{t}),a(\mathring{t}))$

$$\mathcal{D}(f||\lfloor^I f) \equiv \int f(\Delta(\mathring{t}),a(\mathring{t}))\ln\left(\frac{f(\Delta(\mathring{t}),a(\mathring{t}))}{\lfloor^I f(\Delta(\mathring{t}),a(\mathring{t}))}\right) d(\Delta(\mathring{t}),a(\mathring{t})). \qquad (2.28)$$

Proposition 2.4.4 (Solution of fully probabilistic design) *The optimal strategy minimizing the KL distance (2.28) has the form*

$$\begin{aligned}
f(a_t|d(t-1)) &= \lfloor^I f(a_t|d(t-1))\frac{\exp[-\omega_\gamma(a_t,d(t-1))]}{\gamma(d(t-1))}, \quad \textit{where} && (2.29)\\
\gamma(d(t-1)) &\equiv \int_{a_t^*} \lfloor^I f(a_t|d(t-1))\exp[-\omega(a_t,d(t-1))]\,da_t\\
\omega_\gamma(a_t,d(t-1)) &\equiv \int_{\Delta_t^*} f(\Delta_t|a_t,d(t-1))\ln\left(\frac{f(\Delta_t|a_t,d(t-1))}{\gamma(d(t))\,\lfloor^I f(\Delta_t|a_t,d(t-1))}\right) d\,\Delta_t\\
\gamma(d(\mathring{t})) &= 1. && (2.30)
\end{aligned}$$

The solution is performed against the time course, starting at $t=\mathring{t}$.

Proof: Using chain rule, we can write the KL distance in the form

$$\begin{aligned}
\mathcal{D}\left(f||\lfloor^I f\right) &= \mathcal{E}\left\{\sum_{t\in t^*}\int f(a_t|d(t-1))\left[\ln\left(\frac{f(a_t|d(t-1))}{\lfloor^I f(a_t|d(t-1))}\right)+\omega(a_t,d(t-1))\right]da_t\right\} \text{ with}\\
\omega(a_t,d(t-1)) &\equiv \int f(\Delta_t|a_t,d(t-1))\ln\left(\frac{f(\Delta_t|a_t,d(t-1))}{\lfloor^I f(\Delta_t|a_t,d(t-1))}\right)d\Delta_t. \text{ Let us denote } \ln(\gamma(d(t))) \equiv\\
&\equiv \min_{\{f(a_{\tau+1}|d(\tau))\}_{\tau=t}^{\mathring{t}}} \mathcal{E}\left\{\sum_{\tau=t+1}^{\mathring{t}}\int f(a_\tau|d(\tau-1))\left[\ln\left(\frac{f(a_\tau|d(\tau-1))}{\lfloor^I f(a_\tau|d(\tau-1))}\right)+\omega(a_\tau,d(\tau-1))\right]da_\tau|d(t)\right\}.
\end{aligned}$$

Then, this definition implies that $\gamma(d(\mathring{t}))=1$ and

$$\begin{aligned}
-\ln(\gamma(d(t))) &\equiv \min_{f(a_{t+1}|d(t))}\int f(a_{t+1}|d(t))\left[\ln\left(\frac{f(a_{t+1}|d(t))}{\lfloor^I f(a_{t+1}|d(t))}\right)+\omega_\gamma(a_{t+1},d(t))\right]da_{t+1} \text{ with}\\
\omega_\gamma(a_{t+1},d(t)) &\equiv \int f(\Delta_{t+1}|a_{t+1},d(t))\ln\left(\frac{f(\Delta_{t+1}|a_{t+1},d(t))}{\gamma(d(t+1))\,\lfloor^I f(\Delta_{t+1}|a_{t+1},d(t))}\right)d\Delta_{t+1}. \text{ It implies}\\
-\ln(\gamma(d(t))) &\equiv \min_{f(a_{t+1}|d(t))}\int f(a_{t+1}|d(t))
\end{aligned}$$

$$\left[\ln\left(\frac{f(a_{t+1}|d(t))}{\frac{\lfloor^I f(a_{t+1}|d(t))\exp[-\omega_\gamma(a_{t+1},d(t))]}{\int \lfloor^I f(\tilde{a}_{t+1}|d(t))\exp[-\omega_\gamma(\tilde{a}_{t+1},d(t))]\,d\tilde{a}_{t+1}}}\right)da_{t+1} - \ln\left(\int \lfloor^I f(a_{t+1}|d(t))\exp\left[-\omega_\gamma(a_{t+1},d(t))\right]da_{t+1}\right)\right].$$

The first term in the above identity is the KL distance, that reaches its smallest zero value for the claimed pdf. At the same time, it defines the form of the reached minima. For an alternative derivation with more details see [24, 25]. $\square$

Remark(s) 2.4.2

1. *At a descriptive level, the stochastic dynamic programming consists of a sequence of the evaluation pairs*
$$(\textit{take conditional expectation, minimize}).$$
Except of a few numerically solvable cases, some approximation techniques have to be employed. The complexity of the approximated optimum prevents a systematic use of the standard approximation theory and various ad hoc schemes are adopted.

   The fully probabilistic design finds minimizers explicitly and reduces the design to a sequence of conceptually feasible multi-variate integrations.

2. *The found optimal strategy is randomized and obviously causal one. The technological restriction are met trivially if the chosen ideal strategy meets them, if* $\text{supp}\left[\,{}^{\lfloor I}f(a_t|\mathcal{P}_{a_t})\right] \subset a_t^*$, *cf. (2.29).*

2.4.3 Asymptotic of design

The analysis outlined here serves us mainly for interpretation purposes. Thus, all technicalities are suppressed as much as possible.

The asymptotic of the dynamic programming is analyzed for horizon $\mathring{t} \to \infty$ within this section. The data-driven case with an additive loss function (2.22) is considered. Note that the general loss function can always be converted in it by defining

$$z(\Delta(t), a(t))) = \begin{cases} \mathcal{Z}(\Delta(\mathring{t}), a(\mathring{t})) & \text{if } t = \mathring{t} \\ 0 & \text{otherwise} \end{cases}. \tag{2.31}$$

We deal, however, with a simpler but still useful case by assuming that

- there is a finite-dimensional information state, i.e. $\mathcal{P}_{a_t} \equiv x_{t-1} \equiv$ a finite-dimensional vector,
- the partial loss depends on x_t and a_t only $z(\Delta(t), a(t)) \equiv z(x_t, a_t)$, i.e. the considered loss is
$$\mathcal{Z}(\Delta(\mathring{t}), a(\mathring{t})) = \sum_{t \in t^*} z(x_t, a_t).$$

Agreement 2.4.6 (Stabilizing strategy) *Let us consider sequence of decision-making problems with the growing horizon $\mathring{t} \to \infty$. The strategy $\{\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a^*_t\}_{t \in t^* \equiv \{1,\ldots,\mathring{t}\}}$ $\forall \mathring{t} \in \{1, 2, \ldots\}$ is called* stabilizing strategy *if there is a finite constant c such that*

$$\mathcal{E}[z(x_t, a_t)|a_t, \mathcal{P}_{a_t}] \leq c < \infty, \; t \in \{1, 2, 3, \ldots\}. \tag{2.32}$$

Proposition 2.4.5 (Asymptotic design) *Let a stabilizing strategy exist. Then there is an optimal, stationary strategy, formed by a repetitive use of the same rule, for $\mathring{t} \to \infty$ whose decisions are minimizing arguments in the formal analogy of (2.23)*

$$^{\lfloor\infty}\mathcal{V}(x_{t-1}) + {}^{\lfloor\infty}C = \min_{a_t \in a_t^*} \mathcal{E}\left[z(x_t, a_t) + {}^{\lfloor\infty}\mathcal{V}(x_t)|a_t, x_{t-1}\right]. \tag{2.33}$$

with a constant ${}^{\lfloor\infty}C \leq c$ and a time invariant Bellman function ${}^{\lfloor\infty}\mathcal{V}(x)$.

Proof: Let us take any finite horizon $\mathring{t}$ and, within this horizon, denote ${}^{\lfloor\mathring{t}}\tilde{\mathcal{V}}(\mathcal{P}_{a_t}) \equiv {}^{\lfloor\mathring{t}}\tilde{\mathcal{V}}(x_{t-1})$ the optimal loss-to-go, see Agreement 2.4.3.

Let us define ${}^{\lfloor\mathring{t}}C$ as the smallest constant such that ${}^{\lfloor\mathring{t}}\mathcal{V}(x_t) \equiv \tilde{\mathcal{V}}(x_t) - (\mathring{t} - t)\,{}^{\lfloor\mathring{t}}C$ is bounded from above for $\mathring{t} \to \infty$ and any fixed t, x_t. Obviously, the optimal strategy has to be can be stabilizing strategy. Thus, ${}^{\lfloor\mathring{t}}C \leq c$ and $\limsup_{\mathring{t}\to\infty} {}^{\lfloor\mathring{t}}C = {}^{\lfloor\infty}C$ exists.

The optimization is uninfluenced if we subtract the constant ${}^{\lfloor\mathring{t}}C$ from all partial losses. For arbitrary fixed t, x_t, the corresponding modified Bellman function ${}^{\lfloor\mathring{t}}\mathcal{V}(x(t))$, is bounded from above and ${}^{\lfloor\mathring{t}}\mathcal{V}(x(t)) =$

${}^{\lfloor \mathring{t}}\tilde{\mathcal{V}}(x(t)) - (\mathring{t} - t)\, {}^{\lfloor \mathring{t}}C$ = difference of a pair of monotonous sequences (indexed by $\mathring{t}$). Thus, a finite limit ${}^{\lfloor \infty}\mathcal{V}(x_t) = \lim_{\mathring{t}\to\infty} {}^{\lfloor \mathring{t}}\mathcal{V}(x_t)$ exists.

The modified Bellman function fulfills the equation

$$ {}^{\lfloor \mathring{t}}\mathcal{V}(x(t-1)) + {}^{\lfloor \mathring{t}}C = \min_{a_t \in a_t^*} \mathcal{E}\left[z(x_t, a_t) + {}^{\lfloor \mathring{t}}\mathcal{V}(x(t)) | a_t, x_{t-1} \right]. $$

Existence and finiteness of the involved $\limsup$ imply that the asymptotic version of the Bellman equation is fulfilled, too,

$$ {}^{\lfloor \infty}\mathcal{V}(x(t-1)) + {}^{\lfloor \infty}C = \min_{a_t \in a_t^*} \mathcal{E}\left[z(x_t, a_t) + {}^{\lfloor \infty}\mathcal{V}(x(t)) | a_t, x_{t-1} \right]. $$

Limits of ${}^{\lfloor \mathring{t}}\mathcal{V}(x_t)$ exist and, thus, $\limsup_{\mathring{t}\to\infty} {}^{\lfloor \mathring{t}}C = \lim_{\mathring{t}\to\infty} {}^{\lfloor \infty}C$.

The identical optimization is performed for each $t < \infty$. Thus, it provides the same decision rule for each t: the optimal strategy is a stationary one. □

Remark(s) 2.4.3

1. *The same proof is directly applicable to the fully probabilistic design as it can be seen as an instance of the additive loss function.*
2. *Solutions of the Bellman equation obtained for a growing finite horizon $\mathring{t}$ can be interpreted as successive approximations for solving its stationary counterpart (2.33).*
3. *So called* iterations in strategy space *[26] is an alternative and efficient way of finding the asymptotic solution. Essentially, a stabilizing stationary, i.e. repeating a single rule, strategy $\{\mathcal{R}\}$ is selected and the linear equation*
$$ \mathcal{V}(x) + C = \mathcal{E}[z(\tilde{x}, \mathcal{R}(x)) + \mathcal{V}(\tilde{x}) | \mathcal{R}(x), x] $$
is solved for the function $\mathcal{V}(\cdot)$ and constant C. Then, a new approximating strategy is found value-wise $\mathcal{R}(x) = \mathrm{Arg}\min_{a \in a^*} \mathcal{E}[z(\tilde{x}, a) + \mathcal{V}(\tilde{x}) | a, x]$ with such a $\mathcal{V}(\cdot)$. Under general conditions, the newly found strategy is stabilizing and iterations may be repeated till the guaranteed convergence. Details of this procedure are out of scope of this work but it should be considered when searching for efficient numerical procedures.

2.5 Learning

Generally, considered behavior $\mathcal{Q}^*$ contains quantities that are never observed directly and, in spite of this, we want to describe them or influence them. We called them internal quantities, see Agreement 2.4.2. Even in this case, the optimal decision-making needs the outer model (2.20), see Proposition 2.4.1, that should reflect them. Here we describe how to get it. The presented construction is of an independent interest as a good *formal model of learning.*

2.5.1 Bayesian filtration

This section deals with a general case when the construction of the outer model relies on filtering, i.e. learning of *time variant* internal, i.e. directly unobservable, quantities. It requires the following elements.

Requirement 2.5.1 (Models; natural conditions of decision-making)

1. *The innovations Δ_{a_t} are related to experience $\mathcal{P}_{a_t}$ and decisions a_t through the* observation model
$$ \{f(\Delta_{a_t} | a_t, \mathcal{P}_{a_t}, \Theta_t)\}_{t \in t^*} \tag{2.34} $$
that is given up to unknown internal quantities $\Theta_t \in \Theta_t^* \subset \mathcal{F}_{a_\tau}$, $\forall \tau \in t^*$.
2. *The evolution of the quantities $\Theta(\mathring{t}) \in \Theta^*(\mathring{t})$ is described by a known collection of pdfs called the* time evolution model
$$ \{f(\Theta_t | a_t, \mathcal{P}_{a_t}, \Theta_{t-1})\}_{t \in t^*}. \tag{2.35} $$

3. *The quantities $\Theta(\mathring{t})$ are unknown to the strategies considered. The* natural conditions of decision-making *(a slight generalization of natural conditions of control [14]) express it formally. They postulate independence of a_t and Θ_t when conditioned on $\mathcal{P}_{a_t}$*
$$f(a_t|\mathcal{P}_{a_t},\Theta_t) = f(a_t|\mathcal{P}_{a_t}) \qquad \underbrace{\Leftrightarrow}_{\textit{Proposition 2.3.4}} \qquad f(\Theta_t|a_t,\mathcal{P}_{a_t}) = f(\Theta_t|\mathcal{P}_{a_t}). \tag{2.36}$$
4. *The initial values of Θ_0, $\mathcal{P}_{a_1}$ add nothing new to the prior information so that the* prior pdf
$$f(\Theta_1) \equiv f(\Theta_1|a_1,\mathcal{P}_{a_1},\Theta_0) = f(\Theta_1|a_1,\mathcal{P}_{a_1}) \underbrace{=}_{(2.36)} f(\Theta_1|\mathcal{P}_{a_1}) \tag{2.37}$$
is the first term in the sequence of pdfs (2.35).

Remark(s) 2.5.1

1. *Often, the unknown quantities Θ_t together with the decision a_t are assumed to describe the involved conditional pdfs fully. Then, $\mathcal{P}_{a_t}$ can be omitted and Θ_t can be identified with the* information state.
2. *The natural conditions of decision-making express the assumption that $\Theta_t \notin \mathcal{P}_{a_\tau}\ \forall \tau \in t^*$. Thus, values of Θ_t cannot be used by the decision rules forming admissible strategy. Alternatively, see (2.36), we cannot gain information about Θ_t from the decision a_t if the corresponding innovation Δ_{a_t} (the corresponding reaction of the system) is not available.*

   The natural conditions of decision-making are "naturally" fulfilled by strategies we are designing. They have to be checked when data influenced by an "externally chosen" strategy are processed.

Proposition 2.5.1 (Generalized Bayesian filtering) *Let Requirement 2.5.1 be met. Then, the outer model of the system (2.20) is given by the formula*
$$f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t}) = \int_{\Theta_t^*} f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta_t) f(\Theta_t|\mathcal{P}_{a_t})\, d\Theta_t. \tag{2.38}$$
The evolution of the pdf $f(\Theta_t|\mathcal{P}_{a_t})$, called (generalized Bayesian) filtration of unknown quantities Θ_t, *is described by the following recursion that starts from the prior pdf $f(\Theta_1)$:*

- Data updating
$$f(\Theta_t|\mathcal{P}_{a_{t+1}}) = \frac{f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta_t) f(\Theta_t|\mathcal{P}_{a_t})}{f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})} \propto f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta_t) f(\Theta_t|\mathcal{P}_{a_t}) \tag{2.39}$$
  that incorporates the innovation Δ_{a_t} and the decision a_t, and
- Time updating
$$f(\Theta_{t+1}|\mathcal{P}_{a_{t+1}}) = \int_{\Theta_t^*} f(\Theta_{t+1}|a_{t+1},\mathcal{P}_{a_{t+1}},\Theta_t) f(\Theta_t|\mathcal{P}_{a_{t+1}})\, d\Theta_t \tag{2.40}$$
  that reflects the time evolution $\Theta_t \to \Theta_{t+1}$.

Proof: Sequential use of marginalization, chain rule, Proposition 2.3.4, and of the natural conditions of decision making (2.36) implies (2.38)
$$\begin{aligned} f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t}) &= \int_{\Theta_t^*} f(\Delta_{a_t},\Theta_t|a_t,\mathcal{P}_{a_t})\, d\Theta_t = \int_{\Theta_t^*} f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta_t) f(\Theta_t|a_t,\mathcal{P}_{a_t})\, d\Theta_t = \\ &= \int_{\Theta_t^*} f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta_t) f(\Theta_t|\mathcal{P}_{a_t})\, d\Theta_t. \end{aligned}$$
Data updating coincides with the Bayes rule in which the outer model of the strategy cancels as it does not depend on Θ_t due to the natural conditions of decision making (2.36).

Marginalization, chain rule and natural conditions of decision making imply also the formula for time updating. □

Agreement 2.5.1 (Filtering; predictive pdf) *The process of generating filtration is called (generalized Bayesian)* filtering. *The outer model of the system obtained by filtering is called* predictive pdf.

Remark(s) 2.5.2

1. *The term* predictive pdf *reflects the way how the outer model of the system has been obtained. It uses the observed experience and extrapolates it into ignorance assuming that the* mechanism of generating Θ_t does not change.

   This accumulation of experience and its extrapolation represent a good formal model of learning.

2. *It has to be stressed that the accumulation of experience can take place only when the rules governing the behavior are not changed during it, i.e. when we can rely on the validity of the underlying models.*

3. *The construction of the predictive pdf is our key motivation for filtering. Its results are often of an independent interest.*

4. *Under the natural conditions of decision-making, the filtering relies on the knowledge of decisions and not on the knowledge of rules $\mathcal{R} : \mathcal{P}_a^* \rightarrow a^*$ generating them. It is practically important when we learn while decision loop is closed, especially, when decisions are made by a human decision maker.*

5. *The time evolution model $f(\Theta_t|\Theta_{t-1}, a_t, \mathcal{P}_{a_t})$ as well as the observation model $f(\Delta_{a_t}|\Theta_t, a_t, \mathcal{P}_{a_t})$ have to result from a theoretical modelling of the system in question. Such a modelling uses both field knowledge, like laws of conservation, and approximation capabilities of the selected family of the models involved. Often, deterministic relationships are modelled and then the "deviations" from an "expected" trajectory are described.*

6. *The prior pdf $f(\Theta_1)$ allows us to introduce an information based on expert knowledge or analogy to situations observed previously.*

7. *The observed data, the only bridge to reality, enter the evaluations in the data updating step only when the newest innovation-decision pair is processed. This simple observation is important for approximation of the time evolution model, see Section 3.1.*

8. *In summary, the described Bayesian filtering combines prior information in $f(\Theta_1)$, theoretical knowledge of the specific fields described by $f(\Delta_{a_t}|\Theta_t, a_t, \mathcal{P}_{a_t})$, $f(\Theta_t|\Theta_{t-1}, a_t, \mathcal{P}_{a_t})$ and observed data $d((\mathring{t})) = (\Delta(\mathring{t}), a(\mathring{t}))$ by using coherent deductive rules of the calculus with pdfs.*

   This combination of information sources is a powerful internally consistent framework describing the essence of learning. Due to its deductive structure, the incorrect modelling not an incorrect information processing can be blamed for a failure of the specific learning process.

2.5.2 Bayesian estimation

This section deals with a special version of filtering called *estimation*. It arises when the internal quantities Θ_t are time invariant

$$\Theta_t = \Theta, \ \forall t \in t^*. \tag{2.41}$$

The common value Θ is called *parameter*. In this case, the time evolution model is $f(\Theta_t|\Theta_{t-1}, a_t, \mathcal{P}_{a_t}) = \delta(\Theta_t - \Theta_{t-1})$. The employed Dirac delta function $\delta(\cdot)$ is a formal pdf of the measure fully concentrated on the zero argument.

Proposition 2.5.2 (Generalized Bayesian estimation) *Let Requirement 2.5.1 be met with time invariant $\Theta_t = \Theta \in \Theta^* \subset \mathcal{F}_{a_\tau}, \forall \tau \in t^*$. Then, the outer model of the system (2.20) is given by the formula*

$$f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t}) = \int_{\Theta^*} f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t}, \Theta) f(\Theta|\mathcal{P}_{a_t})\, d\Theta. \tag{2.42}$$

The evolution of the pdf $f(\Theta|\mathcal{P}_{a_t})$*, called (generalized Bayesian)* parameter estimation *generating parameter estimate coinciding with the* posterior pdf *of* unknown parameter*, is described by the recursion identical with data updating (2.39)*

$$f(\Theta|\mathcal{P}_{a_{t+1}}) = \frac{f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta)f(\Theta|\mathcal{P}_{a_t})}{f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})} \propto f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta)f(\Theta|\mathcal{P}_{a_t}). \tag{2.43}$$

It starts from the prior pdf $f(\Theta) \equiv f(\Theta|\mathcal{P}_{a_1}, a_1) = f(\Theta|\mathcal{P}_{a_1})$.

The simplicity of the estimation formula allows us to write down its (non-recursive) batch variant

$$f(\Theta|\mathcal{P}_{a_{t+1}}) = \frac{\prod_{\tau\leq t} f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)f(\Theta)}{\int_{\Theta^*}\prod_{\tau\leq t} f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)f(\Theta)\,d\Theta} \equiv \frac{\mathcal{L}(\Theta,\mathcal{P}_{a_{t+1}})f(\Theta)}{\mathcal{I}(\mathcal{P}_{a_{t+1}})}. \tag{2.44}$$

The introduced likelihood function

$$\mathcal{L}(\Theta,\mathcal{P}_{a_{t+1}}) \equiv \prod_{\tau\leq t} f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta) \tag{2.45}$$

evolves according to the recursion identical with that for the posterior pdf (2.43) but it starts from the $\mathcal{L}(\Theta,\mathcal{P}_{a_1})$ *identically equal to 1.*

The normalization factor $\mathcal{I}(\cdot)$ *is defined by the formula*

$$\mathcal{I}(\mathcal{P}_{a_{t+1}}) = \int \mathcal{L}(\Theta,\mathcal{P}_{a_{t+1}})f(\Theta)\,d\Theta \propto f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t}). \tag{2.46}$$

With it, the outer model of the system (2.20) can alternatively be expressed

$$f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t}) = \frac{\mathcal{I}(\mathcal{P}_{a_{t+1}})}{\mathcal{I}(\mathcal{P}_{a_t})}. \tag{2.47}$$

Proof: It is again a simple exercise in calculus with pdfs, marginalization, chain rule and Bayes rule, Proposition 2.3.4 with the natural conditions of decision-making (2.36). □

Remark(s) 2.5.3

1. *The observation model* $f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta)$ *is called* parameterized model *whenever estimation problem is considered. We respect this tradition.*

2. *Note that the recursive evolution of the pdf* $f(\Theta|\mathcal{P}_{a_t})$ *allows us to interpret the posterior pdf as the prior one before processing new observations.*

3. *The data inserted into the "objective" parameterized (observation) model correct gradually the subjectively chosen prior pdf* $f(\Theta)$*. The posterior pdf* $f(\Theta|\mathcal{P}_{a_t})$ *reflects always both ingredients. If the data are informative enough, the relative contribution of the single subjective factor* $f(\Theta)$ *to the posterior pdf is decreasing with increasing* t *as the likelihood function* $\mathcal{L}(\Theta,\mathcal{P}_{a_{t+1}})$ *contains* t *"objective" factors, cf. (2.45).*

4. *Zero values are preserved by multiplication. Thus, the posterior pdf re-distributes probability mass only within the support of the prior pdf, i.e. within the set*

$$\mathrm{supp}\,[\,f(\Theta)] \equiv \{\Theta \in \Theta^* : \ f(\Theta) > 0\}.$$

   This fact allows us to introduce hard bounds on possible parameter values but prevents us to "learn" about parameters Θ *out of the support* $\mathrm{supp}\,[\,f(\Theta)]$*, see (2.44).*

5. *Remarks 2.5.2 related to filtering apply mostly to estimation, too. The parameter estimation is a task on its own; unknown parameters are always in the ignorance of the decision to be chosen; under the natural conditions of decision-making (2.36), decisions are needed and not the strategy* $\{\mathcal{R}_t : \mathcal{P}^*_{a_t} \to a^*_t\}_{t\in t^*}$ *generating them.*

6. *The parameters Θ_t are usually assumed to be finite-dimensional in order to avoid technicalities related to measure theory. In exceptional cases, like in description of so called equivalence approach, see Section 3.4, we deal with* potentially infinite-dimensional parameter. *It means that the number of unknown quantities is finite but increases without limitations. This case is often called* non-parametric estimation.

2.5.3 Asymptotic of estimation

The analysis outlined here serves us mainly for interpretation purposes. Thus, all technicalities are suppressed as much as possible.

The "objective" pdf $f(\mathcal{Q})$, see Section 2.3.1, describing the system behavior is denoted here ${}^{\lfloor o}f(\mathcal{Q})$. The corresponding outer model of the system $f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})$ is denoted ${}^{\lfloor o}f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})$. Its relationship to the predictive pdf $f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})$ obtained through the parameter estimation, Proposition 2.5.2, is inspected.

For the analysis, the notion of *entropy rate* $\mathcal{H}_\infty\left({}^{\lfloor o}f||\Theta\right)$ is needed. For each $\Theta\in\Theta^*$, it is defined by the formula

$$\mathcal{H}_\infty\left({}^{\lfloor o}f||\Theta\right) \equiv \lim_{t\to\infty}\sup \mathcal{H}_t\left({}^{\lfloor o}f||\Theta\right) \equiv \tag{2.48}$$
$$\equiv \lim_{t\to\infty}\sup\frac{1}{t}\sum_{\tau\le t}\int_{\Delta^*_{a_\tau}} {}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})\ln\left(\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}\right)d\Delta_{a_\tau}.$$

Proposition 2.5.3 (Asymptotic of estimation) *Let the natural conditions of decision-making (2.36) hold and $0<\underline{C}_\Theta\le\overline{C}_\Theta\le c<\infty$, $\bar{t}_\Theta\in\{1,2,\ldots\}$ exist, for almost all $\Theta\in\Theta^*$, such that*

$$\underline{C}_\Theta f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta)\le {}^{\lfloor o}f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t})\le\overline{C}_\Theta f(\Delta_{a_t}|a_t,\mathcal{P}_{a_t},\Theta),\ \forall t>\bar{t}_\Theta,\ \forall\mathcal{P}_{a_{t+1}}\in\mathcal{P}^*_{a_{t+1}}. \tag{2.49}$$

Then, the posterior pdf $f(\Theta|\mathcal{P}_{a_t})$ (2.43) converges almost surely to a pdf $f(\Theta|\mathcal{P}_{a_\infty})$. It has the support coinciding with the set of minimizing arguments in

$$\mathrm{supp}\left[f(\Theta|\mathcal{P}_{a_\infty})\right]=\mathrm{Arg}\min_{\Theta\in\mathrm{supp}[f(\Theta)]\cap\Theta^*}\mathcal{H}_\infty\left({}^{\lfloor o}f||\Theta\right). \tag{2.50}$$

Proof: Under the natural conditions of decision-making (2.36), the posterior pdf (2.43) can be written in the form

$$f(\Theta|\mathcal{P}_{a_{t+1}}) \propto f(\Theta)\exp\left[-t\mathcal{H}(\mathcal{P}_{a_{t+1}},\Theta)\right]\quad\text{with} \tag{2.51}$$
$$\mathcal{H}(\mathcal{P}_{a_{t+1}},\Theta) = \frac{1}{t}\sum_{\tau\le t}\ln\left[\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)}\right]. \tag{2.52}$$

This form exploits the fact that non-normalized posterior pdf can be multiplied by any factor independent of Θ.

Let us fix the argument $\Theta\in\Theta^*$ and define

$$e_{\Theta;\tau}\equiv\ln\left[\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)}\right]-{}^{\lfloor o}\mathcal{E}\left[\ln\left[\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)}\right]|a_\tau,\mathcal{P}_{a_\tau}\right]$$
$$\equiv \ln\left[\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)}\right]-\int {}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})\ln\left[\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)}\right]d\Delta_{a_\tau}.$$

A direct check reveals that the introduced deviations $e_{\Theta;\tau}$ of values

$$\ln\left[\frac{{}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})}{f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau},\Theta)}\right]$$

from their conditional expectations ${}^{\lfloor o}\mathcal{E}[\cdot|a_\tau,\mathcal{P}_{a_\tau}]$, given by ${}^{\lfloor o}f(\Delta_{a_\tau}|a_\tau,\mathcal{P}_{a_\tau})$, are zero mean and mutually non-correlated. With them,

$$\mathcal{H}(\mathcal{P}_{a_{t+1}},\Theta)=\mathcal{H}_t\left({}^{\lfloor o}f||\Theta\right)+\frac{1}{t}\sum_{\tau\le t}e_{\Theta;\tau}.$$

The assumption (2.49) implies that the variance of $e_{\Theta;\tau}$ is bounded. Consequently, the last term in the above expression converges to zero almost surely (a.s.), see [27], page 417. The first term on the right hand side of the last equality is non-negative as it can be viewed as a sum of Kullback-Leibler distances, see Proposition 2.4.3. Due to (2.49), it is also finite. Thus, (2.52) converges a.s. to the non-negative value $\mathcal{H}\left({}^{\llcorner o}f || \Theta \right)$. The posterior pdf remains unchanged if we subtract $t \min_{\Theta \in \text{supp}[f(\Theta)] \cap \Theta^*} \mathcal{H}_\infty \left({}^{\llcorner o}f || \Theta \right)$ from the exponent of its non-normalized version (2.51). Then, the exponent contains ($-t\times$ an_asymptotically_non-negative_factor). Thus, the posterior pdf $f(\Theta|\mathcal{P}_{a_\infty})$ may be asymptotically non-zero on minimizing arguments (2.50) only. □

Remark(s) 2.5.4

1. *The entropy rate can be seen as an extension of the Kullback-Leibler distance (2.26) that covers well asymptotic and controlled cases. It coincides with the Kullback-Leibler distance in a range of particular cases.*

2. *The assumption (2.49) can be weakened. It is, however, intuitively well acceptable. It excludes parameterized models, which assign zero belief to data generated by the system with a non-zero probability and vice versa.*

3. *The Bayesian estimation minimizes the asymptotic entropy rate of the objective pdf ${}^{\llcorner o}f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t})$ and a model chosen among candidates $f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t}, \Theta)$, $\Theta \in \Theta^*$. In other words, a* best projection *of the objective pdf to the considered parameterized models is asymptotically found. The prior pdf can be interpreted as a prior belief assigned to the individual parameters $\Theta \in \Theta^*$ that the corresponding parameterized model is the best projection of the objective pdf [28]: not knowing the reality we do not know the best projection we finally arrive at.*

4. *The posterior pdf concentrates on a point if there is a unique minimizer of the entropy rate. In this case, the model is called* identifiable. *The possibility to identify the model can be influenced by*

   - *the considered class of the parameterized models and*
   - *the decisions chosen, for instance, by the controller used: e.g. controller generating constant inputs does not allow us to learn their dynamic influence on outputs.*

5. *If the objective pdf ${}^{\llcorner o}f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t})$ coincides with $f(\Delta_{a_t}|a_t, \mathcal{P}_{a_t}, \Theta)$ for $\Theta = {}^{\llcorner o}\Theta$ with $f\left({}^{\llcorner o}\Theta\right) > 0$ then ${}^{\llcorner o}\Theta$ is in the support of the asymptotic posterior pdf $f(\Theta|\mathcal{P}_{a_\infty})$. If, moreover, the model is identifiable the objective pdf is asymptotically identified by the adopted Bayesian approach. This fact can be expressed in a more appealing form:*

   Bayesian estimate is consistent whenever there is a consistent estimator.

6. *Often, a similar analysis is performed by measuring distance of parameterized models to empirical pdf of data [29]. It gives similar answers if the empirical pdf converges to the objective pdf. Moreover, it provides hints how to approximate the posterior pdf [30], see also Section 3.4. On the other hand, the known conditions of such convergence are more restrictive. For instance, an analysis of the controlled case is much harder.*

Problem 2.5.1 (How to unify statistics?) *Asymptotic analysis and finite-data oriented Bayesian approach are often perceived in an antagonistic way. Their simultaneous and harmonized use still waits for its full exploitation.*

Chapter 3

Approximate and feasible learning

Mixture estimation relies on our ability to solve estimation and prediction tasks for various parameterized models. Here, general tools are prepared for this task. Specifically, estimation with forgetting is recalled that allows us to track slow changes of parameters and thus to make the advisory system adaptive. Then, the estimation of so called *exponential family* of parameterized models is recalled. Its most important special instances *normal parameterized models* and *Markov chain parameterized models* are treated in independent Chapters 8, 10. The special case of *uniform parameterized models* that provides Bayesian interpretation of the important mean tracking (MT) algorithm [11] is dealt with in Chapter 12 and its normal extension is used in Chapter 13.

This chapter also presents so called equivalence approach to parameter estimation [30]. It provides a promising tool for handling non-linear and/or non-Gaussian factors.

3.1 Estimation with forgetting

Advisory system we are constructing relies on validity of the model learnt. We have to grasp all significant time-varying relationships. At the same time, we cannot exclude slow changes caused, for instance, by aging of the maintained process. Thus, it is reasonable to inspect the case of *slowly varying parameters* as a widely met intermediate case bridging estimation and filtering. It admits time variations of Θ_t, but it assumes that $\Theta_{t+1} \approx \Theta_t$. At the same time, a more precise model of changes is not used.

We summarize an approach called *stabilized forgetting*, [31]. It is based on a flexible problem formulation of this formally ill posed problem.

Let $f(\Theta_{t+1} = \Theta_t = \Theta|d(t))$ be the pdf that assumes no parameter changes happened after measuring d_t and before processing d_{t+1}. Let $f_A(\Theta_{t+1} = \Theta|d(t))$ be an *alternative pdf* that describes parameters after some changes. Let $\lambda \in [0,1]$ be the probability that the "correct" unknown pdf ${}^{\llcorner o}f(\Theta_{t+1} = \Theta|d(t))$ has the best projection equal to the former pdf and $1-\lambda$ the probability that the latter one is relevant. We are searching for the best compromise $f(\Theta_{t+1} = \Theta|d(t))$ minimizing its expected KL distance to the pdf ${}^{\llcorner o}f(\Theta_{t+1} = \Theta|d(t))$. It is described by the following proposition.

Proposition 3.1.1 (Geometric representation of a pair of pdfs) *Let an unknown pdf* ${}^{\llcorner o}f \in f^* \equiv \{f_1, f_2\}$ *be equal to* $f_1(x)$ *with a probability* $\lambda \in \lambda^* \equiv [0,1]$ *and equal to* $f_2(x)$ *with the complementary probability* $1-\lambda$*. The pdfs* f_1, f_2 *are supposed to have a common support* x^**. Then, the pdf*

$$ {}^{\llcorner o}\hat{f}(x) \propto f_1^{\lambda}(x) f_2^{1-\lambda}(x), \; x \in x^* \quad \text{is the estimate of } {}^{\llcorner o}f(\cdot) \text{ that} \tag{3.1}$$

- *uses experience* $\mathcal{P}_{\hat{f}} \equiv \{\lambda, f_1, f_2\}$ *and*
- *minimizes the expected KL distance (2.26), i.e. the functional*

$$\mathcal{E}\left[\mathcal{D}\left(\hat{f}||{}^{\llcorner o}f\right)\right] \equiv \lambda \mathcal{D}\left(\hat{f}||f_1\right) + (1-\lambda)\mathcal{D}\left(\hat{f}||f_2\right). \tag{3.2}$$

The reached minimum is

$$\omega(\lambda) \equiv \mathcal{E}\left[\mathcal{D}\left({}^{\llcorner o}\hat{f}||f\right)\right] = -\ln \int_{x^*} f_1^{\lambda}(x) f_2^{1-\lambda}(x)\, dx. \tag{3.3}$$

If $f_1 \not\equiv f_2$, the function $\omega(\lambda)$ reaches its maximum on $(0,1)$.

Proof: The minimized functional (3.2) can be re-written into the form

$$\mathcal{D}\left(\hat{f}||\,{}^{\lfloor o}\hat{f}\right) - \ln \int_{x^*} f_1^{\lambda}(x) f_2^{1-\lambda}(x)\, dx.$$

This form and properties of the KL distance imply the form of the minimizer (3.1) as well as the attained minimum value (3.3).

The last statement of Proposition is implied by the assumption $f_1 \not\equiv f_2$, closeness of the set λ^* and continuity of $\omega(\lambda)$ guaranteed by the common support of f_1, f_2. Moreover, non-negativity of the KL distance and obvious equalities $\omega(0) = \omega(1) = 0$ imply that the extreme in (0,1) must be maximum. □

The direct application of this proposition gives

$$f(\Theta_{t+1} = \Theta|d(t)) \propto [f(\Theta_{t+1} = \Theta_t = \Theta|d(t))]^{\lambda} [f_A(\Theta_{t+1} = \Theta|d(t))]^{1-\lambda}. \quad (3.4)$$

By using this formula, we approximate the time updating step (2.40) in filtering without explicitly specifying the model of time evolution (2.35).

Algorithm 3.1.1 (Stabilized forgetting)
Initial (off line) mode

- *Specify prior pdf $f(\Theta_1 = \Theta) \equiv f(\Theta_1 = \Theta|d(0))$ corresponding to the treated parameterized model $f(\Delta_t|a_t, d(t-1), \Theta_t)$.*
- *Select the probability $\lambda \in [0,1]$ that parameters do not change.*

Sequential (on line) mode, *running for $t \in t^*$,*

1. *Collect the newest data d_t.*
2. *Perform data updating $f(\Theta_t = \Theta|d(t)) \propto f(\Delta_t|a_t, d(t-1), \Theta) f(\Theta_t = \Theta|d(t-1))$.*
3. *Select or update the alternative pdf $f_A(\Theta_{t+1} = \Theta|d(t))$.*
4. *Approximate time updating (*forget*)*

$$f(\Theta_{t+1} = \Theta|d(t)) \propto [f(\Theta_{t+1} = \Theta_t = \Theta|d(t))]^{\lambda} [f_A(\Theta_{t+1} = \Theta|d(t))]^{1-\lambda}.$$

Remark(s) 3.1.1

1. *The pdf (3.4) after the time updating is a compromise between the posterior pdf obtained under the hypothesis that Θ_t is time invariant and an externally supplied alternative f_A. The closer λ is to unity, the slower changes are expected, i.e. the higher weight the posterior pdf corresponding to the time invariant case gets.*
2. *Note that the forgetting operation (3.4) preserves the basic property of time updating: the posterior pdf on parameters propagates without obtaining any new measured information.*
3. *Let us assume that $f_A \propto 1$ and $\lambda < 1$. Then, $f(\Theta_{t+1} = \Theta|d(t)) \propto [f(\Theta_{t+1} = \Theta_t = \Theta|d(t))]^{\lambda}$, i.e. the pdf after time updating is a* flattened version *of the pdf obtained after data updating. It is intuitively appealing as our uncertainty about parameters can hardly decrease without knowing a good time evolution model (2.35) and with no new information processed.*
4. *It is instructive to inspect the influence of forgetting on old data built in through the observation model. The older data are, the stronger flattening is applied to the corresponding model. Consequently, the older data influence the estimation results less than new ones. Data are gradually "forgotten". This explains why the probability λ is called* forgetting factor.

5. *The alternative pdf* $f_A(\cdot)$ *assigned to* Θ^* *expresses our belief where the parameters might move within the time interval* $[t, t+1)$ *while we have no new observable information.*

   Often, the pessimistic uniform alternative pdf ($\propto 1$*) has been used. This special case of stabilized forgetting is called* exponential forgetting. *It allows us to follow relatively fast parameter changes but it forgets the accumulated information with, often too high, exponential rate. For this reason, it is worth to preserve what we feel as a guaranteed information. The prior pdf* $f(\Theta_{t+1} = \Theta)$ *is a typical, reasonably conservative, choice of the alternative pdf* $f_A(\Theta_{t+1} = \Theta|d(t))$.

6. *The non-trivial alternative pdf prevents us to forget the "guaranteed" information as it is always incorporated after flattening (exponential forgetting). This stabilizes whole learning and reflects very positively in its numerical implementations. Without this, the posterior pdf may become too flat whenever the information brought by new data is not sufficient.*

   Note that lack of information brought by new data is more rule than exception. It is true especially in so called regulation problem. In it, the controller tries to make closed-loop as quiet as possible, it tries to suppress any new information brought by data.

7. *The forgetting factor* λ *can be either taken as tuning knob or estimated. The predictive pdf parameterized by it, however, depends on it in a very complex way so that a partitioned estimation has to be applied when its posterior pdf is estimated on a pre-specified grid [32].*

   Alternatively, the forgetting factor can be chosen in a pessimistic way as the maximizer of the reached minima, see Proposition 3.2.1.

8. *Practical importance of this particular case of slowly varying parameters cannot be over-stressed: the vast majority of adaptive systems, rely on a version of forgetting.*

3.2 Exponential family

Agreement 3.2.1 (Exponential family of parameterized factors) *The* i*-th parameterized factor belongs to the dynamic exponential family iff it can be written in the form*

$$f(d_{i;t}|d_{i+1;t}, \ldots, d_{\mathring{d};t}, d(t-1), \Theta) \equiv f(d_{i;t}|\psi_{i;t}, \Theta) = A(\Theta)\exp[\langle B(\Psi_{i;t}), C(\Theta)\rangle + D(\Psi_{i;t})], \quad \textit{where} \quad (3.5)$$

the functional $\langle \cdot, \cdot \rangle$*, linear in the first argument, is defined (within this text)*

$$< x, y >= \begin{cases} x'y & \text{if } x,\, y \text{ are vectors } ' \text{ is transposition} \\ \mathrm{tr}[xy] & \text{if } x,\, y \text{ are matrices, } \mathrm{tr} \text{ is trace} \\ \sum_{i \in i^*} x_i y_i & \text{if } x,\, y \text{ are arrays with a multi-index } i, \end{cases} \quad (3.6)$$

$\Psi'_{i;t} \equiv [d_{i;t}, \psi'_{i;t}]$ *is a finite dimensional* data vector *determined by* $d_{i+1;t}, \ldots, d_{\mathring{d};t}, d(t-1)$ *whose values can be recursively updated according to a known rule* $(\Psi_{i;t-1}, d_t)^* \rightarrow \Psi^*_{i;t}$,

$A(\cdot)$ *is a non-negative scalar function defined on* Θ^*,

$B(\cdot), C(\cdot)$ *are either vector or matrix functions of compatible, finite and fixed dimensions. They are defined on respective arguments in* $\Psi^*_{i;t}$ *and* Θ^*,

$D(\cdot)$ *is a non-negative scalar function defined on* Ψ^*_i.

Remark(s) 3.2.1

1. *Our definition of the exponential family contains the non-standard requirement on the recursive updating of the data vector* $\Psi_{i;t}$*. It is practically important for dynamic cases we deal with.*

2. *Notice that equality is used in (3.5). The normalization of this pdf must not spoil this form. It makes the allowed form rather restrictive. In the dynamic case with a nonempty regression vector* ψ*, normal (Gaussian) linear-in-parameters factors and Markov chains almost cover exponential family.*

3. *The scalar function D entering (3.5) does not influence estimation. The factor* $\exp(D(\Psi_{i;t}))$ *enters prediction unchanged. We use this property only in Section 8.1.6. Otherwise,* $D(\Psi_{i;t})$ *is dropped.*

4. *Further on, we consider factors related to a predicted data entry $d_{i;t}$ with a fixed i. Thus, we can drop this index within this Chapter.*

Practical significance of the exponential family becomes obvious when we specialize to it Proposition 2.5.2 describing the corresponding estimation and prediction.

Proposition 3.2.1 (Estimation and prediction in exponential family) *Let natural conditions of decision-making, Requirement 2.5.1, be met with time invariant $\Theta_t = \Theta \in \Theta^*$. Let the parameterized model belong to the exponential family (3.5). Then, the* predictive pdf, *the outer model of the system, is given by the formula*

$$f(\Delta_t|a_t, d(t-1)) = \frac{\mathcal{I}(V_{t-1} + B(\Psi_t), \nu_{t-1}+1)}{\mathcal{I}(V_{t-1}, \nu_{t-1})} \exp[D(\Psi_t)], \ \textit{with} \tag{3.7}$$

$$V_t = V_{t-1} + B(\Psi_t),\ V_0 = 0;\ \ \nu_t = \nu_{t-1}+1,\ \nu_0 = 0, \quad \textit{and} \tag{3.8}$$

$$\mathcal{I}(V,\nu) = \int_{\Theta \in \Theta^*} A^{\nu}(\Theta) \exp[\langle V, C(\Theta)\rangle] f(\Theta)\chi(\Theta)\, d\Theta, \ \textit{where} \tag{3.9}$$

$f(\Theta)$ is a prior pdf. Its support is restricted by the indicator $\chi(\cdot)$ of the set Θ^* .

The Bayesian parameter estimate (posterior pdf) is

$$f(\Theta|d(t)) = \frac{A^{\nu_t}(\Theta)\exp[\langle V_t, C(\Theta)\rangle]\chi(\Theta) f(\Theta)}{\mathcal{I}(V_t,\nu_t)}, \tag{3.10}$$

i.e. the likelihood function is

$$\mathcal{L}(\Theta, d(t)) \equiv \mathcal{L}(\Theta, V_t, \nu_t) = A^{\nu_t}(\Theta)\exp[\langle V_t, C(\Theta)\rangle]. \tag{3.11}$$

Let us consider the conjugate prior *pdf*

$$f(\Theta) \propto A^{\nu_0}(\Theta)\exp[\langle V_0, C(\Theta)\rangle]\chi(\Theta), \tag{3.12}$$

determined by the "prior statistics" V_0, ν_0. Then the prediction and estimation formulas (3.7), (3.10) are valid if

- *V_0, ν_0 replace the zero initial conditions in (3.8),*
- *the indicator $\chi(\cdot)$ is formally used as the prior pdf.*

Let us allow slow parameter changes with forgetting factor $\lambda \in [0,1]$ and the alternative pdf given in the conjugate form determined by the pair of sufficient statistics $V_{A;t}, \nu_{A;t}$.

Then, the prediction and estimation formulas remain unchanged with statistics evolving according to the recursion

$$\begin{aligned} V_t &= \lambda(V_{t-1} + B(\Psi_t)) + (1-\lambda)V_{A;t},\ V_0 \ \textit{given} \\ \nu_t &= \lambda(\nu_{t-1}+1) + (1-\lambda)\nu_{A;t},\ \nu_0 \ \textit{given.} \end{aligned} \tag{3.13}$$

Remark(s) 3.2.2

1. *Estimation and prediction within the exponential family is very simple, especially with the conjugate prior pdf. The updating of functions (pdfs) converts into the algebraic recursive updating of the finite-dimensional sufficient statistic consisting of V_t and the* sample counter *ν_t. Moreover, a single type of the normalization integral $\mathcal{I}(V,\nu)$ has to be evaluated.*

   The need to have the complete recursion *explains the requirement for possibility to update Ψ_t recursively, see Agreement 3.2.1.*

2. *An inspection whether there is a wider set of parameterized models with advantageous properties of the exponential family opens just a narrow space [33, 34]. Essentially, the exponential family coincides with all parameterized models that are sufficiently smooth functions of Θ and with supports independent of Θ. Uniform distribution with unknown constant boundaries represents one of a few feasible examples of pdfs out of the exponential family.*

3. *The class of models that lead to a finite-dimensional characterization of pdfs occurring in filtering is even more restrictive. Its discussion can be found in [35].*

3.3 Structure estimation in nested exponential family

Usually, specification of the parameterized model can be done by selecting discrete-valued pointers to a finite set of alternative parameterized models. These pointers define *model structure*, see [36] for a detailed discussion. Formally, estimation of the best structure can be performed fully within Bayesian set-up by assigning prior pf to them and computing the posterior one. The problem becomes specific and difficult due to the usual extremely large cardinality of the set of possible structures. Then, special measures have to be taken. Often, we deal with nested models with sufficient statistics contained in a single one corresponding to the *richest structure*.

Proposition 3.3.1 (Nesting in exponential family) *Let the parameterized model with the* richest structure *belong to the exponential family (3.5)* $f(d|\psi_r,\Theta_r) = A_r(\Theta_r)\exp[\langle B_r(\Psi_r), C_r(\Theta_r)\rangle]$. *Let us consider another model* $f(d|\psi,\Theta) = A(\Theta)\exp[\langle B(\Psi), C(\Theta)\rangle]$ *describing the same data* $d(\mathring{t})$.

Let N be a time invariant linear nesting operator *such that* $B(\Psi(d(t))) \equiv N[B_r(\Psi_r(d(t))]$.

Let us consider conjugate prior pdfs given by statistics $V_{r;0}, \nu_{r;0}$ and $V_0 \equiv N[V_{r;0}], \nu_0$.

Then, the V statistics of the posterior pdfs of both models are related by the nesting mapping $V_{\mathring{t}} = N[V_{r;\mathring{t}}]$ and the pdf of describing the predicted data

$$f(d(\mathring{t})) \propto \frac{\mathcal{I}(V_{\mathring{t}},\nu_t)}{\mathcal{I}(V_0,\nu_0)} = \frac{\mathcal{I}\left(N[V_{r;\mathring{t}}],\nu_t\right)}{\mathcal{I}\left(N[V_{r;0}],\nu_0\right)}. \tag{3.14}$$

Proof: It is implied directly by Proposition 3.2.1 and linearity of N. □

For structures $s \in s^*$, that can be obtained by a nesting operator from the richest structure, we are able to evaluate the pdf $f(d(\mathring{t})|s)$ for any specific structure s. Let $f(s)$ be prior pf on s^*, then the Bayes rule, Proposition 2.3.4, gives us formally the posterior pf on structures, i.e. full information needed for its point estimation.

If ψ_r is *richest regression vector* then the number $\mathring{s}$ of possible competitive structures nested in it is $2^{\psi_r^*}$. This is mostly an excessive number that prevents us to evaluate completely posterior probabilities of nested structures. Instead of it, we are searching for *maximum a posterior probability* (MAP) estimate. Of course, we can inform also about highly probable structures met during the search for MAP estimate. The following conceptual algorithm is used for it.

Algorithm 3.3.1 (MAP estimate of a factor structure)
Do while value the pre-specified number of restarts is not exceeded.

1. *Select an initial guess of the structure.*
2. *Do while the value of the likelihood increases and a pre-specified number of searches is not exceeded.*
   (a) *Make full search for the best structure within a "neighborhood" of the current guess of the structure, i.e. maximize the posterior likelihood $f(d(\mathring{t})|s)f(s)$ within it.*
   (b) *Take the maximizer as a new guess of the structure.*

The algorithm is "parameterized" by

- the generator of the initial guesses: a specified number of random choices and/or empty, richest and user-specified regression vectors are used;
- definition of neighborhood: a good choice depends on the specific parameterized model considered and as such it is discussed in the corresponding subsections.

3.4 Equivalence approach

The exponential family and special uniform distributions provide us a basic supply of factors. Sometimes, however, we are forced to go out of this family. Under natural conditions of decision-making (2.36), the generalized Bayesian estimation (Proposition 2.5.2) updates the posterior pdfs according to the Bayes rule (2.43)

$$f(\Theta|d(t)) = \frac{f(\Delta_t|a_t, d(t-1), \Theta) f(\Theta|d(t-1))}{f(\Delta_t|a_t, d(t-1))}, \ t \in t^*. \tag{3.15}$$

Complexity of these pdfs increases quickly with an increasing amount of data, with increasing t. The exponential family (3.5) is essentially the only exception from this rule. This section tries to cope with *recursive estimation applicable out of exponential family.* The equivalence approach presented in this section allows us to find a well justified approximation even in these cases.

3.4.1 Recursively feasible representation

Always limited capabilities of computers call for a reduced representation of the propagated posterior pdfs. It is a peculiar task as the posterior pdfs concentrate quickly on a very narrow support at a priori unknown position in Θ^*. Thus, a representation on a sufficiently fine grid that does not miss the final position becomes soon computationally prohibitive. The way out has been elaborated in a sequence of papers and summarized in [30]. Here, we just outline the essence of this *equivalence approach.*

Proposition 3.4.1 (Equivalence-preserving mapping) *Let the posterior pdf $f(\Theta|d(t-1))$ $\in f^*(\Theta|d(t-1)) \equiv$ a set of pdf with a common, time, data and parameter invariant support. Let the mapping*

$$\mathcal{G}_{t-1} : f^*(\Theta|d(t-1)) \to g^*_{t-1} \tag{3.16}$$

assign to each element of $f^*(\Theta|d(t-1))$ a finite-dimensional vector statistics $g_{t-1} \equiv g(d(t-1))$ representing it.

Then, the values of g_{t-1} can be exactly recursively updated *using only its previous value and the current parameterized model $f(\Delta_t|a_t, d(t-1), \Theta)$ iff $\mathcal{G}_t$ is a* time-invariant linear mapping $\mathcal{G} \equiv \mathcal{G}_t$, $t \in t^*$, *acting on logarithms of the pdfs involved. The logarithmic pdfs are treated as functions of Θ.*

$\mathcal{G}_t$ has to map Θ-independent elements to zero.

Proof: To demonstrate necessity is rather hard and the interested reader is referred to [37, 38]. To show that the conditions on $\mathcal{G}_t \equiv \mathcal{G}$, $t \in t^*$, are sufficient is simple and instructive. They become obvious if we apply $\mathcal{G}$ to the logarithmic version of the Bayes rule (3.15) and use both time-invariance and linearity of G. The normalizing term $\ln(f(\Delta_t|a_t, d(t-1))$ is independent of Θ and as such mapped to zero. The recursion for g_t is then simply

$$g_t = \mathcal{G}\left[\ln\left(f(\Delta_t|a_t, d(t-1), \Theta)\right)\right] + g_{t-1}, \ \text{ with } g_0 = \mathcal{G}(\ln(f(\Theta))) \equiv \mathcal{G}(\ln(\text{prior pdf})). \tag{3.17}$$

□

Note that (3.17) becomes the true recursion if we need not store whole past observed data for evaluating $f(\Delta_t|a_t, d(t-1), \Theta)$. Thus, similarly as in the case of exponential family, see Agreement 3.2.1, we restrict ourselves to such models. Together with their formal description we introduce other notions useful in the subsequent discussion.

Agreement 3.4.1 (Finite memory; Empirical pdf; Riezs representation) *The parameterized model is said to have* finite memory *iff*

$$f(\Delta_t|a_t, d(t-1), \Theta) \equiv M(\Theta, \Psi_t), \ \textit{where} \tag{3.18}$$

$M(\cdot,\cdot)$ is a known function of Θ and of a finite dimensional data vector Ψ_t that can be updated recursively

$$(\Psi_{t-1}, d_t)^* \to \Psi^*_t.$$

Let $\{\Psi_\tau\}_{\tau=1}^{t}$ *be measured data vectors. Then,*

$$f_t(\Psi) \equiv \frac{1}{t}\sum_{\tau=1}^{t} \delta(\Psi - \Psi_\tau), \;\; \Psi \in \Psi^* \equiv \bigcup_{t \in t^*} \Psi_t^* \tag{3.19}$$

is the formal empirical pdf *of* Ψ*. The used Dirac delta function* $\delta(\cdot)$ *is the linear functional assigning to (reasonable) functions* $B(\Psi)$ *their values at zero argument. It has formal Riezs integral representation, [39],*

$$\int_{\Psi^*} B(\Psi)\delta(\Psi)\, d\Psi = B(0), \;\; \textit{where} \tag{3.20}$$

$\delta(\Psi)$ *has to be taken as generalized function [40].*

We assume that the same representation exists for the linear mapping $\mathcal{G}$ *introduced in Proposition 3.4.1, i.e. there is a, possibly generalized, vector function* $G(\Theta)$ *such that*

$$\mathcal{G}(C) \equiv \int_{\Theta^*} C(\Theta)G(\Theta)\, d\Theta \tag{3.21}$$

for any considered function $C : \Theta^* \to (-\infty, \infty)$*.*

Assuming (3.18), using the empirical pdf (3.19) and Riezs representation of G (3.21), we see that the representation g_t of the posterior pdf $f(\Theta|d(t))$ can be written in the form

$$g_t = t \int_{\Psi^*} \left[\int_{\Theta^*} \ln[M(\Theta, \Psi)]G(\Theta)\, d\Theta\right] f_t(\Psi)\, d\Psi + g_0. \tag{3.22}$$

The integral in brackets defines the vector function

$$h(\Psi) \equiv \int_{\Theta^*} \ln[M(\Theta, \Psi)]G(\Theta)\, d\Theta. \tag{3.23}$$

With it, we get the equivalent form of (3.22)

$$g_t = t \int_{\Psi^*} h(\Psi) f_t(\Psi)\, d\Psi + g_0. \tag{3.24}$$

Recall that left hand side of (3.24) is known as it can be update recursively according to (3.17) which has the equivalent “Riezs” form

$$g_t \underbrace{=}_{(3.17)} \int_{\Theta^*} \ln[M(\Theta, \Psi_t)]G(\Theta)\, d\Theta + g_{t-1} \underbrace{\equiv}_{(3.23)} h(\Psi_t) + g_{t-1}, \;\; g_0 = \mathcal{G}(\ln(f(\Theta))). \tag{3.25}$$

Also, for the chosen parameterized model (3.18) and functions $G(\Theta)$ representing the admissible projections to g_t^*, the vector function $h(\Psi)$ (3.23) is known as well as its values in measured data vectors Ψ_t.

3.4.2 Approximation as a point estimation

The posterior pdf, we are interested in, can be expressed in terms of empirical pdf as follows, cf. similar transformation in connection with Proposition 6.4.10,

$$f(\Theta|d(t)) \equiv f(\Theta|f_t(\cdot)) \propto f(\Theta) \exp\left[t \int_{\Psi^*} \ln[M(\Theta, \Psi)] f_t(\Psi)\, d\Psi\right]. \tag{3.26}$$

The empirical pdf is unknown to us as we are not able store it and reduce it to the vector g. Thus, this posterior pdf is unknown, too. We want estimate it, we are facing the decision-making problem.

The selection of a suitable estimate $\hat{f}(\Theta|g_t)$($\equiv$ a pdf acting on Θ^*) fits to our general decision-making framework as follows.

- $\mathcal{Q} \equiv (\mathcal{P}_a, a, \mathcal{F}_a) \equiv \left(g_t, \hat{f}(\Theta|g_t), f(\Theta|f_t(\cdot))\right) \equiv$
(stored statistic, posterior-pdf estimate, posterior pdf given by the unknown empirical pdf $f_t(\Psi),\ \Psi \in \Psi^*$).

- Admissible decision rules are of the form

$$\mathcal{R}_t : g_t^* \to \hat{f}^* \equiv \left\{ \hat{f}(\Theta) : \int \hat{f}(\Theta)\, d\Theta = 1,\ \hat{f}(\Theta) \geq 0,\ \forall \Theta \in \Theta^* \right\}.$$

Note that g_0 is a known fixed representation of the known prior pdf $f(\Theta)$ and as such neither of them is explicitly mentioned as a part of the domain of $\mathcal{R}_t$.

- Loss function is selected as the KL distance (2.26) $\mathcal{D}(\hat{f}(\Theta)||f(\Theta|f_t))$.

Formally, the optimal point estimate of the posterior pdf is

$$\begin{aligned} \hat{f}(\Theta|g_t) &\in \operatorname{Arg}\min_{\hat{f}\in\hat{f}^*} \mathcal{E}\left[\mathcal{D}\left(\hat{f}(\Theta)||f(\Theta|f_t)\right)|g_t\right] = \qquad (3.27) \\ &= \operatorname{Arg}\min_{\hat{f}\in\hat{f}^*} \left[\mathcal{D}\left(\hat{f}(\Theta)||f(\Theta)\right) - t\int_{\Theta^*}\int_{\Psi^*} \hat{f}(\Theta)\ln[M(\Theta,\Psi)]\mathcal{E}[f_t(\Psi)|g_t]\, d\Psi\, d\Theta\right]. \end{aligned}$$

The conditional expectation $\mathcal{E}[\cdot|g_t]$ is taken with respect to the uncertain empirical pdf $f_t(\Psi),\ \Psi \in \Psi^*$.

The minimizing argument can be found explicitly as

$$\hat{f}(\Theta|g_t) \propto f(\Theta)\exp\left\{t\int_{\Psi^*} \ln[M(\Theta,\Psi)]\mathcal{E}[f_t(\Psi)|g_t]\, d\Psi\right\}. \qquad (3.28)$$

3.4.3 Specification of $\mathcal{E}[f_t(\Psi)|g_t]$

The estimate (3.28) depends only on the conditional expectation

$$\hat{f}_t(\Psi) \equiv \mathcal{E}\left[f_t(\Psi)|g_t\right],\ \Psi \in \Psi^*. \qquad (3.29)$$

Its choice and the resulting estimation algorithm are described in this subsection.

First we notice that the empirical pdf $f_t(\Psi)$ obeys the identity (3.24) that is linear in it. Consequently, its conditional expectation $\hat{f}_t(\psi) = \mathcal{E}[f_t(\Psi)|g_t]$ has to fulfill it, too,

$$g_t = \int_{\Psi^*} h(\Psi)\mathcal{E}[f_t(\Psi)|g_t]\, d\Psi + g_0 \underbrace{\Leftrightarrow}_{(3.29)} g_t - g_0 = \int_{\Psi^*} h(\Psi)\hat{f}_t(\Psi)\, d\Psi. \qquad (3.30)$$

Obviously, $\hat{f}_t(\Psi)$ has to be pdf. The identity (3.30) is the only objective knowledge we have about it. Thus, it is reasonable to select it as close as possible to the pdf describing prior information about $f_t(\Psi)$ while respecting the restriction (3.30).

The prior distribution on Ψ is $f(\Psi) = \int f(\Psi|\Theta)f(\Theta)\, d\Theta = f(\psi)\int M(\Theta,\Psi)f(\Theta)\, d\Theta$. Where $f(\psi)$ describes our prior information on possible regression vectors. It can be taken as a flat pdf with its support on ψ^*. The pdf $M(\Theta,\Psi)$ is the model in question and $f(\Theta)$ is the chosen prior pdf on its parameters. As usual, we measure the distance of $\hat{f}_t(\Psi)$ to $f(\Psi)$ with the help of the KL distance. Taking into account the restriction (3.30), we minimize

$$\begin{aligned} &\min_{\hat{f}(\Psi)} \int \hat{f}(\Psi)\left[\ln\left(\frac{\hat{f}(\Psi)}{f(\Psi)}\right) - v_t'h(\Psi)\right] d\Psi = \qquad (3.31) \\ &= \min_{\hat{f}(\Psi)} \mathcal{D}\left(\hat{f}(\Psi)||\frac{f(\Psi)\exp\left(v_t'h(\Psi)\right)}{\int f(\tilde{\Psi})\exp\left(v_t'h(\tilde{\Psi})\right) d\tilde{\Psi}}\right) - \ln\left(\int f(\tilde{\Psi})\exp\left(v_t'h(\tilde{\Psi})\right) d\tilde{\Psi}\right), \quad \text{where} \end{aligned}$$

the vector v_t of Lagrangian multipliers is to be selected so that (3.30) is met. This requirement, the second form of the minimized functional and properties of the KL distance give the optimal $\hat{f}(\Psi)$

$$\hat{f}(\Psi) = \frac{f(\Psi)\exp\left(v_t' h(\Psi)\right)}{\int f(\Psi)\exp\left(v_t' h(\Psi)\right)\, d\Psi}, \quad g_t - g_0 = \int h(\Psi)\hat{f}_t(\Psi)\, d\Psi = \frac{\partial}{\partial v_t}\ln\left(\int f(\Psi)\exp\left(v_t' h(\Psi)\right)\, d\Psi\right) \tag{3.32}$$

This option completes the overall algorithm.

Algorithm 3.4.1 (Recursive estimation based on equivalence approach)
Initial (off line) mode

- *Select the parameterized model* $f(\Delta_t|a_t, d(t-1), \Theta) \equiv M(\Theta, \Psi_t)$.
- *Select the prior pdfs* $f(\Theta)$ *and* $f(\psi)$.
- *Evaluate the prior pdf* $f(\Psi) = f(\psi)\int M(\Theta,\Psi) f(\Theta)\, d\Theta$ *on* Ψ^*.
- *Select the generalized vector function* $G(\Theta)$ *such that* $\int G(\Theta)\, d\Theta = 0$.
- *Prepare the evaluation of the function*

$$h(\Psi) \equiv \int \ln(M(\Theta,\Psi)) G(\Theta)\, d\Theta. \tag{3.33}$$

- *Set* $g_0 = 0$.

Sequential (on line) mode, *running for* $t \in t^*$,

1. *Measure data vector* Ψ_t.
2. *Evaluate* $h_t \equiv h(\Psi_t) \equiv \int \ln(M(\Theta,\Psi_t)) G(\Theta)\, d\Theta$.
3. *Update the weights* $g_t = g_{t-1} + h_t$.
4. *Find the vector* v_t *solving the equation*

$$g_t = \frac{\partial}{\partial v_t}\ln\left(\int f(\Psi)\exp\left(v_t' h(\Psi)\right)\, d\Psi\right)$$

5. *Exploit the obtained approximation of the posterior pdf*

$$\hat{f}(\Theta|g_t) \propto f(\Theta)\exp\left[\frac{\int \ln(M(\Theta,\Psi)) f(\Psi)\exp\left(v_t' h(\Psi)\right)\, d\Psi}{\int f(\Psi)\exp\left(v_t' h(\Psi)\right)\, d\Psi}\right]. \tag{3.34}$$

Remark(s) 3.4.1

1. *Zero value of the initial weights* g_0 *reflects that the weights are always determined by the increment* $g_t - g_0$*, see (3.32).*
2. *The name "equivalence approach" stresses the fact that the set of posterior pdfs is reduced to equivalence classes. The pdfs with the same representation* g *cannot be distinguished.*
3. *The required* commutative updating and projecting *of the posterior pdfs is crucial. The recursion for* g*s is exact and the approximation errors caused by the use of* $\hat{f}(\Theta|g_t)$ *instead of* $f(\Theta|d(t)) \equiv f(\Theta|f_t(\cdot))$ *do not accumulate! Use of a non-commutative projection* $\mathcal{G}_t : f^*(\Theta|d(t)) \rightarrow g_t^*$ *is always endangered by a divergence as the estimation described by the Bayes rule can be viewed as a dynamic system evolving* $f(\Theta|d(t))$ *at stability boundary.*
4. *The indicated integrations represent computationally the most demanding part of the algorithm. They can be performed in off line mode if their results can be efficiently stored (the resulting functions interpolated). The solution of the non-linear equation for* v_t *is also hard but standard problem.*

5. *We would like to get the exact posterior pdf if the model belongs to the exponential family (3.5). This dictates the choice of the mapping $\mathcal{G}$ that should make $h(\Psi) = [B'(\Psi), 1]'$. For it is sufficient, to introduce the prior initial moments of the vector function $\tilde{C}(\Theta) \equiv [C'(\Theta), \ln(A(\Theta))]'$*

$$\bar{C} \equiv \int \tilde{C}(\Theta) f(\Theta)\, d\Theta,\ \mathcal{C} \equiv \mathrm{cov}\left[\tilde{C}\right] = \int \tilde{C}(\Theta)\tilde{C}'(\Theta) f(\Theta)\, d\Theta - \bar{C}\bar{C}'$$

and define the weighting function

$$G(\Theta) \equiv [\tilde{C}(\Theta) - \bar{C}]'\mathcal{C}^{-1} f(\Theta).$$

Problem 3.4.1 (How to choose the mapping $\mathcal{G}$?) *The (generalized) functions G represent the key tuning knobs of the approach. Options leading to discrete versions of the function and/or its derivatives, or $M(\Theta, \Psi_i)$ on a grid of Ψ_i have been tried with a success but a deeper insight is needed in order to arrive to a cook book.*

Possible direction can be found by approximating $M(\Psi, \Theta)$ by a member from exponential family and by using the recommendation given in the step 5 in Remarks 3.4.1.

Chapter 4

Approximate design

The complexity of the optimal design stems from the fact that the optimal "plan" of dynamic decisions is essentially searched for by comparing all possible behaviors resulting from the interaction of the system and the judged strategy. The design complexity is significantly influenced by the richness of the inspected space. Its reduction is behind the majority of available approximation schemes including those discussed here.

4.1 Adaptive systems

The ideal solution of the decision-making under uncertainty is described by the combination of Bayesian filtering, see Proposition 2.5.1, and dynamic programming, see Propositions 2.4.2, 2.4.4. The functional equations describing them are mostly computationally infeasible. Consequently, their use cannot live without approximations. Even the simpler fully probabilistic design, see Proposition 2.4.4, operates on infinite-dimensional objects.

The involved multi-variate functions should be represented in a computer, i.e. in the device that can operate on a high but finite amount of values. Thus, a sort of approximation is needed. Global approximation of functions of many variables we are dealing with is known to be computationally hard. On the other hand, the application of the strategies resulting from the design requires knowledge of the discussed solutions only for the recorded experience. Thus, it is sufficient to know them locally around the actual experience. This can often be approximately achieved. Such *local approximator* can be identified with *adaptive system* [41]. The mixture model we use as well as its quasi-Bayes estimation, see Chapter 6, follow exactly this direction. In this way, we get practically feasible learning.

This Chapter discusses general localization principles used in connection with the design made for the given model.

4.2 Suboptimal design

The design complexity is the key issue addressed repeatedly at various places of this text. At the computational stage we are now discussing, the design complexity stems from

- richness of the space of the ignorance part of the behavior that has to be inspected for the optimal selection of decisions;

- complexity of models describing relationships of the experience and optional decisions to ignorance part of the behavior.

The suboptimal design tries to reduce influence of one or both these sources of complexity. Here, we describe the common ways relevant to our basic mixture model.

4.2.1 Strategies examining reduced space

Here, we mention common techniques oriented on simplification of the space searched for.

Receding horizon

The reduction of the design horizon is the most obvious way to a simplified (suboptimal) design. The reduction obtained by planning just one-step-ahead has been popular for a long time [42]. Dynamic decision-making, however, means that consequences of a decision are encountered far behind the time moment of its application. Consequently, the decision that is optimal when judged from a short-sighted perspective might be quite a bad one from the long-term viewpoint, [13].

This observation has stimulated search for a compromise between the ideal planning over the whole horizon of interest and short-sighted, locally optimizing strategies.

When we plan looking just few steps ahead we get just an approximation of the optimal design. Thus, it is reasonable to correct the planned decisions permanently. After applying the initial decision(s) implied by the temporary design and after exploiting a new measured knowledge item about the system state, the design should be repeated. This is essence of the design technique called *receding horizon strategy*. Let us describe its algorithm in the case of additive loss function (2.22) and for a pre-specified value T of the *receding horizon* $T < \mathring{t}$.

Algorithm 4.2.1 (Receding horizon strategy) *Repeat for* $t \in t^* \equiv \{1, \ldots, \mathring{t}\}$,

1. *Find the strategy* $\mathcal{R}(t\_t+T) = (\mathcal{R}_t, \ldots, \mathcal{R}_{t+T})$ *approximately minimizing*

$$\mathcal{E}\left[\sum_{\tau=t}^{t+T} z(\Delta(\tau), a(\tau)) | \mathcal{P}_{a_t}\right]$$

   using the available experience and outer model of the system.

2. *Apply* $a_{t|t-1} \equiv \mathcal{R}_t(\mathcal{P}_{a_t})$ *for the available* $\mathcal{P}_{a_t}$.

3. *Extend the experience by the new data* Δ_{a_t}, $a_{t|t-1}$ *and use it in learning – perform filtration or estimation – so that an improved outer model of the system is obtained.*

Iterations spread in time

It is intuitively clear and practically verified that the good receding horizon T can be too large if the decision horizon $\mathring{t}$ is large. In the case of additive loss, the asymptotic analysis of the design, [15] shows that for $\mathring{t} \to \infty$ the optimal decision is the minimizer in

$$^{\lfloor\infty}\mathcal{V}(x_{t-1}) + C = \min_{a_t \in a_t^*} \mathcal{E}\left[z(x_t, a_t) + {}^{\lfloor\infty}\mathcal{V}(x_t) | a_t, x_{t-1}\right], \text{ where}$$

x_t denotes observed information state of the system. This stationary form can be interpreted as one-step-ahead design for the partial loss $z(x_t, a_t)$ increased by the stationary Bellman function $^{\lfloor\infty}\mathcal{V}(x_t)$. Knowing this fact, we could use receding horizon $T = 1$. At the same time, dynamic programming has been interpreted as an iterative search for the stationary value $^{\lfloor\infty}\mathcal{V}(x_t)$. These observations led to the following approximate strategy.

Algorithm 4.2.2 (Strategy with iterations spread in time) *Select an initial guess of stationary Bellman function* $\mathcal{V}_0(x)$ *and repeat for* $t \in t^* \equiv \{1, \ldots, \mathring{t}\}$,

1. *Find the strategy* $\mathcal{R}(t\_t+T) = (\mathcal{R}_t, \ldots, \mathcal{R}_{t+T})$ *approximately minimizing*

$$\mathcal{E}\left[\sum_{\tau=t}^{t+T} z(x_\tau, a_\tau) + \mathcal{V}_{t-1}(x_{t+T-1}) | x_{t-1}\right]$$

   using the available experience and outer model of the system.

   Evaluate it for all possible states x^* *in the role of* x_{t-1}.

2. *Take the final Bellman function $\mathcal{V}_t(x)$ resulting from this design as an updated approximation of the stationary Bellman function.*

3. *Apply $a_{t|t-1} \equiv \mathcal{R}_t(\mathcal{P}_{a_t})$ for $\mathcal{P}_{a_t} \equiv$ the measured x_{t-1}.*

4. *Extend the experience by the new data Δ_{a_t}, $a_{t|t-1}$ and use it in learning (perform filtration or estimation) so that an improved outer model of the system is obtained.*

Predictive strategies

There is a whole set of strategies that drastically reduce the space of inspected behavior by working only with point predictions of uncertain quantities to be influenced. Sometimes, they deal with a sort of set predictions by considering credibility prediction sets of a very simplified form. Such strategies are usually labelled as *predictive strategies* [43, 44].

Let us consider an additive loss function with receding horizon T. It serves us an example demonstrating essence of these strategies. For minimization purposes, the predictive strategies use the approximation

$$\mathcal{E}\left[\sum_{\tau=t}^{t+T} z\left(\Delta(\tau), a(\tau)\right)|\mathcal{P}_{a_t}\right] \approx \sum_{\tau=t}^{t+T} z\left(\mathcal{E}\left[\Delta(\tau)|\mathcal{P}_{a_t}\right], a(\tau)\right).$$

Obviously, such an approximation may be quite rough when there is a non-negligible uncertainty, i.e. lack of knowledge or noise influence. On the other hand, the gained simplification allows the user to concentrate on respecting hard bounds on actions or other quantities. The practical significance of technological restrictions justifies this simplification and explains the popularity of such a treatment.

4.2.2 Strategies simplifying models

Let us consider the receding horizon strategy applied at time t. Then, a substantial degree of the design complexity is caused by the use of predictive pdfs $\{f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau})\}_{\tau=t}^{t+T}$ obtained through the Bayesian estimation. They have the form, see Proposition 2.5.2,

$$f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau}) = \int_{\Theta^*} f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau}, \Theta) f(\Theta|\mathcal{P}_{a_\tau})\, d\Theta. \tag{4.1}$$

For a relatively short planning horizon T, the predictors (4.1) can be approximated by

$$f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau}) \approx \int_{\Theta^*} f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau}, \Theta) \hat{f}_\tau(\Theta)\, d\Theta,\ \tau = t, \ldots, t+T, \tag{4.2}$$

with a simpler pdf $\hat{f}_\tau(\Theta) \approx f(\Theta|\mathcal{P}_{a_\tau})$. Bellow we outline wide-spread choices of $\hat{f}_\tau(\Theta)$ used.

Super-cautious strategy

If the posterior pdf $f(\Theta|\mathcal{P}_{a_t})$ is reasonably concentrated then it makes sense to exploit the approximate equality

$$f(\Theta|\mathcal{P}_{a_\tau}) \approx f(\Theta|\mathcal{P}_{a_t}) \equiv \hat{f}_t(\Theta),\ \tau = t, \ldots, t+T. \tag{4.3}$$

The receding horizon strategy combined with the approximation (4.2), (4.3) is called *super-cautious* as it is based on assumption that we learn nothing about unknown parameters till the receding horizon. The uncertainty of the parameter estimates (4.3) projected into the predictors through the formula (4.2) inhibits excessive decision values. It makes the strategy super-cautious.

Cautious strategy

The assumption (4.3) is often too strong. Then, it is reasonable to weaken it to

$$f(\Theta|\mathcal{P}_{a_\tau}) \approx F(\tau, f(\Theta|\mathcal{P}_{a_t})) \equiv \hat{f}_\tau(\Theta),\ \tau = t, \ldots, t+T,\ \Theta \in \Theta^*. \tag{4.4}$$

The mapping $F(\tau, \cdot)$ modifies the posterior pdf $f(\Theta|\mathcal{P}_{a_t})$ to other, more concentrated, pdfs. The mapping F is chosen beforehand and does not use data belonging to the ignorance $\mathcal{F}_{a_t}$. Typically, the generated pdf $\hat{f}_\tau(\Theta)$ has the same 1st moment (expectation) as $f(\Theta|\mathcal{P}_{a_t})$ but its covariance $C_{\tau|t}$ decreases with increasing distance $\tau - t$ in a way specified beforehand. Often,

$$C_{\tau|t} = b(\tau - t)C_t \equiv b(\tau - t) \times \text{ covariance computed for } f(\Theta|\mathcal{P}_{a_t}).$$

The following non-negative scalar functions $b(\kappa)$ serve as representative examples.

$$b(\kappa) = 1/\kappa \text{ or } b(\kappa) = \begin{cases} 1 & \text{if } \kappa = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{4.5}$$

Such a choice can be seen as *sharpening* of the pdf $f(\Theta|\mathcal{P}_{a_t})$ by taking its power $f^{\lambda_\tau^{-1}}(\Theta|\mathcal{P}_{a_t})$ using a pre-specified sequence $\{\lambda_\tau \in (0,1)\}_{\tau=t}^{t+T}$.

Certainty-equivalence strategy

This strategy is the most wide spread one. It replaces unknown parameter in the parameterized model by a current point estimate $\hat{\Theta}_t$ of unknown parameters

$$f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau}) \approx f(\Delta_{a_\tau}|a_\tau, \mathcal{P}_{a_\tau}, \hat{\Theta}_t), \;\; \tau = t, \ldots, t+T. \tag{4.6}$$

It corresponds to the approximate parameter estimate

$$f(\Theta|\mathcal{P}_{a_\tau}) \approx \hat{f}_t(\Theta) \equiv \delta(\Theta - \hat{\Theta}_t), \;\; \tau = t, \ldots, t+T, \text{ where} \tag{4.7}$$

$\delta(\cdot)$ is Dirac delta function, a formal pdf of the measure concentrated on zero value.

Active strategies

All outlined strategies are *passive strategies*. The planned decisions do not care about learning as the used pdf $\hat{f}_\tau(\Theta)$ depend on the experience $\mathcal{P}_{a_t}$ only for all $\tau \geq t$. At the same time, it is known, see [20, 21] that the optimal strategy is *dual strategy*. It cares both about immediate decisions and learning process in a balanced way. This motivates a use of measures that support learning. The resulting strategies are known as *active strategies*. Two basic suboptimal ways are used.

- An external stimulating signal is feeded into the closed loop. It is added to optional quantities like inputs or set points.

- A term is added to the loss original function that reflects quality of learning even when a passive-type design is applied [45].

4.3 Decomposition of decision-making

Repeatedly, splitting of the decision-making task in a chain of sub/tasks is declared as the way of converting optimal design to an approximation of the practical optimal design. At the same time, it is known that any violation of the *golden decision-making rule* – try to solve the problem at hand in its entirety – makes the solution to depart from its optimum. Lack of the formal tools for the decomposition leave us with empirical rules in this area.

Here, we give, as an example, the list describing such decomposition in the case of adaptive control with learning part based on parameter estimation. Each item in the list has been found as a relatively self-containing decision sub-problem.

The design, as any human activity, is iterative. Naturally, the majority of iterations should be concentrated to the off line phase in order to minimize expenses related to the commission of the controller.

4.3.1 Off line phase

The following indicative list of sub-tasks is solved, often with internal iterations, until the potential user is satisfied.

- Formulate the addressed problem, i.e. get the specification of
  - the technical control aims,
  - the system,
  - the available data, system inputs and outputs,
  - the technological and complexity restrictions,
  - the knowledge available.
- Perform experimental design and collect data for off line analysis.
- Make data preprocessing.
- Select class of parameterized models.
- Quantify prior knowledge.
- Estimate model structure and control period.
- Estimate forgetting factor.
- Perform generalized Bayesian estimation based on prior knowledge and available data.

  These results are used as the prior and/or alternative pdf in on line phase.
- Select the loss function.

  Typically, weights of quadratic loss or the ideal in fully probabilistic design have to be chosen. The iterative choice runs as follows:

  Do until the results cannot be improved
  - Select suboptimal control design and its parameters.
  - Perform prior design of controller and prediction of closed loop (coupling system $\leftrightarrow$ controller) behavior by transforming all uncertain quantities into the user-defined quality indicators.

    The formal solution provides Proposition 2.3.5 with a Monte Carlo evaluation employed. It becomes feasible using sequential stopping rules [46, 47].
  - Compare results with user's specification. Set estimation is useful to it.
- Validate the results.

  Again Monte Carlo techniques are quite useful.

4.3.2 On line phase

These decision-making sub-tasks are solved in real time for $t \in t^*$. Here, there is almost no freedom for iterative trial-and-error solutions. Of course, it is wise to store the data collected during the on line phase and use them for an improved off line design.

- Collect and preprocess data.
- Generate reference signals.
- Estimate with stabilized forgetting.
- Use receding horizon or iterations-spread-in time design, in an appropriate version.
- Generate input using the designed strategy and measured data.

- Check possible discrepancies and make a finer tuning of optional parameters of the design.

  This supervision can be based on theory of operator control [48], that exploits mixture estimation, and that is discussed in this work.

Problem 4.3.1 (Completion of the design support) *The presented outline shows the complexity of the overall design chain. Some steps are weakly supported by theory and algorithms. Some of them are not supported at all. In spite of this, our experience indicates that the presented theory provides the most complete unified support available. Still a development of this support represents a significant research challenge.*

Chapter 5

Problem formulation

This chapter states the problem of the advisory system design and outlines its conceptual solution.

Relationships of the encountered systems – the managed one, its connection with operator and advisory system – are clarified in Section 5.1. A special attention is paid to relationships of data spaces accessible to the operator and advisory system. The considered advisory scenario and the adopted design principle are discussed in Subsection 5.1.1.

Necessary reductions and extensions of involved data spaces are treated in Subsections 5.1.2 and 5.1.3. This prepares a formal discussion of advisory systems contained in Section 5.4.1. Learning conditions of the advisory system are specified in Section 5.2. Dynamic predictions, forming bridge between learning and advising parts, are discussed in Section 5.3. The design of advices offered to the operator is described in Section 5.4. The design of the presentation part of the advisory system is in Subsections 5.5.1, 5.5.2. Design steps making backbone of detailed solutions are listed in Section 5.6.2.

5.1 Systems and data spaces

Here, relationships of the managed system, its operator and his advisory system are inspected. First, we introduce relevant notations.

Agreement 5.1.1 (Nomenclature of systems) *The system managed by the operator is called* m-system. *The closed loop formed by the m-system and its operator (supervisor) is called* o-system. *The* advisory system*, a computer program feeded by information measured on the o-system and giving advices to the operator, is called* p-system *(*p stands for ProDaCTool*).*

If need be, the quantities related to m-, o- and p-systems are distinguished by prefixes m-, o- and p- or by subscripts m, o, p.

Out of this chapter, we deal mostly with data handled by the p-system. Then, their subscript p is dropped whenever there is no danger of misunderstanding.

In order to formulate and solve the overall design properly, we have to clarify relationships of quantities dealt with by o- and p-systems.

Agreement 5.1.2 (Data, observation and action spaces) *Values of the quantities available to the operator form the* data space of the operator $d_o^*(\mathring{t})$*. This space is the Cartesian product of the* observation space of the operator $\Delta_o^*(\mathring{t})$ *and of the* action space of the operator $a_o^*(\mathring{t})$.

Values of the quantities available to the advisory system form the data space of the advisory system $d_p^*(\mathring{t})$*. This space is the Cartesian product of the* observation space of the advisory system $\Delta_p^*(\mathring{t})$ *and of the* action space of the advisory system $a_p^*(\mathring{t})$.

Values of the quantities available to the operator but unavailable to the advisory system form the surplus data space of the operator $d_{o+}^*(\mathring{t})$.

Values of the quantities available to the advisory system but unavailable to the operator form the surplus data space of the advisory system $d_{p+}^*(\mathring{t})$.

Possible differences of the data spaces, i.e. non-emptiness of the surplus data spaces, single out the addressed decision-making problem from the standard formulation of decision making.

Obviously, the design of the advisory system is meaningful only when the data spaces overlap.

Requirement 5.1.1 (Overlap of data spaces) *The data spaces available to the operator $d_o^*(\mathring{t})$ and to the advisory system $d_p^*(\mathring{t})$ have a non-empty intersection $d_{op}^*(\mathring{t}) \neq \emptyset$. Thus,*

$$d_o^*(\mathring{t}) = d_{op}^*(\mathring{t}) \cup d_{o+}^*(\mathring{t}),\ d_{op}^*(\mathring{t}) \cap d_{o+}^*(\mathring{t}) = \emptyset,\quad d_p^*(\mathring{t}) = d_{op}^*(\mathring{t}) \cup d_{p+}^*(\mathring{t}),\ d_{op}^*(\mathring{t}) \cap d_{p+}^*(\mathring{t}) = \emptyset. \tag{5.1}$$

$d_{o+}^*(\mathring{t})$ is the surplus data space of the operator and $d_{p+}^*(\mathring{t})$ is the surplus data space of the advisory system.

5.1.1 Basic scenario and design principle

The following basic scenario is considered within the time span determined by a *horizon* $\mathring{t} \leq \infty$.

The operator handling the m-system deals with a sequence $d_o(\mathring{t})$ of the o-data. A non-empty part of the data record $d_{o;t}$ is formed by his actions $a_{o;t} \in a_o^*$. The rest is formed by innovations $\Delta_{o;t}$, i.e. $d_{o;t} = (\Delta_{o;t}, a_{o;t}) \equiv$(o-innovations, o-actions). Formally, he implements his causal strategy

$$\left\{d_o^*(t-1) \to a_{o;t}^*\right\}_{t\in t^*}.$$

The strategy of an experienced operator should minimize the expected value $\mathcal{E}$ of a loss function

$$\mathcal{Z} : d_o^*(\mathring{t}) \to [0,\infty] \tag{5.2}$$

that reflects the management aims. The joint pdf $f(d_o(\mathring{t}))$ would be the adequate description of the o-system needed for a formal optimization.

The p-system works on a sequence of the p-data $d_p(\mathring{t})$. Each record $d_{p;t}$ contains data items $d_{pi;t},\ i \in i^* \equiv \{1,\ldots,\mathring{d}_p\}$ with either real or discrete values. The record $d_{p;t} \equiv (\Delta_{p;t}, a_{p;t})$ includes p-innovations $\Delta_{p;t}$ observed by the p-system on the o-system and actions of the p-system $a_{p;t}$, called *advices*.

The p-system implements its causal strategy $\left\{d_p^*(0), d_p^*(t-1) \to a_{p;t}^*\right\}_{t\in t^*}$, where $d_p(0)$ denotes experience collected before using the p-system.

The p-system strategy is designed so that the o-system accepting advices minimizes the expected loss (5.2)

Remark(s) 5.1.1

We care about an exchange of data between considered systems (parts of the world of interest) and about the influence of actions considered. The m-system communicates with the operator and with the advisory system in different ways:

- *it provides the o-innovations to the operator as responses to his o-actions,*
- *it provides some data to the p-system, generally different from $d_{o;t}$,*
- *it is influenced by the p-system indirectly through the actions of the operator who is stimulated by advices, by the actions of the advisory system.*

Problem 5.1.1 (Advising on inner quantities)

Generally, the operator cares about inner, directly unobservable states, too. He has to guess them using observed data and to optimize also behavior of these guesses. At the end, he thus optimizes the observed-data dependent loss function. For simplicity, the advising is formulated directly as the data-driven design, cf. Agreement 2.4.2. The relevant theory should be, however, extended explicitly to the case when the operator deals with inner, directly unobservable states, too.

We want to stimulate the operator to behave as an experienced one. The advisory system cannot force the operator to follow its advices. The p-system can only present a "target" to be reached. The *ideal pdf* in the sense of the fully probabilistic design, see Agreement 2.4.1, can and will provide such a target representation.

The complete outer description of an inter-connection of o- and p-systems is given by the joint pdf $f(d_p(\mathring{t}), d_{o+}(\mathring{t}))$ of all involved data $(d_p(\mathring{t}), d_{o+}(\mathring{t}))$, cf. (5.1). The adopted *design principle* is expressed in terms of these data. An assumption, formulated in Section 5.1.2, will allow us to drop the surplus data of the o-system $d_{o+}(\mathring{t})$.

Algorithm 5.1.1 (Design principle of the advisory system)

1. *Express managing aims as* user ideal *pdf* ${}^{\lfloor U}f\left(d_p(\mathring{t}), d_{o+}(\mathring{t})\right)$.

2. *Estimate an outer multi-modal model* $f(d_p(\mathring{t}), d_{o+}(\mathring{t}))$ *describing relationships among the considered data* $(d_p(\mathring{t}), d_{o+}(\mathring{t}))$.

3. *Create the ideal pdf* ${}^{\lfloor I}f\left(d_p(\mathring{t}), d_{o+}(\mathring{t})\right)$ *that*

   - *is as close as possible to the user ideal pdf* ${}^{\lfloor U}f\left(d_p(\mathring{t}), d_{o+}(\mathring{t})\right)$,
   - *inherits those constituents of* $f\left(d_p(\mathring{t}), d_{o+}(\mathring{t})\right)$ *that cannot be changed even by the fully co-operating operator.*

4. *Present low dimensional projections of the ideal pdf* ${}^{\lfloor I}f\left(d_p(\mathring{t}), d_{o+}(\mathring{t})\right)$ *as the target to be followed by the operator.*

The optimization with properly preserved elements of the m-system guarantees that a good past practice is followed. The ideal pdf constructed in the outlined way

- is reachable,
- relates all measured consequences to their measured causes: it improves guiding comparing to that exploiting on a few observed indicators of a good behavior.

The elements that are optimized as well as the ideal pdf resulting from optimization are distinguished by the left *superscript* ${}^{\lfloor I}$.

5.1.2 Reduction of surplus data of the operator

The p-system has no information on $d^*_{o+}(\mathring{t})$. Thus, it has to leave this part of the o-behavior to its fate. Formally, we have to restrict the constructed ideal ${}^{\lfloor I}f(\cdot)$ by the requirement

$$ {}^{\lfloor I}f\left(d_{o+}(\mathring{t})|d_p(\mathring{t})\right) \equiv f\left(d_{o+}(\mathring{t})|d_p(\mathring{t})\right), \text{ where} \tag{5.3}$$

$f(d_{o+}(\mathring{t})|d_p(\mathring{t}))$ is the "objective" pdf – in the sense discussed in Chapter 2 – describing the data $d_{o+}(\mathring{t})$ unavailable to the p-system.

According to the concept of the fully probabilistic design, we measure the distance of the inspected behavior to its ideal through the KL distance (2.26). For that, the requirement (5.3) implies a simple but important consequence.

Proposition 5.1.1 (Ideal pdf offered by the advisory system) *Let the ideal pdf* ${}^{\lfloor I}f(d_p(\mathring{t}), d_{o+}(\mathring{t}))$ *offered by the advisory system meet the requirement (5.3). Then,*

$$\begin{aligned}
\mathcal{D}\left(f||{}^{\lfloor I}f\right) &\equiv \int f(d_p(\mathring{t}), d_{o+}(\mathring{t}))\ln\left(\frac{f(d_p(\mathring{t}), d_{o+}(\mathring{t}))}{{}^{\lfloor I}f(d_p(\mathring{t}), d_{o+}(\mathring{t}))}\right) d(d_p(\mathring{t}), d_{o+}(\mathring{t})) = \qquad (5.4)\\
&= \int f(d_p(\mathring{t}))\ln\left(\frac{f(d_p(\mathring{t}))}{{}^{\lfloor I}f(d_p(\mathring{t}))}\right) dd_p(\mathring{t}) \quad \textit{and}\\
\mathcal{D}\left({}^{\lfloor I}f||f\right) &\equiv \int {}^{\lfloor I}f(d_p(\mathring{t}))\ln\left(\frac{{}^{\lfloor I}f(d_p(\mathring{t}))}{f(d_p(\mathring{t}))}\right) dd_p(\mathring{t}).
\end{aligned}$$

Thus, for our design purposes, the outer description of the o-system $f(\cdot)$ *and the ideal pdf* ${}^{\lfloor I}f(\cdot)$ *have to be specified on* $d^*_p(\mathring{t})$ *only.*

Proof: Using the basic rules for pdfs – chain rule and normalization, see Proposition 2.3.4 – we can directly verify the following identities

$$\mathcal{D}\left(f||{}^{\lfloor I}f\right) \equiv \int f(d_p(\mathring{t}), d_{o+}(\mathring{t})) \ln\left(\frac{f(d_p(\mathring{t}), d_{o+}(\mathring{t}))}{{}^{\lfloor I}f(d_p(\mathring{t}), d_{o+}(\mathring{t}))}\right) d(d_p(\mathring{t}), d_{o+}(\mathring{t})) =$$

$$\underbrace{=}_{chain\ rule} \int f(d_{o+}(\mathring{t})|d_p(\mathring{t}))f(d_p(\mathring{t})) \ln\left(\frac{f(d_{o+}(\mathring{t})|d_p(\mathring{t}))f(d_p(\mathring{t}))}{{}^{\lfloor I}f(d_{o+}(\mathring{t})|d_p(\mathring{t}))\ {}^{\lfloor I}f(d_p(\mathring{t}))}\right) d(d_p(\mathring{t}), d_{o+}(\mathring{t})) =$$

$$\underbrace{=}_{(5.3)} \int f(d_{o+}(\mathring{t})|d_p(\mathring{t}))f(d_p(\mathring{t})) \ln\left(\frac{f(d_p(\mathring{t}))}{{}^{\lfloor I}f(d_p(\mathring{t}))}\right) d(d_p(\mathring{t}), d_{o+}(\mathring{t})) =$$

$$\underbrace{=}_{normalization} \int f(d_p(\mathring{t})) \ln\left(\frac{f(d_p(\mathring{t}))}{{}^{\lfloor I}f(d_p(\mathring{t}))}\right) dd_p(\mathring{t}).$$

The equality for $\mathcal{D}\left({}^{\lfloor I}f||f\right)$ can be proved in the same way. □

Consequently, the accepted condition (5.3) allows us to leave the surplus data of the operator $d^*_{o+}(\mathring{t})$ completely out of our considerations. Thus, under (5.3), our design results obtained for empty and non-empty $d^*_{o+}(\mathring{t})$ are the same. Notation is, however, simpler if $d^*_{o+}(\mathring{t}) = \emptyset$. In this case, the definition (5.1) implies

$$d^*_o(\mathring{t}) = d^*_{op}(\mathring{t}) \ \Rightarrow d^*_o(\mathring{t}) \subset d^*_p(\mathring{t}). \tag{5.5}$$

From now on, we adopt this formal simplification that the surplus data space of the operator $d^*_{o+}(\mathring{t})$ is empty and thus the data space of the operator $d^*_o(\mathring{t})$ is a subset of the data space of the advisory system $d^*_p(\mathring{t})$.

5.1.3 Extension of the user ideal to the surplus data of the p-system

The adopted condition (5.5) implies that an implementation of conceptual Algorithm 5.1.1 requires specification of a user ideal pdf ${}^{\lfloor U}f(\cdot)$ on the data space of the advisory system $d^*_p(\mathring{t})$. This ideal is, however, defined "naturally" on the data space of the operator $d^*_o(\mathring{t}) \underbrace{\equiv}_{(5.5)} d^*_{op}(\mathring{t})$. At the same time, the fact that the advisory system deals with a wider data set than the operator should be exploited for reaching a better quality of advices. Thus, it is reasonable to extend the user ideal from $d^*_o(\mathring{t})$ to $d^*_p(\mathring{t})$.

By definition, the operator is not aware and consequently interested in the surplus data of the advisory system $d_{p+}(\mathring{t})$. Thus, he has to leave this surplus data to their fate determined by the managed system and by the influence of advisory system. It motivates the choice

$$\begin{aligned} {}^{\lfloor U}f(d_p(\mathring{t})) &\underbrace{=}_{(5.1),(5.5)} {}^{\lfloor U}f(d_o(\mathring{t})|d_{p+}(\mathring{t}))\ {}^{\lfloor U}f(d_{p+}(\mathring{t})) \equiv \\ &\equiv {}^{\lfloor U}f(d_o(\mathring{t})|d_{p+}(\mathring{t}))\ {}^{\lfloor I}f(d_{p+}(\mathring{t})) \equiv {}^{\lfloor U}f(d_o(\mathring{t}))\ {}^{\lfloor I}f(d_{p+}(\mathring{t})). \end{aligned} \tag{5.6}$$

The first equivalence in the second row means that $d^*_{p+}(\mathring{t})$ is left to its fate: the user leaves the p-system to identify this part of the user target with the option made by the p-system. The second one says that the user ideal on $d^*_o(\mathring{t})$ is constructed irrespective of the inaccessible values in $d^*_{p+}(\mathring{t})$. The identity (5.6) is adopted as a basic assumption further on. If need be the pdf ${}^{\lfloor U}f(d_o(\mathring{t}))$ is called *true user ideal pdf.*

Remark(s) 5.1.2

1. *Operator may be aware but uninterested in some entries of d_o. Then, they can be formally thought as entries in the surplus data space of operator.*

2. *The designer may alternatively use the entries out of direct user's interests as tuning knobs in design. This option is indeed exploited when we try to create user-friendly advisory system that control the burden on operator. For instance, frequency of changes in advising is often damped using them.*

5.2 Learning conditions

The design of the advisory system can be meaningful under specific conditions discussed here. Before progressing in a formal way, it is worth to recognize the following facts.

Need for any advisory system arises when there are good and bad modes in managing of the considered m-system.

Chance for designing of a good advisory system exists if both good and bad modes are reflected in its data space d_p^*.

It implies that the pdf $f(d_p(\mathring{t}))$, describing completely $d_p^*(\mathring{t})$, is expected to have *multiple modes.*

Possibility of designing of an efficient advisory system arises if the available design experience contains well pronounced information about all significant operation modes that may occur while the operator handles the managed system. The data providing this experience have to be rich and informative.

Top position of the operator in the control hierarchy implies, see Chapter 1, that the experience of the advisory system is to be predominantly data-based.

Rate of operator's actions is often (much) slower than the sampling rate of the sensors and controllers within the m-system. Consequently, all considered data can be and should be grouped or reduced to a sequence generated with the rate of operator's actions. We suppose it is done.

Experience of the p-system includes also surplus data of the advisory system $d_{p+}^*(\mathring{t}) \equiv d_p^*(\mathring{t}) \setminus d_o^*(\mathring{t})$. They are directly at disposal to the advisory system but not to the operator. These data may be invaluable in discovering of various operating modes. They have to be considered in the design of any efficient p-system. Consequently, we have to deal with all data $d_p(\mathring{t})$ and thus we can mostly drop further on the subscript p and adopt the identity

$$d(\mathring{t}) \equiv d_p(\mathring{t}). \tag{5.7}$$

The general theory, see Chapter 2, and the conditions adopted above imply that a *multiple-mode pdf* $f(d(\mathring{t})|d(0))$ is the description required for a design of the considered advisory system. The condition $d(0) \equiv d_p(0)$ stands for the experience available to the p-system before its use. Further on, $d(0)$ is fixed and formally included into $d(\mathring{t})$, Then, the notation can be simplified by "hiding" $d(0)$.

The pdf $f(d(\mathring{t}))$ can be factorized according to the chain rule $f(d(\mathring{t})) = \prod_{t\in t^*} f(d_t|d(t-1))$. The assumed low action rate of the operator implies that the real dynamics of the system is diminished in the grouped or reduced data. Thus, slow transitions among quasi-steady-state working points are modelled. This can be well described by a *Markov model of a finite order.* Thus, we can assume that the non-guided o-system is described by the joint pdf

$$f(d(\mathring{t})) = \prod_{t\in t^*} f(d_t|\phi_{t-1}), \text{where}$$

$\phi_t \in \phi^*,\ t \in t^*$, are known finite-dimensional vectors called *observable states.*

The intended *on line use* of the advisory system implies that we have to deal with models containing ϕ_t, that can be evaluated in a recursive manner, i.e. $\phi_t = \Phi(\phi_{t-1}, d_t)$ with a known function Φ.

The advisory system can only be successful if the relationships learnt from experience are valid almost permanently. Thus, we have to assume that the outer model $\{f(d_t|\phi_{t-1})\}_{t\in t^*}$ of the o-system complemented by the p-system is time invariant, cf. Agreement 2.4.1. Hypothesis on existence of several operation modes implies that we have to search for multiple-mode models. It is known that multiple mode probabilistic models can (almost) always be approximated by a *finite mixture* of uni-modal models [49], called *components.*

We estimate the appropriate model using – at least approximately – the standard Bayesian methodology. The fixed advisory system learns from historical data only. The adaptive advisory system updates the model even during its use.

For reference purposes, we summarize learning conditions adopted in the construction of the advisory system.

Requirement 5.2.1 (Learning conditions for a design of the advisory system)

1. *The data space* $d^*(\mathring{t})$ *of the advisory system (the p-system) has a non-empty intersection with the data space* $d_o^*(\mathring{t})$ *of the o-system (coupling of the managed system with operators).*

2. *The sampling rate is harmonized with the operating rate. The number of data items* $\mathring{d}$ *sent to the p-system at each time moment is fixed.*

3. *No attempt is made to influence quantities lying in the data space of the o-system and being out of the data space of the p-system.*

4. *The pdf on records* $d(\mathring{t})$ *available to the p-system is modelled by the pdf*

$$f(d(\mathring{t})|\Theta) \equiv \prod_{t \in t^*} f(d_t|\phi_{t-1}, \Theta), \ \textit{where}$$

   the pdf $f(d_t|\phi_{t-1}, \Theta)$, *determined by the measurable state* ϕ_{t-1} *and by the parameter* Θ, *is a time invariant finite mixture, see (5.8) below.*

5. *An approximate Bayesian learning, see Section 6.5, provides the pdf* $f(d(\mathring{t})) \equiv f(d(\mathring{t})|d(0)) = \prod_{t \in t^*} f(d_t|d(t-1))$ *needed for the design of the p-system.*

The finite mixture approximating distribution of data $d(\mathring{t})$ with multiple modes is assumed to have the form

$$\begin{aligned}
f(d(\mathring{t})|\Theta) &\equiv \prod_{t \in t^*} f(d_t|\phi_{t-1}, \Theta) \text{ with } \textit{finite mixtures} \text{ forming individual parameterized model} \\
f(d_t|\phi_{t-1}, \Theta) &\equiv \sum_{c \in c^*} \alpha_c f(d_t|\phi_{c;t-1}, \Theta_c, c), \ c^* = \{1, \ldots, \mathring{c}\}, \ \mathring{c} < \infty, \quad \text{where} \qquad (5.8) \\
f(d_t|\phi_{c;t-1}, \Theta_c, c) &\equiv \textit{component} \text{ given by component parameters } \Theta_c \text{ and the state} \\
\phi_{c;t} &= \Phi_c(\phi_{c;t-1}, d_t), \text{ i.e. the state can be recursively updated using new data } d_t, \\
\alpha_c &\equiv \text{the probabilistic } \textit{component weight} \\
\Theta &\equiv \text{mixture parameter formed by the component weights and parameters} \\
\Theta \in \Theta^* &\equiv \left\{ \{\Theta_c \in \Theta_c^*\}_{c \in c^*}, \ \alpha \equiv [\alpha_1, \ldots, \alpha_{\mathring{c}}] \in \alpha^* \equiv \left\{ \alpha_c \geq 0, \ \sum_{c \in c^*} \alpha_c = 1 \right\} \right\}.
\end{aligned}$$

The mixture parameter $\Theta = [\alpha_c, \Theta_c]_{c \in c^*}$ may include the number of components $\mathring{c}$.

Before fixing and refining nomenclature related to the mixture, we split the individual components in so called factors that

- provide flexibility of the parametric description,
- allow us to describe jointly continuous and discrete valued variables,
- permit us to respect common dependencies reflected in several components,
- open a way for use of different models for different entries of d_t.

Using the chain rule, Proposition 2.3.4, the pdfs $f(d_t|\phi_{c;t-1}, \Theta_c, c)$ can be written as a product of pdfs predicting individual entries of d_t. Before applying the chain rule, entries of d_t can be permuted and some permutations may lead to parameterizations with less parameters. It makes us to include into the model description the permutations

$$d \to d_c \quad \text{with} \quad d_{ic} = d_{j_{ic}}, \ \text{where} \tag{5.9}$$

j_{ic} is i-th entry of the permuted indices $[1, \ldots, \mathring{d}]$. The assignment (5.9) is applied component-wise and together with the chain rule, Proposition 2.3.4, give

$$f(d_t|\phi_{c;t-1}, \Theta_c, c) = \prod_{i \in i^*} f(d_{ic;t}|d_{(i+1)\_\mathring{d}c;t}, \phi_{c;t-1}, \Theta_{ic}, c) \equiv \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c). \tag{5.10}$$

The additional subscript i of the parameter Θ_{ic} indicates that only some entries of Θ_c may occur in i-th pdf (factor) in (5.10). Similarly, the vector $\psi_{ic;t}$ is generally a sub-vector of the vector

$$[d_{(i+1)\_\mathring{d}c;t}, \phi'_{c;t-1}]'. \tag{5.11}$$

Agreement 5.2.1 (Nomenclature related to mixtures)

Pdfs: *The pdf* $f(d_t|\phi_{c;t-1}, \Theta_c, c)$ *in (5.8) is called* parameterized component *of a mixture and* α_c *is the* weight of the c-th parameterized component.

The pdf $f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)$ *in (5.10) is called* parameterized factor.

A parameterized factor occurring in several components is common parameterized factor.

Factors are called adjacent factors *if* $d_{(i+1)c;t}$ *is in the regression vector of the factor predicting* $d_{ic;t}$ *or vice versa.*

The predictive pdf $f(d_t|d(t-1), c)$ *is called* component.

The predictive pdf $f(d_{ic;t}|d_{(i+1)\_\mathring{d}c;t}, d(t-1), c)$ *is called* factor.

The pdf $f(\Theta_{ic}|d(t))$ *is called* factor estimate.

The pdf $f(\Theta_c|d(t))$ *is called* component estimate.

The pdf $f(\alpha|d(t))$ *is called* component-weight estimate.

The pdf $f(\Theta|d(t))$ *is called* mixture estimate.

The estimate is called prior estimate *if* $t = 0$.

The estimates are called posterior estimates *if* $t \in t^*$. *Often, the posterior estimate is called estimate only.*

Data: *The vector* d_t *containing data measured at time* t *is called* data record.

The predicted scalar variable $d_{ic;t}$ *is called* output of the factor.

Data entries never playing role of the output of a factor are called non-modelled data.

The vector $\phi_{c;t-1}$*, that can be updated recursively using newest data* d_t*, is the observable* state of the parameterized component.

The parameterized factor is determined by regression vector $\psi_{ic;t}$ *defined as a sub-selection of the vector* $[d_{i+1\_\mathring{d}c;t}, \phi'_{c;t-1}]'$ *(5.11).*

The coupling $\Psi_{ic;t} \equiv [d_{ic;t}, \psi'_{ic;t}]'$ *is called* data vector *of the factor.*

The data vector Ψ *is in the* phase form *if it consists of a selection of entries from the data record* d_t *and its several delayed values* $d_t, \ldots, d_{t-\partial}$, $\partial \in \partial^* \equiv \{0, 1, \ldots, \mathring{\partial}\}$, $\mathring{\partial} < \infty$.

Structures: *Model structure is defined in a hierarchical way starting from the simplest elements, i.e. factors.*

List of regressors*, it means entries of the regression vector of a parameterized factor, is called* structure of the parameterized factor.

Structure of a parameterized factor in the phase form *is the list of pairs* (j, ∂_j) *stating that* $d_{jc;t-\partial_j}$, $\partial_j \in \partial^*$ *belong to the data vector* $\Psi_{c;t}$.

Structure of a parameterized component *is an ordered list of factors creating it, the order characterizes the chosen permutation (5.9).*

Structure of the parameterized mixture *is the list of parameterized components creating it.*

Remark(s) 5.2.1

1. *The adopted dynamic mixture model is not sufficiently general. The component weights should also depend on the state vector. The choice is driven by our inability to estimate this "natural" and more realistic model. This important aspect is discussed more deeply in Section 5.3.*

2. *The modelled p-data may contain a part that is not in the data space of the o-system. Some of them might just bring complementary information and their evolution need not to be modelled. They are used only as entries in the state vectors ϕ_t. In this way, the introduced non-modelled data may arise.*

   It is worth of stressing that their use is rather limited, more or less to one-step-ahead predictions. Otherwise, such variables have to be modelled as their predictions are needed.

3. *Redundancy and contradictions in specification of structures have to be checked when being specified.*

4. *Let us assume that*
$$d_t \sim \mathcal{N}_{d_t}\left(\begin{bmatrix} \theta \\ 0 \end{bmatrix}, \begin{bmatrix} r_1 & r_{12} \\ r_{12} & r_2 \end{bmatrix} \right), \text{ where}$$
   d_t is a two-dimensional vector, $\mathcal{N}_d(\mu, R)$ is normal pdf of d with the expectation μ and covariance matrix R, $\theta \neq 0$ is a scalar parameter determining μ.

   It is straightforward to show that both possible factorized parameterization differ in the number of parameters whenever $r_{12} \neq 0$. This provides counter-example against the hypothesis that number of parameters to be estimated does not depend on the order of factors. Thus, the notation of the component structure as ordered list of factors *is meaningful.*

5. *Formally, the measured data, the mixture model and a suitably specified prior pdf allow us to get the needed model of the o-system $f(d(\mathring{t}))$ through the Bayesian estimation and prediction. Practically, the estimation of mixture models on tens of thousand of records d_t with tens of entries is computationally very intensive. Thus, we are forced to use an approximate learning developed in detail in Section 6.5.*

5.3 Mixtures as approximate models and predictors

The considered mixture has dynamic components but constant weights. They express that the c-th component of the o-system is active for $100 \times \alpha_c\%$ of the operating time. The adopted model (5.8) allows mutually and past-state independent changes of active components at any time moment. For dynamic mixtures, this is an "unnatural" choice. The restriction to constant component weights has a pragmatic motivation discussed here together with a proposal how to weaken this limitation.

Let us consider that the parameterized model is a projection of a more complex model labelled by Θ and by an additional parameter $M \in M^* \equiv \cup_{c \in c^*} M_c^*$ where the finite collection of sets $\{M_c^*\}_{c \in c^*}$ covers M^*, i.e. $M_c^* \cap M_{\tilde{c}}^* = \emptyset$ for $c \neq \tilde{c}$. Then,

$$\begin{aligned} f(d_t|d(t-1), \Theta) &= \int_{M^*} f(d_t, M|d(t-1), \Theta)\, dM = \\ &= \int_{M^*} f(d_t|d(t-1), \Theta, M) f(M|d(t-1), \Theta)\, dM = \\ &= \sum_{c \in c^*} \int_{M_c^*} f(d_t|d(t-1), \Theta, M) f(M|d(t-1), \Theta)\, dM. \end{aligned} \tag{5.12}$$

Let us assume that for each $c \in c^*$ there is $M_c \in M_c^*$ for which the following approximation

$$f(d_t|d(t-1), \Theta, M) \approx f(d_t|d(t-1), \Theta, M_c) \equiv f(d_t|d(t-1), \Theta_c, c)$$

is good for all $M \in M_c^*$ and for all possible data sequences $d(t),\ t \in t^*$. Using this assumption, we arrive to the finite mixture with data-dependent weights

$$f(d_t|d(t-1), \Theta) \approx \sum_{c \in c^*} f(d_t|d(t-1), \Theta_c, c) \tilde{\alpha}_c(d(t-1), \Theta), \text{ where} \tag{5.13}$$

$\tilde{\alpha}_c(d(t-1), \Theta) = \int_{M_c^*} f(M|d(t-1), \Theta)\, dM \equiv \text{Prob}(M \in M_c^*|d(t-1), \Theta)$. The approximating mixture on the right-hand side of (5.13) has to be pdf. The individual components are pdfs, thus the right-hand side of (5.13) becomes pdf iff we assume $\sum_{c\in c^*}\tilde{\alpha}_c(d(t-1),\Theta)=1$ for all $d(t-1)$. In order to make the consequences of this condition explicit, we write $\tilde{\alpha}_c(d(t-1),\Theta)$ as normalized product of constant probabilistic weights α_c and non-negative parameterized functions of data $\beta_c(d(t-1),\Theta)$

$$\tilde{\alpha}_c(d(t-1),\Theta)=\frac{\alpha_c\beta_c(d(t-1),\Theta)}{\sum_{\tilde{c}\in c^*}\alpha_{\tilde{c}}\beta_{\tilde{c}}(d(t-1),\Theta)}.$$

The mixtures with constant component weights are obtained if the functions $\{\beta_c(\cdot)\}_{c\in c^*}$ are constant. If they really depend on data, we get the mixture

$$f(d_t|d(t-1),\Theta)=\sum_{c\in c^*}\alpha_c\frac{f(d_t|d(t-1),\Theta_c,c)\beta_c(d(t-1),\Theta)}{\sum_{\tilde{c}\in c^*}\alpha_{\tilde{c}}\beta_{\tilde{c}}(d(t-1),\Theta)}. \tag{5.14}$$

The components of the mixture (5.14) do not, however, belong to exponential family so that their efficient estimation on large data sets is practically impossible. Consequently, no efficient algorithm for estimation of the overall mixture is known.

We show that our restricted model (5.8) with constant weights can be interpreted as a limit of the model (5.13).

Proposition 5.3.1 (Relationship of constant-weights mixture to the approximation (5.13)) *Weights of the approximating pdf (5.13) converge almost surely to constant values. Thus, the adopted model with constant weights can be seen as an asymptotic version of the "correct" approximating pdf (5.13).*

Proof: For a fixed c and Θ, it holds

$$\begin{aligned}\mathcal{E}[\tilde{\alpha}_c(d(t),\Theta)|d(t-1),\Theta] &\equiv \int f(M_c^*|d(t))f(d_t|d(t-1),\Theta)\,dd_t \underbrace{=}_{chain\ rule}\\ &= \int \frac{f(M_c^*,d_t|d(t-1),\Theta)}{f(d_t|d(t-1),\Theta)}f(d_t|d(t-1),\Theta)\,dd_t \underbrace{=}_{canceling} \int f(M_c^*,d_t|d(t-1),\Theta)\,dd_t \underbrace{=}_{marginalization}\\ &= f(M_c^*|d(t-1),\Theta)\equiv\tilde{\alpha}_c(d(t-1),\Theta).\end{aligned}$$

Thus, $\{\tilde{\alpha}_c(d(t),\Theta),d(t)\}_{t\in t^*}$ is martingale, which is moreover non-negative and bounded by 1 as $\tilde{\alpha}_c(d(t),\Theta)$ is probability. Thus, the martingale convergence theorem [27] applies and $\tilde{\alpha}_c(d(t),\Theta)$ converges almost surely to a constant probability. □

The mixture serves mainly as the predictor of the future behavior of the o-system. The assumed invariance of weights may deteriorate its quality substantially. It can be seen on a simple mixture with a well separated pair of components of similar symmetric shapes and equal weights. For such a mixture, the expected value, which is taken as a good point estimate of the future values d_t, sits in the improbable area between them. The following proposition shows how the problem can be resolved in a generic way that fits to the considered context.

Proposition 5.3.2 (Mixtures on grouped data) *Let us decompose the data sequence $d(\mathring{t})$ into adjacent non-overlapping groups of a length $n>1$. Thus, we consider the following probabilistic description of the whole sequence (with a negligible exception of initial and terminal groups of the length n)*

$$f(d(\mathring{t})|\Theta)=\prod_{1\le\tau\le\mathring{t}/n}f(d_{[(\tau-1)n+1]\_\tau n}|d((\tau-1)n),\Theta). \tag{5.15}$$

Let the individual parameterized pdfs have the mixture form with constant component weights

$$f(d_{[(\tau-1)n+1]\_\tau n}|d((\tau-1)n),\Theta)=\sum_{c\in c^*}\alpha_c f(d_{[(\tau-1)n+1]\_\tau n}|d((\tau-1)n),\Theta_c,c). \tag{5.16}$$

Then, the predictor of $d_{\tau n}$ based on $d(\tau n-1)$ (notice braces!) has data dependent weights. Specifically,

$$\begin{aligned}f(d_{\tau n}|d(\tau n-1),\Theta) &= \sum_{c\in c^*}\tilde{\alpha}_c(d(\tau n-1),\Theta)f(d_{\tau n}|d(\tau n-1),\Theta_c,c) \qquad (5.17)\\ \tilde{\alpha}_c(d(\tau n-1),\Theta) &= \frac{\alpha_c f(d_{[(\tau-1)n+1]\_(\tau n-1)}|d((\tau-1)n),\Theta_c,c)}{\sum_{\tilde{c}\in c^*}\alpha_{\tilde{c}}f(d_{[(\tau-1)n+1]\_(\tau n-1)}|d((\tau-1)n),\Theta_{\tilde{c}},\tilde{c})}.\end{aligned}$$

Proof: The derived formula is directly implied by chain rule for pdfs. □

Remark(s) 5.3.1

Note that we predict the data at the end of the grouping interval. It is sufficient for the assumed low rate of operator interventions for which the predictions serve. The other data records in the group can be predicted similarly but the very first data record in the group is still predicted with constant weights.

Problem 5.3.1 (Mixtures at factor level) *The mixture modelling of grouped data indicates that mixtures can be used at various level of decomposition of the pdf $f(d(\mathring{t}))$. For instance, mixture modelling at factor level could bring an additional freedom.*

Problem 5.3.2 (Shifted and repeatedly used predictors) *Poorer predictions of "earlier" entries in the group can be suppressed by dealing with n mutually shifted predictors or by using the single one working on shifted data. Experiments indicate that these techniques are worth of elaborating.*

Problem 5.3.3 (Rational approximations) *The discussed way of getting data-dependent weights seems to be acceptable. In spite of this, a theoretical and algorithmic solution admitting rational forms instead of finite mixtures would be highly desirable. Approximate estimation developed in Section 6.5 could be probably extended by applying it both to numerator and denominator of the "rational" approximation. An attempt in this direction is made in Section 13.4.*

5.4 Design of advisory system

Here, details related to the optimization part of the basic design Algorithm 5.1.1 are presented.

5.4.1 Advisory systems

Under the adopted notations and assumptions, we are able to specify a formal description of the advisory systems and to distinguish its basic types.

Agreement 5.4.1 (Advisory systems) *The advisory system is a system that constructs and presents the ideal pdf ${}^{\lfloor I}f(d_p(\mathring{t})|\mathcal{P})$ to the operator. A specification of the available experience $\mathcal{P}$ categorizes basic advisory systems.*

The fixed advisory system *constructs the ideal pdf in off line mode using the experience $\mathcal{P} \equiv d_p(0)$, cf. Chapter 2, obtained before the use of the advisory system in its on line mode. Thus, its construction is formally described by a mapping*

$$d_p^*(0) \to \left\{ {}^{\lfloor I}f(d_p(\mathring{t})|d_p(0)) \right\}.$$

The adaptive advisory system *extends its experience during its on line use. Thus, its construction is formally described by the sequence of mappings*

$$\left\{ d_t^* \equiv (d_p^*(0), d_p^*(t)) \to \left\{ {}^{\lfloor I}f(d_p((t+1)\_\mathring{t})|d_p(t)) \right\} \right\}_{t \in \{0\} \cup t^*}.$$

The use of the advisory system in on line mode consists of a sequential presentation of the ideal pdf. It is evaluated at measured and (or) contemplated arguments $d_o(t)$ as well as at the measured values $d_{p+}(t),\ t \in t^*$.

Remark(s) 5.4.1

1. *The presented or contemplated variables have to belong to the data space d_o^* of the o-system.*
2. *Advices, the actions of the p-system, are presented to the operator. Thus, they have to be projected in some way on the data space d_o^* of the operator.*

Before describing *advices* as actions of the p-system, we summarize conditions under which the design of the advising part is performed.

Agreement 5.4.2 (Design conditions of the advisory system)

1. *Systematic design of an optimal advisory system is addressed assuming that quantitative indicators, called* quality markers, *can be evaluated using data that are at disposal to the p-system. The quality markers* $m(\mathring{t}) \in m^*(\mathring{t})$ *qualify behavior of the o-system through the known multiple-valued function*
$$\mathcal{M} : d^*(\mathring{t}) \rightarrow m^*(\mathring{t}) \equiv \textit{(partially) ordered space.} \quad (5.18)$$
2. *Desired properties of quality markers are assumed to be "translated" into the true user ideal pdf* ${}^{\lfloor U}f(d_o(\mathring{t}))$.
3. *Learning provides mixture-based, one-stage-ahead predictors* $f(d_t|d(t-1))$ *used in on line mode.*
For the fixed advisory system, *they have the form*
$$f(d_t|d(t-1)) = \int f(d_t|d(t-1), \Theta) f(\Theta|d(0))\, d\Theta, \ \textit{where}$$
the Bayesian parameter estimate $f(\Theta|d(0))$ *is gained from the "historical" off line data* $d(0)$.
For the adaptive advisory system, *they have the form*
$$f(d_t|d(t-1)) = \int f(d_t|d(t-1), \Theta) f(\Theta|d(t-1))\, d\Theta, \ \textit{where}$$
the Bayesian parameter estimate $f(\Theta|d(t-1))$ *is the posterior pdf* <u>*initiated*</u> *by the prior pdf* $f(\Theta|d(0))$ *gained from the historical off line data* $d(0)$.
4. *Optimization of the advising part has to be formulated so that the ideal pdf presented is reachable by co-operating operators.*

Generally, the p-system may not be aware which variables in $d_o^*(\mathring{t})$ are operator actions. For instance, the operator can change both pressure and temperature of a managed gas system. The advisory system measures changes both of them but may have no information on the command button pressed by the operator.

This incomplete knowledge is another important difference that makes the design of the advisory system a very specific decision-making problem. We found useful to single out the case when this information lack is complete.

Agreement 5.4.3 (Academic advisory system; academic design) *The advisory system designed without knowing which entries of* $d_{o;t}$ *belong to the action space of the operator* $a_o^*(\mathring{t})$ *is called* academic advisory system. *The corresponding design is called* academic design.

Agreement 5.4.4 (Industrial advisory system; industrial & simultaneous designs) *The advisory system designed with knowledge of a non-empty part of the action space of the operator, say* $u_o^*(\mathring{t}) \subset a_o^*(\mathring{t})$, $\emptyset \neq u_o^*(\mathring{t})$, *is called* industrial advisory system. *The o-actions* $u_{o;t}$ *are called* recognizable actions. *The design of recognizable actions is called* industrial design. *The joint academic and industrial design is called* simultaneous design.

Whenever possible, the industrial advisory system is optimized through the simultaneous design.

5.4.2 Advices as actions of the p-system

Here we classify actions available to the p-system.

Agreement 5.4.5 (Nomenclature of actions of the p-system) *The actions available to p-systems*
$$a_{p;t} \equiv (c_t, u_{o;t}, s_t, p_t) \ \ \textit{are interpreted as follows.} \quad (5.19)$$

Recommended pointers $\{c_t\}_{t\in t^*}$ *are pointers to the components that are recommended to be kept active at respective time moments.*

Recommended pointers are academic advices.

Academic advices cannot be directly presented to the operator and have to be reflected in the ideal pdf $^{\lfloor I}f(d_{o;t}|d(t-1))$ *offered to him.*

Recommended recognizable actions $\{u_{o;t}\}_{t\in t^*}$ *guide the user in selecting recognizable actions.*

These advices result from the industrial or simultaneous designs.

The recommended recognizable actions can be interpreted as ordinary inputs of the o-system with the operator serving as an imperfect actuator.

Ideally, the recommended recognizable actions should be directly feeded into the o-system. This motivates the identical notation is being used for of the recommended recognizable actions and recognizable actions.

Priority actions $\{p_t\}_{t\in t^*}$ *select entries of* $\{d_t\}_{t\in t^*}$ *shown to the operator.*

These advices result from the design called assigning priorities.

The priority action p_t *is* $\mathring{p}$ *vector* ($\mathring{p} \leq \mathring{d}_o$) *of differing indices* $p_{i;t} \in \{1,\ldots,\mathring{d}_o\},\ i \in \{1,\ldots \mathring{p}\}$.

The user is presented the marginal pdfs $\left\{ {}^{\lfloor I}f(d_{p_t;t}|d(t-1)\right\}_{t\in t^*}$ *of the ideal pdf resulting from the previous design.*

Signaling actions $\{s_t \in s^* \equiv \{0,1\}\}_{t\in t^*}$ *stimulate the operator to take some measures when behavior of the o-system significantly differs from the desired one.*

These advices result from the design called signaling.

The user is presented probabilities of the need to be alert. They are coded, for instance, as traffic lights.

5.4.3 Model of non-guided and guided of the o-system

We assume that the learning part of the advisory system provides us a good model $f(d_t|d(t-1))$ of the o-system uninfluenced by the p-system. It is the model of the *non-guided o-system.*

A systematic design of the p-system requires a model relating its advices to responses of the o-system. We need model of the *guided o-system.* A *speculative class of models* is proposed here. The term "speculative" underlines that the model is based on hypothesis that operator fully co-operates and is able to lead the o-system to the desired state.

Consequences of this speculation can be made weaker in adaptive advisory system but at the off line design stage. We have no other way leading to the systematic design.

Advices $a_{p;t}$ try to influence the o-system so that the management aims are achieved in a better way. It is done by designing an ideal description of the o-system ${}^{\lfloor I}f(d(\mathring{t}))$. This ideal pdf is made, unlike the user ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$, so that it can be realized. Its feasibility is implied by the collected experience and by the adopted constructions. We restrict our designs so that they change in $f(d(\mathring{t})) = \prod_{t\in t^*} f(d_t|d(t-1))$ only those elements that are, at least indirectly, under the operator control. The ideal pdf is shown to the operator, i.e. advices are feeded into the o-system via the operator.

The guided model ${}^{\lfloor I}f(d(\mathring{t}))$ is gradually build in forthcoming sections. We describe individually designs of good strategies for the academic, industrial and simultaneous cases. Then, signaling strategies and strategies generating the presentation priorities are discussed.

5.4.4 Academic design

We have at disposal the multiple-mode model of the o-system $f(d(\mathring{t}))$. As discussed above, see Agreement 5.4.1, the advisory system maps $f(d(\mathring{t}))$ on an ideal pdf ${}^{\lfloor I}f(d(\mathring{t}))$, whose projections are presented to the operator.

The joint pdf $f(d(\mathring{t}))$ describes the probability distribution of achievable modes within the data space of the advisory system. Thus, the reachable ideal pdf should be created from these modes. The selection of modes leading to a higher management quality should be advised. The academic design selects the recommended mode through the recommended pointer $c_t \in c^*$ by defining the ideal pdf

$${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t})) \equiv \prod_{t \in t^*} {}^{\lfloor I}f(d_t, c_t|d(t-1)) \equiv \prod_{t \in t^*} f(d_t|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1)).$$

The pdfs $f(d_t|d(t-1), c),\ c \in c^*$ are estimated components and the optional probabilities ${}^{\lfloor I}f(c_t|d(t-1))$ describe the randomized causal strategy

$$\{d^*(t-1) \to c_t \in c^*\}_{t \in t^*}. \quad \text{to be designed.} \tag{5.20}$$

Remark(s) 5.4.2

Use of the notation ${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ is a bit inconsistent as $c(\mathring{t})$ is a part of $d(\mathring{t})$. It helps, however, to focus attention on the discussed advices.

The recommended pointer $c_t \in d^*_{p+;t}$, thus, a projection on d^*_o has to be presented to the operator. The corresponding marginal predictive pdf has the form

$${}^{\lfloor I}f(d_{o;t}|d(t-1)) = \sum_{c \in c^*} {}^{\lfloor I}f(c_t|d(t-1)) f(d_{o;t}|d(t-1), \Theta_c, c). \tag{5.21}$$

The advising strategy $\left\{ {}^{\lfloor I}f(c_t|d(t-1)) \right\}_{t \in t^*}$ is optimized in the sense of the fully probabilistic design, see Section 2.4.2. For that, we have to specify a user *ideal for the inter-connection of the o- and p-systems* ${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) \equiv {}^{\lfloor I}f(d_t|d(t-1))\, {}^{\lfloor U}f(c_t|d(t-1))$. The constructed ideal pdf ${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t})) = \prod_{t \in t^*} {}^{\lfloor I}f(d_t|d(t-1))\, {}^{\lfloor U}f(c_t|d(t-1))$ should be as close as possible to the pdf ${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t}))$.

The recommended pointers c_t belong to $d^*_{p+;t}$ so that the general extension of the user ideal pdf, see Section 5.1.3, could be used. Practical reasons make us to take this part of the "user" ideal pdf as tuning knob of the design. For instance, it is reasonable to inhibit fast changes of values of $c(\mathring{t})$ in order to get relatively "stable" advices given to the operator. This can be achieved by selecting such "user" ideal pdf

$$\prod_{t \in t^*} {}^{\lfloor U}f(c_t|d(t-1))$$

that assigns high probabilities to sequences $c(\mathring{t})$ with small differences $c_t - c_{t-1}$. Also, an off line analysis may discourage operating at some dangerous modes of $f(d(\mathring{t}))$ completely. For that, it is sufficient to restrict support of ${}^{\lfloor U}f(c_t|d(t-1))$ on pointers to the permitted components. Such considerations determine the discussed part of the user ideal pdf

$$\left\{ {}^{\lfloor U}f(c_t|d(t-1)) \right\}_{t \in t^*}.$$

With it, the complete user ideal pdf gets the form

$${}^{\lfloor U}f(d(\mathring{t})) \quad = \quad \prod_{t \in t^*} {}^{\lfloor I}f(d_{p+;t}|d(t-1))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|d(t-1)). \tag{5.22}$$

It follows from the discussed extension of the "true" user ideal ${}^{\lfloor U}f(d_o(\mathring{t}))$ to ${}^{\lfloor U}f(d(\mathring{t}))$, see Section 5.1.3, from the chain rule and the fact, that the user can deal with variables about which he is aware of.

This selection of the extended user ideal completes the formulation of the academic design that makes the general fully probabilistic design formally applicable, see Proposition 2.4.4. The practical evaluation requires approximations that are discussed in Chapter 7.

As mentioned above, some components might be handled as "dangerous". We take a component as dangerous if a permanent operation on it leads, in a long run, to an unacceptable behavior of the o-system.

Agreement 5.4.6 (Dangerous component) *Let $f(\Psi|c)$ be the* steady state pdf *of the data vector Ψ assigned to the permanent activity of the c-th component of the mixture*

$$f(\Psi|c) = \int f(\Psi|\tilde{\Psi}, c) f(\tilde{\Psi}|c)\, d\tilde{\Psi}. \tag{5.23}$$

Here, $f(\Psi|\tilde{\Psi}, c) \equiv f(\Psi_t = \Psi|\Psi_{t-1} = \tilde{\Psi}, c)$ is a formal, state-space version of the c-th component that describes the evolution of the data vector Ψ_t.

Let us consider average marker *of the form*

$$\mathcal{M}(d(\mathring{t})) = \frac{1}{\mathring{t}} \sum_{t \in t^*} m(\Psi_t) \tag{5.24}$$

given by a partial marker $m(\Psi_t)$. Then, the component c is called dangerous if the probability of a given set $\bar{m}^*$ of non-acceptable values of $m(\Psi)$

$$\int \chi_{\bar{m}^*}(m(\Psi)) f(\Psi|c)\, d\Psi \tag{5.25}$$

is too high. The symbol $\chi_{x^*}(\cdot)$ means indicator of the set x^*.

5.4.5 Industrial design

The industrial design optimizes recommended recognizable actions. Ideally, these actions are directly feeded into the o-system and their consequences are predicted by the model of the o-system. They are similar to ordinary inputs of the o-system with the operator serving as an imperfect actuator.

The constructed randomized strategy is described by the pdfs $\left\{ {}^{\lfloor I}f(u_{o;t}|d(t-1)) \right\}_{t \in t^*}$. These pdfs replace $\{f(u_{o;t}|d(t-1))\}_{t \in t^*}$ forming a part of the estimated, non-guided, o-model $f(d_t|d(t-1))$. Thus, the ideal pdf generated by this design has the form

$$\begin{aligned} {}^{\lfloor I}f(d_t|d(t-1)) &= f(\Delta_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1)) \\ &\equiv {}^{\lfloor I}f(u_{o;t}|d(t-1)) \frac{\sum_{c \in c^*} \alpha_c f(\Delta_t|u_{o;t}, d(t-1), c) f(u_{o;t}|d(t-1), c)}{\sum_{c \in c^*} \alpha_c f(u_{o;t}|d(t-1), c)}. \end{aligned} \tag{5.26}$$

The strategy determining the optimal ${}^{\lfloor I}f(u_{o;t}|d(t-1))$ is obtained through the fully probabilistic design, Proposition 2.4.4. The needed user ideal pdf

$${}^{\lfloor U}f(d(\mathring{t})) = \prod_{t \in t^*} {}^{\lfloor U}f(\Delta_t|u_{o;t}, d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d(t-1)),$$

that includes the target for the recognizable actions, is constructed exactly as described in Section 5.1.3.

Remark(s) 5.4.3

1. *Note that $u_{o;t} \in d^*_{o;t}$, thus it is the only advice that can be directly presented to the operator.*

2. *The estimated strategy, generating the recognizable actions of the non-guided o-system and described by the marginal pdfs $f(u_{o;t}|d(t-1), c)$ of individual components, influences the resulting ideal pdf, see (5.26). This effect is specific for non-trivial mixtures, in which the strategies used at various components do not cancel in the inspected conditional pdf (5.26).*

3. *Properties of the components creating the ideal pdf are influenced by the advising strategy $\{ {}^{\lfloor I}f(u_{o;t}|d(t-1))\}_{t \in t^*}$ influencing the recognizable actions $u_o(\mathring{t})$ applied. It may, for instance, convert the dangerous components in non-dangerous ones and vice versa.*

5.4.6 Simultaneous academic and industrial design

The simultaneous design optimizes both recommended pointers to components as well as recommended recognizable actions. It should lead to a better advising strategy than a sequential use of academic and industrial designs. The simultaneous design is also surprisingly simpler than the industrial one, see Chapter 7.

The ideal pdf ${}^{\lfloor I}f(d_t|d(t-1))$, generated by the fully probabilistic design, Proposition 2.4.4, minimizes the KL distance to the user ideal (target) pdf

$$^{\lfloor U}f(d(\mathring{t})) = \prod_{t\in t^*} {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\, {}^{\lfloor I}f(\Delta_{p+;t}|u_{o;t}, d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))$$

that includes the user ideal pdf ${}^{\lfloor U}f(u_{o;t}|d_o(t-1))$ for the recognizable actions $u_{o;t}$ as well as the target probability ${}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))$ for the recommended pointers. The latter element is discussed in connection with the academic design, Section 5.4.4. In simultaneous design, it may depend on $u_{o;t}$, too.

The result is similar to (5.26) with the component weights replaced by the designed probabilities ${}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))$ of recommended actions. Specifically,

$$\begin{aligned} {}^{\lfloor I}f(d_t|d(t-1)) &= {}^{\lfloor I}f(\Delta_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1)) = \\ \equiv\quad & {}^{\lfloor I}f(u_{o;t}|d(t-1))\frac{\sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))f(\Delta_t|u_{o;t}, d(t-1), c_t)f(u_{o;t}|d(t-1), c_t)}{\sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))f(u_{o;t}|d(t-1), c_t)}. \end{aligned} \tag{5.27}$$

Remark(s) 5.4.4

It is worth of repeating that the elements in the formula (5.27) with the superscript ${}^{\lfloor I}$ are optimized. The elements without the superscript ${}^{\lfloor I}$ are those obtained through estimation of the non-guided o-system. They reflect those parts of operating practice that are not expected to be changed by the advising.

5.5 Interactions with operator

The discussed advisory systems can be seen as specific versions of a high-level control system. The ideal pdf ${}^{\lfloor I}f(d(\mathring{t}))$ resulting from the designs discussed above has to be, however, perceived by a human being. This implies a non-standard task, namely, the optimal *presentation of advices*. Essentially, low dimensional projections of the high-dimensional ideal pdf have to be shown to the operator. This calls for generating additional actions of the p-system. Specifically,

- Recommended ranges of a few selected variables can only be shown. Presentation priorities have to be assigned.
- The choice of variables to be changed has to be made. Formally, it coincides with designing of presentation priorities but it is more tightly related to the effort to minimize risk of malfunctioning or maximize benefit.
- The need for making the operator to act has to be indicated. Signaling strategy has to be designed.

These problems are addressed in subsequent Sections 5.5.1, 5.5.2 under typical advising scenarios.

5.5.1 Assigning priorities

Presentation of variables worth of the operator interest is simple when priorities of critical variables to be shown are fixed by technological prescriptions. The situation is also simple if a full question and answer mode of the dialog is adopted. In this case, the operator asks the p-system: "What happens to a variable $d_{i;t}$ if I assign a value $\bar{d}_{j;t}$ to the variable $d_{j;t}$?" In this case, the predictive pdf $f(d_{i;t}|\bar{d}_{j;t}, d(t-1))$ is simply shown. It is gained from the model $f(d_t|d(t-1))$ by marginalization and conditioning. Similarly, a simple case arises when a few recognizable actions $u_{o;t}$ are provided by the industrial advisory system. Then, marginal pdfs ${}^{\lfloor I}f(u_o|d(t-1))$ are to be provided as well.

The situation becomes more complex when there is the need or freedom to show only marginal pdfs of a few critical variables contained in the full record d_t. It leads to a specific decision problem.

The p-system is given task to generate a $\mathring{p}$-vector of priority actions p_t with entries $p_{i;t} \in \{1, \ldots, \mathring{d}_o\}$, $i \in i^* \equiv \{1, \ldots, \mathring{p}\}$, $\mathring{p} < \mathring{d}_o$. The value of $p_{i;t} = j$ means that j-th entry of $d_{o;t}$ should be presented to the operator.

Note that the number of shown variables $\mathring{p}$ has to be small, say 5, in order to respect limited perceiving abilities of human beings.

The ideal pdf ${}^{\lfloor I}f(d(\mathring{t}))$ resulting from academic, industrial or simultaneous designs as well as the user ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$ are assumed to be fixed and available when designing the presentation strategy $\left\{ {}^{\lfloor I}f(p_t|d(t-1))\right\}_{t\in t^*}$.

Similarly as in the academic design, the target pdf for the adopted fully probabilistic design ${}^{\lfloor U}f(d(\mathring{t}))$ is extended by the factor $\prod_{t\in t^*} {}^{\lfloor U}f(p_t|d(t-1))$. It is a tuning knob that allows us to enforce technological preferences or wishes to minimize changes while selecting the presented variables.

The designed p-actions are feeded "through" the operator to the inter-connection of the p- and o-systems. Let the co-operating operator be presented the target pdf for entries of d_t with indices p_t. Then the operator is expected to act so that the resulting description of this inter-connection ${}^{\lfloor I}f(d(\mathring{t}))$ approaches the pdf with the marginal pdf $f(d_{p_t;t}|d(t-1)) = {}^{\lfloor I}f(d_{p_t;t}|d(t-1))$, while entries not shown follow the model of the non-guided o-system. Here, the pdf ${}^{\lfloor I}f(d_t|d(t-1))$ is the ideal pdf offered by the advisory system, i.e. resulting from the previous optimization. This discussion explains why the consequences of individual presentations are modelled as follows, cf. Remark 5.4.2,

$$ {}^{\lfloor I}f(d(\mathring{t}), p(\mathring{t})) = \prod_{t\in t^*} {}^{\lfloor I}f(d_t|p_t, d(t-1))\, {}^{\lfloor I}f(p_t|d(t-1)), \text{ where} \tag{5.28} $$

the pfs $\left\{ {}^{\lfloor I}f(p_t|d(t-1))\right\}_{t\in t^*}$ describe the randomized *presentation strategy* to be designed.

The compromise ${}^{\lfloor I}f(d_t|p_t, d(t-1))$ between the *non-guided o-system* $f(d_t|d(t-1))$, acting without the p-system, and the *guided o-system* ${}^{\lfloor I}f(d_t|d(t-1))$, acting fully according to the p-system, is modelled as follows

$$ {}^{\lfloor I}f(d_t|p_t, d(t-1)) = f(d_t|d(t-1)) \frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))}. \tag{5.29} $$

In this model, the marginal pdf of the presented variables – computed from the estimated mixture – is replaced by the corresponding marginal pdf gained from the designed ideal pdf ${}^{\lfloor I}f$, which is generally also mixture. This makes the model rather complex as it becomes ratios of mixtures. Consequently, the fully probabilistic design must be approximated in order to get a feasible presentation strategy, cf. Chapter 7. Moreover, the number of variants to be compared $\binom{\mathring{d}_o}{\mathring{p}}$ is mostly large. It motivates us to select $\mathring{p} = 1$ and to use ${}^{\lfloor I}f(p_t = i|d(t-1))$, $i \in \{1, \ldots, \mathring{d}_o\}$ as a degree of the presentation priority for the i-th entry of d_t.

Problem 5.5.1 *Alternative design of presentation priorities This part of the design is, a bit surprisingly, the hardest one. It calls for an alternative formulation. For instance, it would be possible to perform other designs with a fixed choice of presented variables and then to compare the predicted quality of the guided closed loop behavior. This formulation is worth of considering.*

5.5.2 Stimulating the operator

Operator may actively call the p-system for advices. Typically, however, the operator has other duties. Then, his attention has to be attracted when the state of the managed system requires it. This problem is addressed here.

The p-system is given the task to generate scalar actions $\{s_t\}_{t\in t^*}$, $s_t \in \{0, 1\}$. The value $s_t = 0$ means that system is in a good state and no extra operator activity is needed. The value of $s_t = 1$ urgently demands operator actions. We search for an admissible strategy, called *signaling strategy* given by the causal rules

$$\{d^*(t-1) \rightarrow s_t^* \equiv \{0,1\}\}_{t\in t^*}.$$

The ideal pdf ${}^{\lfloor I}f(d(\mathring{t}))$ as well as the user ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$ in the fully probabilistic design are assumed to be fixed. The user ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$ is extended to signaling actions $s(\mathring{t})$ by the pf ${}^{\lfloor U}f(s_t|d(t-1))$ reflecting the desired damping of the stimulation.

Stimulation, when respected by the operator, modifies the non-guided behavior of the o-system $f(d_t|d(t-1)) \equiv f(d_t|s_t = 0, d(t-1))$ to the behavior described by the ideal pdf ${}^{\lfloor I}f(d_t|d(t-1)) \equiv f(d_t|s_t = 1, d(t-1))$ that arises from the academic, industrial or simultaneous design of the p-system. Thus, the resulting model of the guided o-system has the form

$${}^{\lfloor I}f(d(\mathring{t}), s(\mathring{t})) = \prod_{t\in t^*} {}^{\lfloor I}f(d_t|s_t, d(t-1))\, {}^{\lfloor I}f(s_t|d(t-1)), \text{ where}$$

the probabilities $\left\{ {}^{\lfloor I}f(s_t|d(t-1))\right\}_{t\in t^*}$ describe the constructed *signaling strategy* and ${}^{\lfloor I}f(d_t|s_t, d(t-1)) = \delta_{s_t,0} f(d_t|d(t-1)) + (1-\delta_{s_t,0})\, {}^{\lfloor I}f(d_t|d(t-1))$.

As $s_t \in d^*_{p+;t}$, it cannot be directly presented to the operator. Instead,

$$\hat{s}_t = \mathcal{E}[\delta_{s_t,0}|d(t-1)] = {}^{\lfloor I}f(s_t = 0|d(t-1))$$

is shown, usually, as a colored traffic light. Here, *Kronecker symbol* is used

$$\delta_{s,\tilde{s}} = \begin{cases} 1 & \text{if } s = \tilde{s} \\ 0 & \text{otherwise} \end{cases}.$$

5.6 Design summary

5.6.1 Influence of advices on the o-system

During the discussion of particular designs, the following overall model of the interconnecting of the o-system and the p-system has arisen. Recall, the superscript ${}^{\lfloor I}$ indicates that the corresponding object results from the optimization of advices $a_{p;t} = (c_t, u_{o;t}, p_t, s_t)$

$$f(d_{o;t}|a_{p;t}, d(t-1)) \equiv \delta_{s_t,0} f(d_{o;t}|d(t-1)) + (1-\delta_{s_t,0})\, {}^{\lfloor I}f(d_{o;t}|d(t-1)), \text{ where} \tag{5.30}$$

$$\begin{aligned}
s_t & \quad \text{is 0 if operator do not take recommended actions and otherwise } s_t \text{ is 1.} \\
& \quad \text{He reacts on signaling chosen according to } {}^{\lfloor I}f(s_t|d(t-1)). \\
& \quad \text{The first predictor above is obtained in learning of non-guided model} \\
f(d_{o;t}|d(t-1)) & \equiv \sum_{c\in c^*} \alpha_c f(d_{o;t}|d(t-1), c) \text{ and} \\
{}^{\lfloor I}f(d_{o;t}|d(t-1)) & \equiv f(d_{o;t}|d(t-1)) \frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))}, \text{ where} \\
p_t & \quad \text{is presentation action chosen according to } {}^{\lfloor I}f(p_t|d(t-1)) \\
{}^{\lfloor I}f(d_{o;t}|d(t-1)) & \equiv {}^{\lfloor I}f(u_{o;t}|d(t-1)) \times \\
& \times \frac{\sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1)) f(\Delta_{o;t}|u_{o;t}, d(t-1), c_t) f(u_o|d(t-1), c_t)}{\sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1)) f(u_o|d(t-1), c_t)}.
\end{aligned}$$

The probabilities $\left\{ {}^{\lfloor I}f(s_t|d(t-1))\right\}_{t\in t^*}$, $s_t \in s^* \equiv \{0,1\}$ describe signaling strategy. They result from the corresponding, fully probabilistic, signaling design. This design is the last in the sequence of respective designs.

The probabilities $\left\{ {}^{\lfloor I}f(p_t|d(t-1))\right\}_{t\in t^*}$, $p_t \in p^* \equiv \left\{[p_1, \ldots, p_{\mathring{p}}],\ p_i \in \{1, \ldots, \mathring{d}_o\}\right\}$ describe presentation strategy. They result from the presentation design made after finishing some of designs listed below.

The pdfs $\left\{ {}^{\lfloor I}f(u_{o;t}|d(t-1))\right\}_{t\in t^*}$, $u_{o;t} \in u_o^*$ describe strategy generating recommended recognizable actions. They result from industrial or simultaneous design. The industrial design relies on availability of the component weights $f(c_t|u_{o;t}, d(t-1))$. They are obtained either from learning, then $f(c_t|u_{o;t}, d(t-1)) = \alpha_c$ or from the previous academic design, then $f(c_t|u_{o;t}, d(t-1)) = {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))$.

The simultaneous design – when adopted – is the first in the sequence of designs.

The probabilities $\left\{ {}^{\lfloor I}f(c_t|d(t-1))\right\}_{t\in t^*}$, $c_t \in c^* \equiv \{1,\ldots,\mathring{c}\}$ describe strategy generating academic advices. They result from the academic or simultaneous designs that start the overall sequence of designs.

Remark(s) 5.6.1

Recall that the adopted models relating advices to the behavior of the guided o-system *are of a speculative nature. They serve as the necessary departing point but the model should be corrected through the use of the p-system: we should estimate model $f(d_t|a_{p;t}, d(t-1), \Theta)$ describing explicitly dependence of data on d_t on adopted advices $a_{p;t}$. For it, the use of the adaptive version of the p-system becomes highly desirable.*

5.6.2 Overall scenario and design sub-tasks

The overall scenario of the addressed advising problem and its solution is in Figure 5.1, that anticipates steps elaborated in subsequent Chapters.

The solution of the overall design problem includes a number of particular steps summarized in the algorithm below. The algorithm serves us as a design guide. Its steps are discussed in Chapters 6, 7 at general level and specialized to specific mixture elements in subsequent Chapters. For fixed advisory system, all steps, except steps 16c 17, are made in off line mode. For adaptive advisory system, steps 7, 14, 15 have to be run in on line mode, too. Note that the recursively performed estimation step 7 is computationally cheap and re-designs 14, 15 can be performed with much slower rate without harm.

Algorithm 5.6.1 (Design of the advisory system)

1. *Collect the learning data that have to reflect all modes of operating.*
2. *Select quality markers and express their desirable values as the true user ideal pdf ${}^{\lfloor U}f(d_o(\mathring{t}))$.*
3. *Collect prior physical information on the managed system, operating and measuring conditions.*
4. *Preprocess the learning data by*
   (a) *grouping data records according to the operator rate,*
   (b) *reducing dimensionality using expert knowledge: let them exclude surely irrelevant record entries d_i,*
   (c) *removing outliers,*
   (d) *replacing missing data,*
   (e) *suppressing high frequency noise.*
5. *Select the largest acceptable structure of the estimated mixture.*
6. *Select a prior pdf initializing estimation of the mixture describing the o-system.*
7. *Estimate the mixture, if need be, go back to the step 6.*
8. *Estimate structure of the mixture in its full hierarchy from factors to the whole mixture and, if need be, go back to the step 6.*

9. *Reduce dimensionality of the problem by removing data that do not influence even indirectly quantities in* ${}^{\lfloor U}f(d_o(\mathring{t}))$ *and, if need be, go back to the step 6.*
10. *Use physical prior information for individual factors and, if need be, go back to the step 6.*
11. *Validate quality of the obtained model using expert opinion as well as independent testing data set and if the result is unsatisfactory repeat whole learning procedure, possibly from the step 4.*
12. *Analyze individual components and qualify good and dangerous ones according to the stationary probability that markers will stay in desirable or undesirable areas.*
13. *Select basic advising scenario (academic, industrial or simultaneous).*
14. *Design the advising strategy.*
15. *Design presentation and signaling strategies.*
16. *Validate advising strategy by*
    (a) *comparing real actions of the operator with those generated by advising system in static mode,*
    (b) *judging proximity of good modes in data with behavior of a good operator,*
    (c) *using advising system in full scale.*
17. *Use the p-system in the fixed or adaptive mode by feeding the currently measured data into advising mixture as well as into presentation and signaling strategies. Show the low-dimensional projection of the advising mixture to the operator. The shown variables are selected by the presentation strategy and call for actions is driven by signaling strategy.*

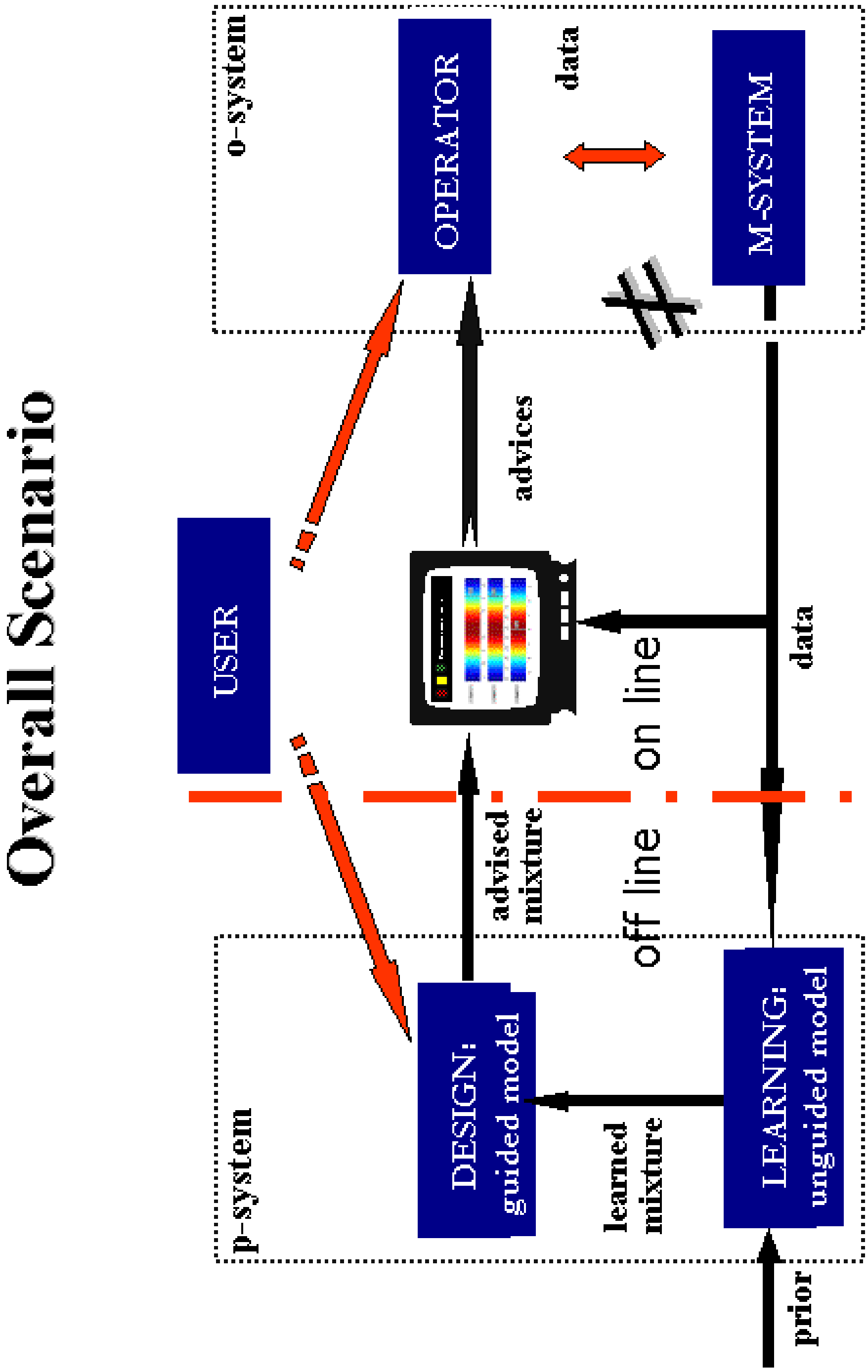


Figure 5.1: Interconnecting the o-system with the structured p-system. Learning and design parts are treated in Chapters 6 and 7, respectively.

Chapter 6

Solution and principles of its approximation: learning part

A formal solution of the design of the advisory system helps us to clarify the structure of evaluations that should be made. Their practical usefulness is restricted by the "curse of dimensionality". Thus, the formal solution has to be complemented by approximate but feasible evaluations. A discussion of their principles for the *learning part of the advisory system* is presented here. Thus, the off line mode is predominantly addressed and data are mostly historical ones.

The treated advisory system is based on the parameterized model of the o-system. Its behavior observed by the p-system, grouped according to the rate of operator actions, is described by the mixture (5.8)

$$f(d_t|d(t-1),\Theta) = f(d_t|\phi_{1\_\mathring{c};t-1},\Theta) = \sum_{c\in c^*} \alpha_c f(d_t|\phi_{c;t-1},\Theta_c,c). \tag{6.1}$$

Each parameterized component $f(d_t|\phi_{c;t-1},\Theta_c,c)$ is decomposed into the product of parameterized factors, see Agreement 5.2.1,

$$f(d_t|\phi_{c;t-1},\Theta_c,c) = \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t},\Theta_{ic},c),\ i^* \equiv \{1,\ldots,\mathring{d}\},\ \text{where} \tag{6.2}$$

the regression vectors $\psi_{ic;t}$ are made of $[d'_{i+1\_\mathring{d}c;t}, \phi'_{c;t-1}]'$. The factors $f(d_{ic;t}|\psi_{ic;t},\Theta_{ic},c)$ are parameterized by individual parameters Θ_{ic}. Collection of these parameters, together with the probabilistic weights of components $\alpha \in \alpha^*$ (5.8), form the multi-variate parameter Θ of the mixture. Parameters, structures of factors and components as well as the structure of the mixture, see Agreement 5.2.1, have to be estimated. The estimation is inspected, here.

Section 6.1 prepares common tools used throughout this chapter. First, the considered class of Bayesian estimates is specified. Then, predictors serving for comparison of alternative estimates are presented in Sections 6.1.1, 6.1.2. The alternatives are generated through versions of branch and bound techniques, see Section 6.1.3.

The formal Bayesian estimation is described by Proposition 2.5.2 with experience formed by the p-data. These data result from preprocessing of raw data observed by the p-system, see Section 6.2.

Application of the Bayesian paradigm requires specification of a prior pdf $f(\Theta)$. Use of a prior knowledge to this purpose forms the content of Section 6.3. Even with such a knowledge available, a proper specification of the prior pdf is non-trivial. An extensive discussion of promising ways is described in Section 6.4.

For a chosen $f(\Theta)$, the evaluations of the likelihood function $\mathcal{L}(\Theta,\mathcal{P}_{a_{t+1}})$ (2.45) and its integral $\mathcal{I}(\mathcal{P}_{a_{t+1}})$ (2.46) represent the main computational burden. The difficulty of the mixture estimation stems from the fact that likelihood function is product of sums of pdfs depending on data $d(t)$ and on unknown parameter Θ. Thus, its formal analytical expression contains huge amount of terms that cannot be handled exactly. For this reason, an approximate treatment is necessary. It is discussed in Section 6.5.

The important structure estimation task is outlined in Section 6.6. Then, Section 6.7 covers model validation. It forms a natural bridge to Chapter 7 that treats design of the advising and presentation strategies, Section 5.4.

6.1 Common tools

Here, tools used throughout this chapter are prepared. Their description requires specification of conditions under which the adopted learning is performed.

Agreement 6.1.1 (Considered forms of pdfs on Θ^*) *Prior $f(\Theta) \equiv f(\Theta|d(0))$ and posterior $f(\Theta|d(t))$ Bayesian estimates of the mixture, Agreement 5.2.1, are considered in the common form*

$$
\begin{aligned}
f(\Theta|d(t)) &= Di_\alpha(\kappa_t) \prod_{i\in i^*, c\in c^*} f(\Theta_{ic}|d(t)),\ t \in \{0\} \cup t^*, && (6.3)\\
Di_\alpha(\kappa_t) &\equiv \mathcal{B}^{-1}(\kappa_t) \prod_{c\in c^*} \alpha_c^{\kappa_{c;t}-1} \chi_{\alpha^*}(\alpha) \equiv \text{ Dirichlet pdf on } \alpha^* \equiv \left\{\alpha_c,\ \sum_{c\in c^*}\alpha_c = 1\right\}\\
\mathcal{B}(\kappa_t) &\equiv \frac{\prod_{c\in c^*}\Gamma(\kappa_{c;t})}{\Gamma\left(\sum_{c\in c^*}\kappa_{c;t}\right)} \equiv \text{ multi-variate beta function given by} && (6.4)\\
\kappa_t &\equiv [\kappa_{1;t},\ldots,\kappa_{\mathring{c};t}]' \in \kappa^* \equiv \{\kappa_t : \kappa_{c;t} > 0\}.
\end{aligned}
$$

In words, parameters Θ_{ic}, $i \in i^* \equiv \{1,\ldots,\mathring{d}\}$, $c \in c^*$, of individual parameterized factors are mutually conditionally independent and independent of the component weights α. The component weights have Dirichlet distribution $Di_\alpha(\kappa)$ with the support on the probabilistic simplex α^*.

Dirichlet distribution $Di_\alpha(\kappa_t)$ is recalled and analyzed in Chapter 10 in detail. Here, we only need to know that the expectation assigned to $Di_\alpha(\kappa_t)$ is

$$
\mathcal{E}\left[\alpha_c|d(t)\right] = \mathcal{E}[\alpha_c|\kappa_t] = \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}} \equiv \hat{\alpha}_{c;t}. \tag{6.5}
$$

6.1.1 Prediction and model selection

The constructed p-system predicts consequences of the observed behavior and operator actions on the future behavior of the o-system. Thus, its quality depends heavily on the quality of the used predictive pdf. Under Agreement 6.1.1, the value of the predictive pdf of a component c at a possible data vector $\Psi_{t+1} = [d'_{t+1}, \phi'_t]'$, Agreement 5.2.1, conditioned on measured data $d(t)$ is

$$
\begin{aligned}
f(d_{t+1}|d(t),c) &\equiv \mathcal{E}\left[\prod_{i\in i^*} f(d_{ic;t+1}|\psi_{ic;t+1},\Theta_{ic},c)|d(t),c\right] = && (6.6)\\
&= \prod_{i\in i^*}\int f(d_{ic;t+1}|\psi_{ic;t+1},\Theta_{ic},c) f(\Theta_{ic}|d(t))\,d\Theta_{ic} = \prod_{i\in i^*} f\left(d_{ic;t+1}|\psi_{ic;t+1},d(t),c\right).
\end{aligned}
$$

The overall predictive pdf of a mixture is

$$
\begin{aligned}
f(d_{t+1}|d(t)) &\equiv \mathcal{E}\left[\sum_{c\in c^*}\alpha_c \prod_{i\in i^*} f(d_{ic;t+1}|\psi_{ic;t+1},\Theta_{ic},c)|d(t)\right] = && (6.7)\\
&= \sum_{c\in c^*}\frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}}\prod_{i\in i^*}\int f(d_{ic;t+1}|\psi_{ic;t+1},\Theta_{ic},c) f(\Theta_{ic}|d(t))\,d\Theta_{ic} =\\
&\underbrace{=}_{(6.5)} \sum_{c\in c^*}\hat{\alpha}_{c;t}\prod_{i\in i^*} f(d_{ic;t+1}|\psi_{ic;t+1},d(t),c).
\end{aligned}
$$

Similarly, we get predictor for the fixed advisory system. For reference reasons, we summarize these predictors in the following proposition.

Proposition 6.1.1 (Mixture-based one-step-ahead predictor) *Under Agreement 6.1.1, the estimation within the adaptive advisory system provides the value of one-step-ahead predictor at a possible data vector* $\Psi_{t+1} = [d'_{t+1}, \phi'_t]'$*, Agreement 5.2.1, in the form*

$$f(d_{t+1}|d(t)) = \sum_{c \in c^*} \hat{\alpha}_{c;t} \prod_{i \in i^*} f(d_{ic;t+1}|\psi_{ic;t+1}, d(t), c). \tag{6.8}$$

The adaptive predictor *of a factor output* $d_{ic;t+1}$

$$f(d_{ic;t+1}|\psi_{ic;t+1}, d(t), c) \equiv \int f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c) f(\Theta_{ic}|d(t))\, d\Theta_{ic} \tag{6.9}$$

eliminates the unknown parameters of the parameterized factor by its integration weighted by the posterior pdf $f(\Theta_{ic}|d(t))$*, by the posterior factor estimate, Agreement 5.2.1.*

The component weights are replaced by the expectation (6.5) determined by the statistic κ_t*.*

In the fixed advisory system, measured data do not enter the condition of the distribution on parameters, i.e. $f(\Theta|d(t)) \equiv f(\Theta|d(0)) \equiv f(\Theta)$*. Thus, the one-step-ahead predictor becomes*

$$f(d_{t+1}|d(t)) \quad \equiv \quad \sum_{c \in c^*} \hat{\alpha}_{c;0} \prod_{i \in i^*} f(d_{ic;t+1}|\psi_{ic;t+1}, c). \tag{6.10}$$

The fixed predictor *of a factor output* $d_{ic;t+1}$

$$f(d_{ic;t+1}|\psi_{ic;t+1}, c) \equiv \int f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c) f(\Theta_{ic})\, d\Theta_{ic} \tag{6.11}$$

eliminates the unknown parameters of the parameterized factor by its integration weighted by the prior pdf $f(\Theta_{ic})$*, by the prior estimate of factor, Agreement 5.2.1 .*

The component weights are replaced by the expectation (6.5) determined by the statistic κ_0*.*

6.1.2 Likelihood on variants

During the learning phase of the p-system construction, the predictors (6.8) or (6.10) serve for selecting the best variant among those differing in initialization, forgetting, structure etc. Let $d(\mathring{t})$ be learning data and $v_k,\ k \in k^* = \{1, \ldots, \mathring{k}\}$ label a finite collection of such, a priori equally probable, variants. Then, the Bayes rule provides their posterior probabilities

$$f(v_k|d(\mathring{t})) \propto f(d(\mathring{t})|v_k).$$

The variants $v_{\bar{k}}$ with high values of the predictive pdf $f(d(\mathring{t})|v_k)$ evaluated at measured data $d(\mathring{t})$ are obviously preferable. For conciseness, we call the repeatedly evaluated values $f(d(\mathring{t})|v)$ *v-likelihoods*.

For the adaptive advisory system, the chain rule for pdfs implies that the v-likelihood is obtained as the product of one-step-ahead predictors with d_{t+1} equal to the measured data item. This is not true for the fixed advisory system as

$$f(d(\mathring{t})|v) = \int \prod_{t \in t^*} f(d_t|d(t-1), \Theta, v)\, f(\Theta|v)\, d\Theta \neq \prod_{t \in t^*} \int f(d_t|d(t-1), \Theta, v)\, f(\Theta|v)\, d\Theta.$$

The product of fixed one-step-ahead predictors is just an approximation of the v-likelihood corresponding to the fixed advisory system. This approximation is reasonable if the fixed prior estimate $f(\Theta|v)$ of the mixture parameters, Agreement 5.2.1, has almost one-point support. Otherwise, the adopted approximation may be misleading.

We measure distance of v-likelihood values while searching for the best variant. To this purpose, we can use a data-space sampled version of the KL distance that is traditionally called *entropy rate.*

Agreement 6.1.2 (Entropy rate of a pair of mixtures) *Let* v*,* $\tilde{v}$ *be a pair of variants corresponding to a pair of parameterized mixtures and the prior mixture estimates, Agreement 5.2.1, i.e. prior pdfs* $f(\Theta_v|v), f(\Theta_{\tilde{v}}|\tilde{v})$*, have the form (6.3), on their respective parameters* $\Theta_v, \Theta_{\tilde{v}}$*. Let* $d(\mathring{t})$ *be measured data at disposal.*

The entropy rate of factors used in components c_v, $c_{\tilde{v}}$ with a common factor output $d_i = d_{ic_v} = d_{ic_{\tilde{v}}}$ is, for the fixed predictors,

$$\begin{aligned}
\mathcal{D}_{ic_v c_{\tilde{v}}} &\equiv \frac{1}{\mathring{t}} \sum_{t \in t^*} f(d_{i;t}|\psi_{ic_v;t}, c_v, v) \ln\left(\frac{f(d_{i;t}|\psi_{ic_v;t}, c_v, v)}{f(d_{i;t}|\psi_{ic_{\tilde{v}};t}, c_{\tilde{v}}, \tilde{v})}\right), \quad \textit{with} \\
f(d_i|\psi_{ic_v}, c_v, v) &\equiv \int f(d_i|\psi_{ic_v}, \Theta_{ic_v v}, c_v, v) f(\Theta_{ic_v v}|v)\, d\Theta_{ic_v v} \qquad (6.12) \\
f(d_i|\psi_{ic_{\tilde{v}}}, c_{\tilde{v}}, \tilde{v}) &\equiv \int f(d_i|\psi_{ic_{\tilde{v}}}, \Theta_{ic_{\tilde{v}}\tilde{v}}, c_{\tilde{v}}, \tilde{v}) f(\Theta_{ic_{\tilde{v}}\tilde{v}}|\tilde{v})\, d\Theta_{ic_{\tilde{v}}\tilde{v}}.
\end{aligned}$$

The entropy rate of the components c_v, $c_{\tilde{v}}$ is defined

$$\mathcal{D}_{c_v c_{\tilde{v}}} \equiv \sum_{i \in i^*} \mathcal{D}_{ic_v c_{\tilde{v}}}, \; i^* \equiv \{1, \ldots, \mathring{d}\}. \qquad (6.13)$$

The same quantities are defined for the adaptive predictors by using the posterior pdf on parameters instead of the prior one.

Remark(s) 6.1.1

The entropy rate of factors measures how far is the judged predictor from another predictor. The entropy rate of components measures their pair-wise similarity. They can be used, for instance, in cancelling of superfluous or merging similar components.

Branch and bound algorithm, Section 6.1.3, searches for the highest v-likelihood among a finite set of variants. They are, however, evaluated approximately only. Thus, the maximum value can be blurred by the "approximation noise". Thus, a tool is needed distinguishing whether compared v-likelihood functions are practically equal or not. The following simple test of hypothesis serves to this purpose.

Proposition 6.1.2 (Test on equality of log-likelihoods) *Let us consider a pair of random sequences $l_1(\mathring{t})$, $l_2(\mathring{t})$ of real scalars $l_{1;t}, l_{2;t}$. Let us formulate the following hypotheses*

$$\begin{aligned}
H_0 &: l(\mathring{t}) \equiv \big(l_1(\mathring{t}) - l_2(\mathring{t})\big) \sim \mathcal{N}(0, rI_{\mathring{t}}), \quad \textit{with unknown variance } r > 0, \; \textit{and} \qquad (6.14) \\
H_1 &: l(\mathring{t}) \equiv \big(l_1(\mathring{t}) - l_2(\mathring{t})\big) \sim \mathcal{N}(m\mathbf{1}_{\mathring{t}}, rI_{\mathring{t}}), \quad \textit{with unknown mean } m \in (-\infty, \infty) \;\&\; r > 0, \; \textit{where}
\end{aligned}$$

*$I_{\mathring{t}}$ is $(\mathring{t}, \mathring{t})$-*unit matrix *and $\mathbf{1}_{\mathring{t}}$ is $\mathring{t}$-vector consisting of units.*

Let both hypothesis are a priori equally probable and conjugate prior Gauss-inverse-Wishart pdfs $GiW_r(\varepsilon, \varepsilon)$, $GiW_{m,r}(\varepsilon I_2, \varepsilon)$, $\varepsilon > 0$, see Chapter 8, are used for respective hypotheses.

Then, for $\varepsilon \to 0$,

$$f(H_1|l(\mathring{t})) = \left[1 + \left(1 - \frac{\bar{l}^2}{\overline{l^2}}\right)^{0.5\mathring{t}}\right]^{-1}, \quad \bar{l} \equiv \frac{1}{\mathring{t}} \sum_{t \in t^*} l_t, \quad \overline{l^2} \equiv \frac{1}{\mathring{t}} \sum_{t \in t^*} l_t^2. \qquad (6.15)$$

Thus, for a given probability $\beta \in (0,1), \beta \approx 1$, we take both variants as equivalent iff

$$1 - \left(\frac{\beta}{1-\beta}\right)^{2/\mathring{t}} \geq \frac{\bar{l}^2}{\overline{l^2}}. \qquad (6.16)$$

Proof: This is a special case of Bayesian prediction and hypothesis testing related to normal pdfs that are discussed in detail in Chapter 8. The general sufficient statistics are just given a specific form. □

Problem 6.1.1 (Alternative tests of equality) *Experiments indicate excessive sharpness of the test. Other ways have to inspected. For instance, assuming asymptotic normality of the sequence $t^{-0.5}\sum_{\tau=1}^{t} l_\tau$, the test on the zero mean of the limiting distribution applied. Alternatively, expected value in H_1 may be assumed as $\mathring{t}$-vector m.*

6.1.3 Branch and bound techniques

In subsequent sections, we search for the highest v-likelihood in a relatively complex setting. The search itself can be seen as a version of *branch and bound techniques*. This methodology is widely used in optimization [50]. It generates sets of alternatives, evaluates values of the optimized functional (function) and then it bounds the set of the inspected alternatives.

More formally, let maximum of the functional (function) $F : X^* \rightarrow (-\infty, \infty)$ over X^* be searched for. Then, the considered generic algorithm looks as follows.

Algorithm 6.1.1 (Generic branch-and-bound algorithm)
Initial mode

- *Select upper bound* $\mathring{n}$ *on the number* n *of iterations.*
- *Set the iteration counter* $n = 1$ *and the initial guess of the maximum value* $\bar{F} = -\infty$.
- *Select the initial set of alternatives* $X_n^* \equiv \{X_{in} \in X^*,\ i = 1, \ldots, \mathring{i}_n\},\ \ \mathring{i}_n < \infty$.
- *Select optional parameters of branching and bounding mappings.*

Iterative mode

1. *Evaluate the values* $F_n^* \equiv \{F(X_{in}),\ X_{in} \in X_n^*\}$ *of* F *on the set of alternatives* X_n^*.
2. *Find* $\bar{F}_n \equiv \max F_n^*$.
3. *Take a maximizing argument* $\bar{X}_n$ *of* F *on* X_n^* *as an approximation of the maximizing argument searched for.*
4. *Take* $\bar{X}_{n-1}$ *as the maximizing argument of* F *on* X^* *and stop if* $\bar{F}_n \leq \bar{F}$. *Otherwise set* $\bar{F} = \bar{F}_n$ *and go to the next step.*
5. *Branch the set of alternatives* X_n^* *by a* branching mapping $\mathcal{A}$
$$\mathcal{A} : (X_n^*, F_n^*) \rightarrow X_{n+1|n}^* \equiv \{X_{i(n+1|n)} \in X^*,\ i = 1, \ldots, \mathring{i}_{n+1|n}\},\ \mathring{i}_{n+1|n} \geq \mathring{i}_n. \tag{6.17}$$
6. *Evaluate the values of* F *on the set* $X_{n+1|n}^*$ *of alternatives*
$$F_{n+1|n}^* \equiv \left\{F(X_{i(n+1|n)}),\ X_{i(n+1|n)} \in X_{n+1|n}^*\right\}.$$
7. *Bound the set of alternatives* $X_{n+1|n}^*$ *by a* bounding mapping $\mathcal{U}$
$$\mathcal{U} : (X_{n+1|n}^*, F_{n+1|n}^*) \rightarrow X_{n+1}^* \equiv \{X_{i(n+1)} \in X^*,\ i = 1, \ldots, \mathring{i}_{n+1}\},\ \mathring{i}_{n+1} \leq \mathring{i}_{n+1|n}. \tag{6.18}$$
8. *Increase the counter* $n = n + 1$. *Take* $\bar{X}_n$ *as the maximizing argument of* F *and stop if* $n > \mathring{n}$. *Otherwise go to the beginning of* Iterative mode.

Proposition 6.1.3 (Properties of branch and bound algorithm) *Let us select* $\bar{X}_n \in \text{Arg}\max F_n^*$ *and* $\bar{X}_{n+1|n} \in \text{Arg}\max F_{n+1|n}^*$. *Let* $\bar{X}_n$ *stay in* $X_{n+1|n}^*$ *and* $\bar{X}_{n+1|n}$ *in* X_{n+1}^*.

Then Algorithm 6.1.1 may stop at local maximum only in the step 4. The argument found during the premature break at the step 8 cannot be worse then the initial guess.

Proof: The statement is implied directly by construction of Algorithm 6.1.1 and by the assumption that the best arguments is either preserved or even improved during the each complete step of the algorithm. □

Remark(s) 6.1.2

1. *Algorithm may not stop at the step 4.*

2. *Attaining of the global maximum is guaranteed when the maximized F has a single mode and the algorithm stops at the step 4 without the enforced break in the step 8.*

3. *The specific algorithm is obtained by selecting*

   - *the initial set of alternatives X_1^*,*
   - *the branching $\mathcal{A}$ and bounding $\mathcal{U}$ mappings.*

   Efficiency of this algorithm depends on the adapted choice of these tuning knobs. Section 6.4 specifies the functional F we maximize and justifies options X_1^*, $\mathcal{A}, \mathcal{U}$ we consider as promising.

4. *Selection of the bounding mapping $\mathcal{U}$ is straightforward if we want to respect conditions of Proposition 6.1.3. Entries $X_{n+1|n}^*$ with low values of F are simply omitted. Thus, the choice of the number of the preserved arguments $\mathring{i}_{n+1}$ is the only real option made in connection with bounding mapping $\mathcal{U}$. In our application, this number is restricted from above by computational complexity. It is also restricted from below as X_{n+1}^* has to provide sufficient amount points for branching. Thus, the choice of the initial set X_1^* and of the problem-adapted branching mapping $\mathcal{A}$ decide on the achieved efficiency.*

Problem 6.1.2 (Critical review of branching mapping) *Choice of the branching mapping is hardest problem addressed. The options made below could be and should be complemented by "stealing" the most promising ideas occurring in various fields of optimization, for instance, in genetic algorithms [51]. The energy spent on a detailed review should pay back.*

6.2 Data preprocessing

The advisory system relies on its ability to model significant relationships observable on the o-system, Agreement 5.1.1. An attempt to reflect all relationships is usually hopeless as it leads to a hardly manageable model. For this reason, it is necessary to separate rare events and suppress superfluous details from modelling. Data have to be preprocessed. The preprocessing tasks can be classified as follows.

Data transformation that covers:
data scaling that suppresses numerical problems and simplifies the choice of optional learning and design parameters, for instance, it allows us to standardize prior pdfs;
reduction of dimensionality that decreases computational and numerical load by exploiting measurement redundancy;
tailoring of time-scale that harmonizes the modelling and operator-actions rates.

Outlier removal excludes rarely occurring data that differ significantly from the behavior of the rest of data. Mixture estimation separates frequently occurring outliers automatically as it assigns them specific components. The outlier preprocessing can often be decomposed into:
outlier detection when the outlying data items are marked as missing ones;
data interpolation when missing data are substituted by their estimates.

Outlier detection is often based on use of non-linear median or linear moving average filters [52, 53, 54]. The local smoothing and local approximation based on heavy-tailed models are studied. A more sophisticated class of methods, with dynamically changing boundaries for outliers, is described in [55] and [56]. Modelling of the residual sequence by a heavy tailed α-stable distribution makes essence of this approach.

Filtering removes parts of data unrelated to the underlying dynamic relationships of quantities of interest. The removed parts reflect usually deficiencies of the measurement process. The noise suppression is be done by a variety of signal processing techniques [57]. For instance, differences in frequency contents are exploited.

The final aim of the data processing has to be kept in mind while applying it. Any data preprocessing adds a dynamic module into the closed loop to be handled by the advisory system. This simple statement implies that some classical signal-processing techniques, that add a substantial dynamic delay, have to be avoided. The mixture estimation brings also the following additional problem.

Preprocessing operations applied to individual signals must not mix data coming from various components!

Techniques solving of the discussed tasks partially overlap. The following table guides in this respect.

Tasks	**Methods**	**Section**
Data transformation		6.2.1
Data scaling Reduction of dimensionality Tailoring of time scale	Physical range and sample moments Principal component analysis Local filtering	
Outlier removal		6.2.2
Detection Interpolation	Check of physical boundaries Model-based tests Mixtures of normal and outlying data Model-based interpolation Filter-based interpolation Wavelet-based filtering Local filtering	
Filtering		6.2.3
Removal of well-separable noise Removal of poorly separable noise & grouping Noise removal by non-causal filtering	Classical smoothing Local filtering Wavelet-based filtering	

6.2.1 Data transformation

Data scaling

For numerical reasons, it is necessary to scale the processed data to similar numerical ranges. Data scaling and shifting should be done on the basis of usual ranges of considered variables. The wide-spread scaling based on sample moments may be sensitive to outlying data. Use of extreme values for scaling is even less robust.

It is reasonable to deal with scaled data throughout learning and design of advisory system. Rescaling to the user units is needed just in presentation of the design results to the operator during advising, Section 9.1.4.

Dimensionality reduction

Principal component analysis (PCA) is one of the most important techniques for dimensionality reduction of strongly correlated multi-dimensional data [58]. The standard PCA assumes a single multi-variate (normal) distribution. The data clustering techniques assume that data are generated by a mixture of (normal) distributions. So, a direct use of PCA is slightly illogical. It would be better to fit a mixture in the original data space and then apply PCA to individual normal components. We need, however, to reduce the dimensionality before clustering in order to improve it. A sort of data pre-segmenting provides a compromise between these contradictory needs. It is possible in the learning phase as the processed data can often be segmented into good and bad or into other classes. PCA then can be applied separately to each of them.

Alternatively, mixtures with components having low-rank expectations could be and should be considered. The idea [59, 60] to solve Bayesian estimation as minimization of a suitable form of the KL distance seems to be proper way for doing this task.

Problem 6.2.1 (PCA on components) *Feasibility and real influence of the proposed ways of PCA application on components should be studied in a quantitative way.*

Tailoring of time scale

The discrete-time model are built for the sampling rate given by the highest rate feasible by the human being. Data are mostly collected with higher rates. Thus, a representative has to be searched for a group of measured data. This is can be done with help of a *local filter*. Its use allows us, moreover, to suppress both high-frequency noise and partially also outliers. The skipping of data, often recommended, falls into this category but it is obvious that it looses precious information.

6.2.2 Outlier removal

Outlier removal consists of interconnected detection of outliers and interpolation of data they hide.

Outlier detection based on physical boundaries

Mostly, the inspected data d_t have a specific physical meaning with well defined boundaries $\underline{d}$, $\overline{d}$ on their usual values, $\underline{d} \leq d_t \leq \overline{d}$.

The censored data that do not fall in this hyper-cube can be immediately interpolated or marked as missing for a postponed interpolation. Often, the ranges on signal-rate changes are also available. Then, a similar hyper-cube on temporal changes of data is defined and exploited in the same way.

Model-based detection

Comparison of the current signal value with a value predicted by a suitable models is the efficient method for detection of an unexpected value. The detection discussed in previous paragraph is essentially of this type with physically-justified expected characteristics of the inspected signal.

Comparison of the output of a fixed filter with the value in question rests on the assumption that the normal signal passes the filter without distortion, on assumption that the filter models its course properly. Of course, the model can be built explicitly and even recursively estimated.

The simplest, widely used technique checks the inequality $\sigma_{i;t} h \leq |d_{i;t} - \mu_{i;t}|$. In it, $\mu_{i;t}$ and $\sigma_{i;t}$ are the sample mean or median and the standard deviation of measurements $d_{i;t-\mathring{k}}, \ldots, d_{i;t+\mathring{k}}$, respectively. The size of the window is $2\mathring{k}+1$ and h is an optional threshold. Outlier is detected if the inequality is satisfied. This is the non-causal version of the detector. The median and standard deviation are calculated from $d_{i;t-\mathring{k}}, \ldots, d_{i;t-1}$ in its causal counterpart.

The two plots in Figure 6.1 show the outlier removal for artificial three-dimensional data. The discussed outlier removal technique was used. The data $d_{i;t}$ marked as outliers are interpolated by $\mu_{i;t}$.

Mixtures of normal and outlying data

Rare outliers can be processed at level of individual signal. Frequently occurring outliers feeded into mixture estimation create naturally extra component(s). The overall mixture estimation finds the component to which the current signal value belongs. Thus, we can assume that the signal comes from a single “global” component and model it by a local simple, typically two-component mixture. This model is recursively estimated using quasi-Bayes estimation with forgetting, Section 6.5. Again, the estimation detects the component to which the inspected value belongs. It detects essentially whether the value is outlier or not and uses it for correction of the corresponding parameter estimates. Data interpolation is then straightforward. Outlier is replaced by the value predicted by the component corresponding to ”normal” course of the treated signal. Thus, the model-based interpolation is used.

Success of the specific application of this idea depends heavily on a good decision whether the signal belongs to a single global component. A careful selection of forgetting factors used estimation of the local mixture is even more important. The component describing “normal” course has to be highly adapted in order to use effectively local model that modifies the normal signal as little as possible. The component describing outliers should be adapted very little otherwise there is danger that the separation of both modes will be lost.

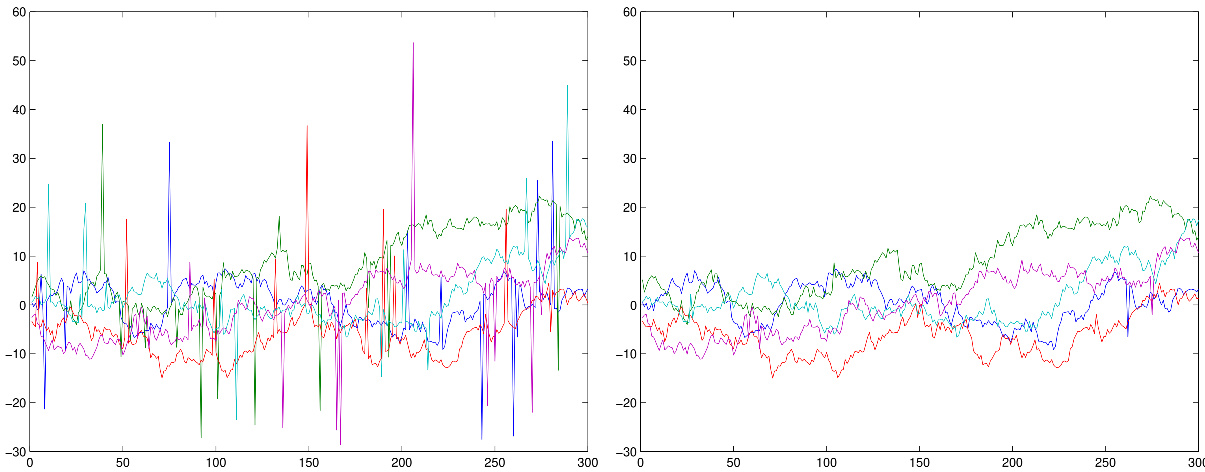


Figure 6.1: Outlier removal based on checking of deviations from sample mean computed on a moving window. Shown are raw and processed data.

Model-based interpolation

This method replaces a missing data item by the point prediction made by a simple model fitted to the outlier-free data. The replacement of the invalid data item by the last measured value is its practically attractive variant. It adds small filtering dynamics. The assumption that the basic signal varies slowly represents the model behind this variant.

Generally, model-based methods are strongly application-domain dependent. For example, the paper [61] describes interesting, model-based technique for data interpolation of degraded audio signal. Another method of the model-based interpolation as well as dynamic-boundaries test are described in [55].

Filter-based interpolation

This type of interpolation generates approximation of actual data item by classical smoothing filters.

The three plots in Figure 6.2 illustrate the interpolation of missing data. The artificial data with outliers are processed. The raw data are in the first plot. The data out of the given physical boundaries are marked as missing. The lower bound $\underline{d} = -10$ is considered only. In the second plot, the data marked as missing are set to be equal to the value of this boundary. At the third plot, the data are interpolated by a filter-based smoothing.

6.2.3 Filtering

High-frequency measurement noise is a typical disturbing element that should be removed. There is a lot of techniques available to this purpose. They are excellent if the frequency content of the removed noise differs substantially from that of the recovered signal. Otherwise, they add, often substantial, dynamics that is harmful in the closed loop.

Local filters overcome this problem. They are applicable whenever grouping of data is desirable. In off line mode even non-causal filtering is acceptable. Then, the wavelet filters are powerful tool for removing selected, typically, high frequencies.

Classical filters for smoothing

The classical smoothing filters remove the high frequency noise. The mean and median filters are justified whenever we can assume that the system behaves for a long time period according to a single component. Forgetting or moving-window versions can be employed. The mean filter calculated on a moving window is defined $x_{i;t} = \sum_{k=1}^{\mathring{k}} d_{i;t-k+1}$, where $x_{i;t}$ is the filter output, and $\mathring{k}$ is the window size. The *median filter* on the moving window is defined $x_{i;t} = \mathsf{median}(d_{i;t}, d_{i;t-1}, \ldots, d_{i;t-\mathring{k}+1})$, where the function

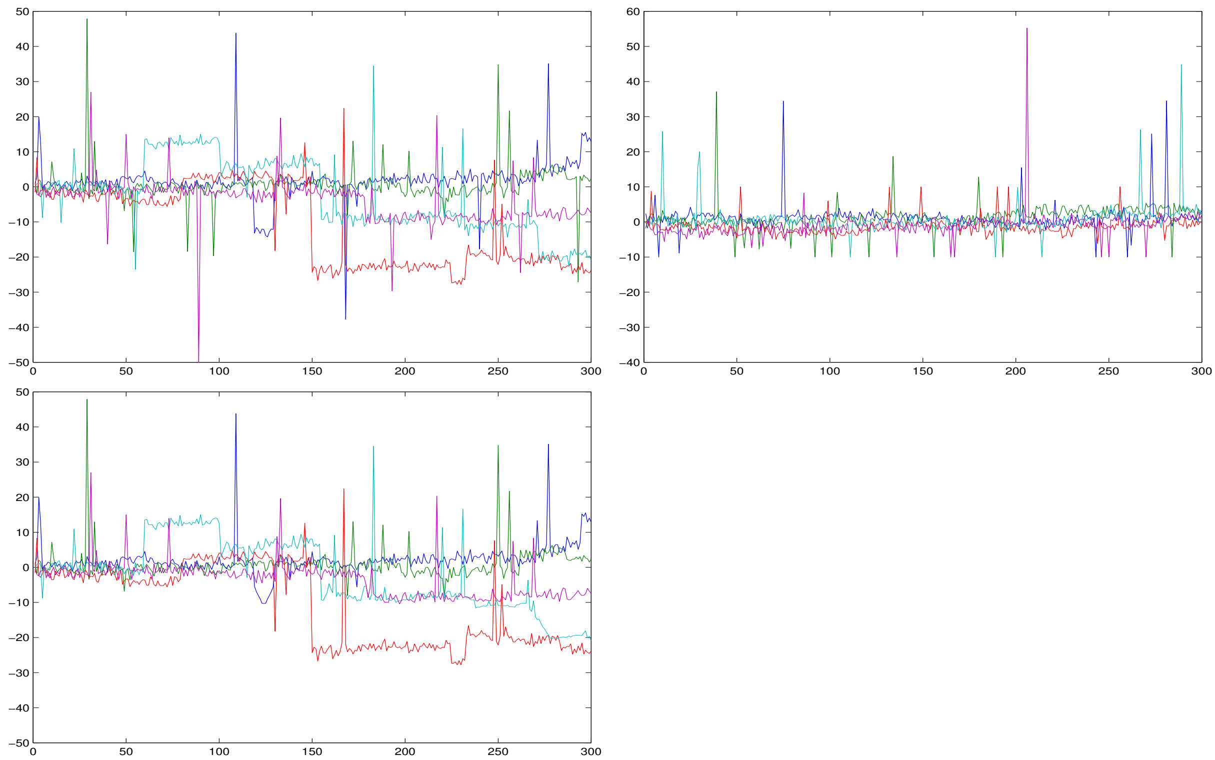


Figure 6.2: Interpolation of missing data. Shown are raw data, data cut at given lower bound and smoothed data.

$\mathsf{median}(\cdot)$ evaluates median of the argument. The *mean filter* with forgetting is generated recursively $x_{i;t} = \lambda_i x_{i;t-1} + (1-\lambda_i) d_{i;t}$ and it is determined by the forgetting factor $\lambda_i \in (0,1)$. The initial filtered value is set $x_{i;1} = d_{i;1}$. The filtered value is the weighted mean of all previous values of $d_{i;t-k+1}$

$$x_{i;t} = \sum_{k=1}^{t-1} \rho_{ik;t} d_{i;t-k+1}.$$

The weights are $\rho_{ik;t} = (1-\lambda_i)\lambda_i^{k-1}$, except for $\rho_{i1;t} = \lambda_i^{t-1}$. Notice that $\sum_{k=1}^{t} \rho_{ik;t} = 1$.

Analogy to the mean filter with forgetting leads to the median filter with forgetting defined by $x_{i;t} = \mathsf{wmedian}(d_{i;t}, d_{i;t-1}, \ldots, d_{i;t-\mathring{k}+1}; \rho_{i1;t}, \ldots, \rho_{i\mathring{k};t})$, where $\mathsf{wmedian}$ denotes the median of data weighted by $\rho_{i1;t}, \ldots, \rho_{i\mathring{k};t}$, [56].

Local filters

Local filters are based on the assumption that the underlying signal does not change too quickly and on the possibility to group several data. Then, the signal can be modelled by a simple function, say by a straight line, within the time-span determined by a single period of operator actions. The redundant data are used for estimating this function whose single value is offered as the grouped and filtered value. Models of the corrupting noise (light tailed Gaussian, mixture of a pair of Gaussian pdfs differing in noise variance, heavy tailed Cauchy, one-sided exponential etc.) feeded into the Bayesian estimation and prediction provide directly the required estimate. The piece-wise application of the filter guarantees that a small dynamic delay is introduced. It is at most single period at the operator rate. Irregularities in the original sampling can be suppressed by choosing evenly distributed time moments at which the filtered samples are evaluated.

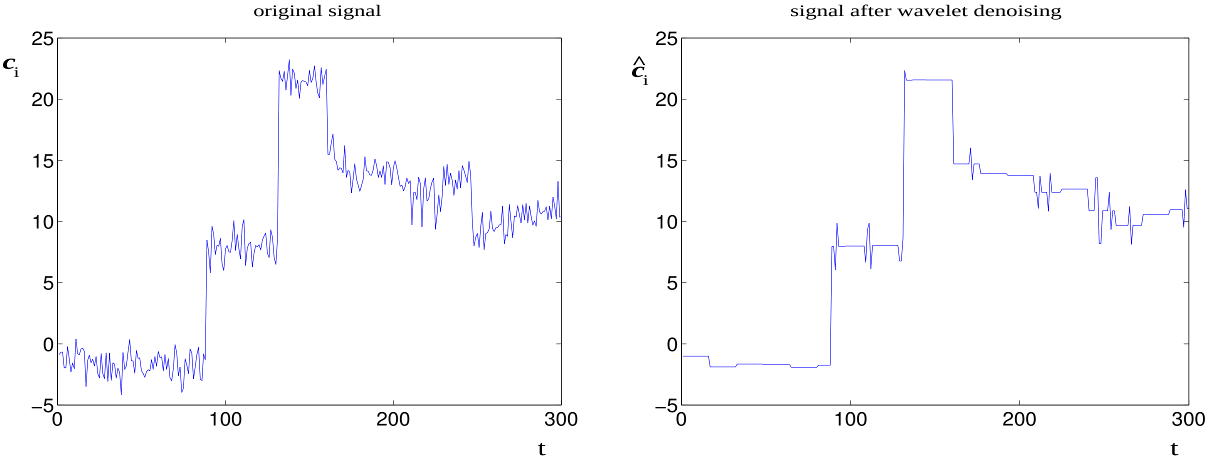


Figure 6.3: Wavelet de-noising of a piece-wise smooth signal. Raw and de-noised data are shown.

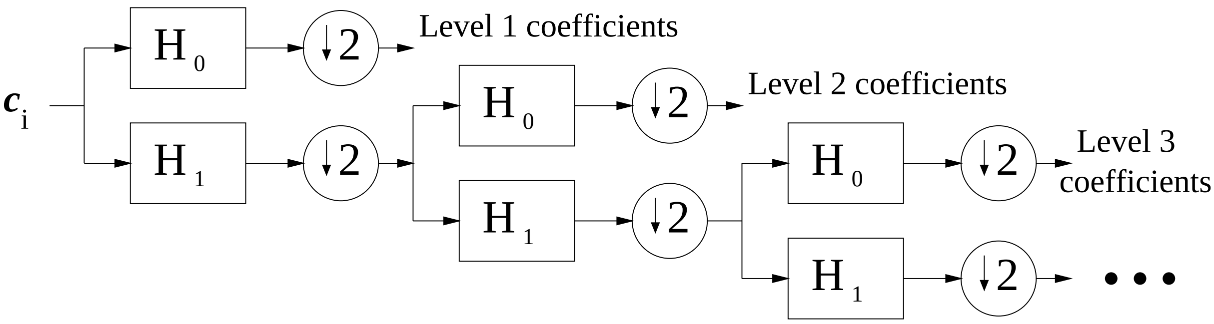


Figure 6.4: Multilevel Haar-wavelet decomposition of the signal by the filter bank. $\downarrow 2$ denotes down-sampling and H_0, H_1 mark filter elements.

Problem 6.2.2 (Outlier cancelling filters) *Local regression with heavy-tailed noise seems to be proper direction inspected. Obsolete filters of this type [62] should be renewed using advanced techniques for on line non-linear filtering, see Section 3.4.*

Wavelet filtering

Wavelet transformation converts original data into different resolution levels with respect to their spectrum. By manipulating the gained spectra, high frequencies can be suppressed without removing "edges" from the original signal. This non-causal filter can be used in the learning phase only.

The Figure 6.3 shows an example, of wavelet de-noising applied to artificial data containing significant edges (steps). Data are processed for each channel $d_i(\mathring{t})$ separately. The figure shows a single channel processed. The Haar wavelet is employed. The original signal is analyzed into wavelet coefficients, Figure 6.4, by the discrete wavelet transformation, [63]. H_0 denotes the low pass filter operator and H_1 the high pass filter operator and the symbol $\downarrow 2$ denotes down sampling at the rate of $2:1$. Thereafter, the smallest coefficients are suppressed – using an appropriate threshold – and the signal is reconstructed by a synthesis process, which is the inverse to that shown in Figure 6.4.

Problem 6.2.3 (Comparison of quality of filters) *Mostly, preprocessing reduces image of reality. It makes conceptually hard to compare quality of competitive filters. Models obtained on differently filtered data surely cannot be compared on them.*

Probably, the performance on predictions of raw data conditioned on filtered ones should be compared. It is reasonable when preprocessing is "mild", but it is not clear whether such a comparison is fair, for instance, when removing outliers. Obviously, a solution of this problem is of a direct practical interest.

Remark(s) 6.2.1

Recently a novel filtering technique based on a mixture whose components differ just in filtering used has been proposed [64]. It seems to be an efficient and promising way how to merge a bank of filters while decreasing sensitivity to the choice of filter parameters.

6.2.4 Filters generating factors in the exponential family

The dynamic exponential family, Section 3.2, Agreement 3.2.1, that provides the dominant supply of factors, contains essentially normal regression models linear in parameters and Markov chains. Its richness stems mainly from allowed transformations of data vectors, i.e. from filtering. This aspect is discussed here in order to prepare conceptual framework for implementation of such variants.

The i-th parameterized factor, predicting the factor output $d_{i;t}$ and belonging to the dynamic exponential family is described by p(d)f

$$f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1), \Theta) \equiv f(d_{i;t}|\psi_{i;t}, \Theta) = A(\Theta)\exp\left[\langle B(\Psi_{i;t}), C(\Theta)\rangle + D(\Psi_{i;t})\right].$$

Two ingredients determine it.

1. The filter $F(\cdot)$ updating recursively the data vector $\Psi'_{i;t} \equiv [\tilde{d}_{i;t}, \psi'_{i;t}] \equiv$ [*filtered factor output*,filtered regression vector]
$$[\Psi_{i;t}, \Omega_t] = F\left(d_{i\_\mathring{d};t}, \Psi_{i;t-1}, \Omega_{t-1}\right). \tag{6.19}$$
The symbol Ω_t denotes a finite-dimensional state of the filter called shortly *filter state.*

2. The elements in the functional form of the exponential family,

$$\begin{aligned} \langle\cdot,\cdot\rangle &\equiv \text{ a functional, linear in the first argument,} \\ A(\Theta) &\equiv \text{ a non-negative scalar function defined on } \Theta^*, \\ B(\Psi),\, C(\Theta) &\equiv \text{ compatible array functions defined on } \Psi^*,\ \Theta^*, \text{ respectively, and} \\ D(\Psi) &\equiv \text{ scalar function on } \Psi^*. \end{aligned} \tag{6.20}$$

The intended use of the filtered data vectors in prediction and design of actions imposes restrictions on the admissible filters.

Requirement 6.2.1 (Allowed filters) *The filter (6.19)*

1. *determines the filtered regression vector ψ_t independently of the value $d_{i;t}$;*
2. *allows to reconstruct uniquely the original $d_{i;t}$ when $\tilde{d}_{i;t}$ and $\psi_{i;t}$ are given;*
3. *allows to reconstruct uniquely the original actions among $d_{i+1;t}, \ldots, d_{\mathring{d};t}$ when $\psi_{i;t}$ is given.*

A limited supply of used the filters (6.19) and elements (6.20) allows us to label them by the *data vector type* $\equiv {}^{\dagger}\Psi \in {}^{\dagger}\Psi^* \equiv \{1, \ldots, {}^{\dagger}\mathring{\Psi}\}$ and by the *functional form type* $\equiv {}^{\dagger}F \in {}^{\dagger}F^* \equiv \{1, \ldots, {}^{\dagger}\mathring{F}\}$.

At present, the following options are either finished or partially elaborated.

Filtering of data vectors

- State ϕ_t in the phase form, see Agreement 5.2.1.
  The time-invariant filter state Ω is the list determining the structure of the data vector with the state in the phase form, see Agreement 5.2.1.
- Scale state ϕ_t in the phase form, see Agreement 5.2.1.
  Scaling factor, additive term and structure of the data vectors form the time-invariant filter state Ω.
- Log-normal factors, Section 8.1.6, with state in the phase form, see Agreement 5.2.1.

- ARMAX factors with a filter generated by a known fixed finite correlations specifying the noise part of the model [65].

  They are equivalent to a state model leading to a special Kalman filtering [66] that can be combined with the parameter estimation without an approximation.

  Filter state Ω_t contains the time-invariant description of the noise correlations, i.e. both structure and parameters. It also contains the time-varying filter resulting from it, see [66]. In terms of Kalman filter, they are equivalent to the time invariant noise description and time varying Kalman gain.

- Mixture of ARMAX factors of the above type with a common ARX part, [64].

  Filter state Ω_t contains several different $\Omega_{j;t}$ each of the type described in the previous item.

- Mixture of ARX factors with the common ARX part and components differing in filters applied on the original state in the phase form [67].

  Filter state Ω_t contains information on the structure and time-invariant parameters of the underlying filters.

Functional forms within the exponential family

- Normal factors, Chapter 8.
- Markov chain factors, Chapter 10.
- MT normal factors, Chapter 11.
- *Static factors* in exponential family.

It can be expected that both these lists will be slowly and gradually extended. For instance, specific incremental versions of the state in phase form may arise. Non-linear/non-Gaussian factors embedded into the exponential family in the way described in Section 3.4 serve as another example.

Both the small extent and slow expansion allow us to use just the introduced types.

Remark(s) 6.2.2

The log-normal factors can be taken as a representative of a whole group that is gained from linear normal model by a one-to-one, unknown-parameter-free transformation of the factor output [14]. This simple possibility should be kept in mind as such transformation increase the modelling power surprisingly much.

6.2.5 Statistics for filtering and estimation in exponential family

The Bayesian estimation in the exponential family is described by Proposition 3.2.1. We use conjugate prior pdfs. It has the form

$$f(\Theta) \equiv f(\Theta|d(0)) \propto A(\Theta)^{\nu_0} \exp\left[\langle V_0, C(\Theta)\rangle\right].$$

The estimation is combined with stabilized forgetting, Algorithm 3.1.1. It is specified by the forgetting factor $\lambda \in [0, 1]$ and alternative pdf ${}^{\lfloor A}f(\Theta|d(t))$. We select it as a conjugate pdf, too

$${}^{\lfloor A}f(\Theta|d(t)) \equiv {}^{\lfloor A}f(\Theta) \propto A(\Theta)^{\,{}^{\lfloor A}\nu_t} \exp\left[\left\langle {}^{\lfloor A}V_t, C(\Theta)\right\rangle\right].$$

In our context, it is mostly chosen as a time invariant pdf. So that, without substantial restriction of generality, we can assume that the formal structure of its updating coincides with that of the factor statistics in question.

With the options made, the posterior pdf has then conjugate form and its statistics ν_t, V_t can be updated recursively. Approximate estimations considered, Section 6.5, modify these recursions, giving a weight $w_t \in [0,1]$ to increments of these statistics. The compound recursions, describing also evolution of the data vector, have the following form. The first part reflects data updating, cf. Proposition 2.5.1,

$$\begin{aligned}
[\Psi_t, \Omega_t] &= F(d_t, \Psi_{t-1}, \Omega_{t-1}),\ \Psi_0,\ \Omega_0 \text{ must be given,} \qquad (6.21)\\
\tilde{V}_t &= V_{t-1} + w_t B(\Psi_t),\ \ \tilde{\nu}_t = \nu_{t-1} + w_t,\ V_0,\ \nu_0 \text{ are chosen as description of prior pdf,}\\
\left[{}^{\lfloor A}\Psi_t,\ {}^{\lfloor A}\Omega_t\right] &= {}^{\lfloor A}F(d_t,\ {}^{\lfloor A}\Psi_{t-1},\ {}^{\lfloor A}\Omega_{t-1}),\ {}^{\lfloor A}\Psi_0,\ {}^{\lfloor A}\Omega_0 \text{ must be given,}\\
{}^{\lfloor A}V_t &= {}^{\lfloor A}V_{t-1} + {}^{\lfloor A}w_t\,{}^{\lfloor A}B\left({}^{\lfloor A}\Psi_t\right),\ \ {}^{\lfloor A}\nu_t = {}^{\lfloor A}\nu_{t-1} + {}^{\lfloor A}w_t,\ {}^{\lfloor A}V_0,\ {}^{\lfloor A}\nu_0\\
&\quad \text{are chosen as a prior alternative.}
\end{aligned}$$

The *time updating* closes recursion by the stabilized forgetting formula, Algorithm 3.1.1,

$$V_t = \lambda\tilde{V}_t + (1-\lambda)\,{}^{\lfloor A}V_t,\ \ \nu_t = \lambda\tilde{\nu}_t + (1-\lambda)\,{}^{\lfloor A}\nu_t. \qquad (6.22)$$

The externally supplied weights w_t, ${}^{\lfloor A}w_t$, the functions $B(\cdot)$, ${}^{\lfloor A}B(\cdot)$ and updating of data vectors Ψ_t, ${}^{\lfloor A}\Psi_t$ may differ. But both values $B(\cdot)$, ${}^{\lfloor A}B(\cdot)$ have to be compatible with the commonly considered array function $C(\Theta)$. This can be checked when preparing the time updating.

Thus, the complete description of respective factors can be composed of the above elements. Let us make it formal.

Agreement 6.2.1 (Complete description of factors in the exponential family) *The factor in the exponential family is described by*

$$\left({}^{\dagger}\Psi,\ {}^{\dagger}F, \Omega_\tau, \Psi_\tau, V_\tau, \nu_\tau\right) \equiv (\textit{filter type, functional type, filter state,}V\textit{-statistics,degrees of freedom}).$$

The initial values, $\tau = 0$, are stored for application of the iterative learning, Section 6.4.1, in addition to the current values with $\tau = t$.

The filter has to meet Requirement 6.2.1.

6.2.6 Prediction in exponential family with statistics gained by filtering

The term $D(\Psi)$ occurring in the definition of exponential family does not influence estimation. Also, the estimation need not refer to the original data and can use completely the filtered versions of data vectors. For prediction, however, both these aspects are significant.

Proposition 3.2.1 implies that the prediction of the i-th filtered factor output has the form (3.7)

$$\begin{aligned}
f(\tilde{d}_{i;t}|d_{i+1_\mathring{d};t}, d(t-1)) &= \frac{\mathcal{I}(V_{i;t-1} + B(\Psi_{i;t}), \nu_{i;t-1}+1)}{\mathcal{I}(V_{i;t-1}, \nu_{i;t-1})}\exp[D(\Psi_{i;t})],\ \text{with} \qquad (6.23)\\
\mathcal{I}(V,\nu) &\equiv \int A(\Theta)^{\nu}\exp\left[\langle V, C(\Theta)\rangle\right] d\Theta.
\end{aligned}$$

Proposition 2.3.5 on transformation of random quantities implies

$$\begin{aligned}
f(d_{i;t}|d_{i+1_\mathring{d};t}, d(t-1)) &= J(\Psi_t)\frac{\mathcal{I}(V_{t-1} + B(\Psi_t), \nu_{t-1}+1)}{\mathcal{I}(V_{t-1}, \nu_{t-1})}\exp[D(\Psi_t)],\ \text{where} \qquad (6.24)\\
J(\Psi_t) &\equiv \frac{\partial F(d_{i_\mathring{d};t}, \Psi_{i;t-1}, \Omega_{t-1})}{\partial d_{i;t}}.
\end{aligned}$$

This prediction is needed for presenting predictions to the operator as well as for computing v-likelihood values, see Section 6.1.2, used for comparing of various estimation variants. These values can be used even for comparison of different filters applied to the same signal. This provides solution of Problem 6.2.3.

6.3 Use of prior knowledge at factor level

The design of the advisory system relies mostly on vast amount of informative data but a strong prior knowledge may improve quality of the resulting model. This opportunity should not be missed and its inspection forms the core of this section.

We deal with a prior knowledge at the factor level: a prior pdf for each factor is considered as an entity on its own. This fits well to the product form of the considered pdfs, see Agreement 6.1.1.

Computational restrictions force us to use predefined functional forms of prior estimates of factors. Mostly, they are taken as conjugate prior pdfs (3.12). In this case, values of statistics determining them serve for expressing the prior knowledge.

Initially, the available knowledge pieces are translated to a common basis of *fictitious data*, i.e. the data observed or potentially observable by the p-system. Fictitious data are split into internally consistent data blocks. Each of them is processed by the Bayes rule with *forgetting*, see Section 3.1. Then, the estimation results are merged. The merging task is non-trivial due to the unknown, possibly not fully consistent, relationships of the processed data blocks. The particular *knowledge sources* are bellow labelled by $K_k,\ k \in k^* \equiv \{1, \ldots, \mathring{k}\}$.

6.3.1 Internally consistent fictitious data blocks

Some knowledge sources provide knowledge pieces directly in the form of data blocks. They include a bit obsolete data, data from identification experiments differing from usual working conditions – like measurement of step responses – and data gained from a realistic simulation. Section 6.3.2 discusses other sources of knowledge that have to be "translated" into data blocks. The common expression as data blocks allows us to deal with all of them in a unified way.

We sort the knowledge sources into groups whose members provide *internally consistent data blocks.*

Agreement 6.3.1 (Internally consistent data blocks) *The data block D_K reflecting the knowledge piece K is called internally consistent iff $f(\Theta|K)$ can be expressed as the result of Bayesian estimation, possibly with appropriately chosen forgetting, see Section 3.1. The estimation inserts data D_K into the considered parameterized model and starts from a flat* pre-prior pdf $\bar{f}(\Theta)$.

The adequate forgetting should always be used in order to counteract under-modelling. This is of an extreme importance for simulators that may provide a huge amount of data and make over-fitting of the posterior pdfs well possible.

Algorithm 6.3.1 (Processing of internally consistent data blocks)
Initial mode

- *Select the structure of the considered factor.*
- *Specify, usually flat, pre-prior pdf $\bar{f}(\Theta)$.*
- *Select the set of possible alternative forgetting factors $\lambda \in \lambda^* \equiv \{\lambda_1, \ldots, \lambda_{\mathring{\lambda}}\}$, $\lambda_i \in (0, 1]$. Typically,*

$$\lambda_k = 1 - 10^{-k},\ k = 1, 2, 3, 4, 5. \tag{6.25}$$

The choice is implied by the knowledge that the effective memory is $(1 - \lambda_k)^{-1}$ and by the wish to cover a wide time span with a few grid points.
Processing mode

1. *Perform Bayesian estimation and prediction starting from $\bar{f}(\Theta)$ and using the data block D_K for all $\lambda \in \lambda^*$, i.e. obtain $f(\Theta|K, \lambda_i) \equiv f(\Theta|D_K, \lambda_i)$ and v-likelihood $f(d(\mathring{t})|K, \lambda_i)$, $i \in \{1, \ldots, \mathring{\lambda}\}$ evaluated on real data $d(\mathring{t})$*
2. *Choose $f(\Theta|K) \equiv f(\Theta|K, \bar{\lambda})$, where $\bar{\lambda}$ is a point estimate of the forgetting factor within the set λ^*. The value $\bar{\lambda}$ is mostly taken as the maximizing argument of the obtained v-likelihood $f(d(\mathring{t})|K, \lambda_i)$.*

Remark(s) 6.3.1

1. *The construction is applicable if the data D_K allow us to evaluate the value of the parameterized model $f(d|\psi,\Theta)$ at least for a single data vector $\Psi = [d, \psi']'$.*

2. *Selection of the best forgetting factor can be completely avoided if we take the pairs (K, λ_i), $i \in i^*$ as different knowledge items and merge them as described in Section 6.3.3. The weighting used there is able to distinguish good and bad values of forgetting.*

6.3.2 Translation of input-output characteristics into data

Often, guessed or designed input-output characteristics of the modelled system are available. Static gain, a point on frequency response, time-constants serve as examples. Majority, if not all of them, have experimental basis. They describe the expected response of the system to a given stimulus. Such knowledge item K gives characteristics of the predictor

$$f(d|\psi) \equiv f(d|\psi, K) \equiv \int f(d|\psi,\Theta) f(\Theta|K)\, d\Theta \tag{6.26}$$

for a scalar d (a factor is dealt with) and a fixed regression vector ψ. Mostly, some moments of the predictive pdf can be guessed. Formally, we assume the knowledge piece in the form

$$h(\psi) = \int H(d,\psi) f(d|\psi)\, dd, \text{ where} \tag{6.27}$$

$h(\psi)$ and $H(\Psi) \equiv H(d,\psi)$ are known vector functions. Typically, we know

$$\hat{d} = \int d f(d|\psi)\, dd, \qquad r_d = \int (d-\hat{d})^2 f(d|\psi)\, dd. \tag{6.28}$$

The case (6.28) corresponds to the knowledge $h(\psi) = [\hat{d}, r_d]$, $H(d,\psi) \equiv H(\Psi) = [d, (d-\hat{d})^2]$.

If no pdf $f(\Theta|K)$ fulfilling (6.27) exists then this information source cannot be internally consistent, see Agreement 6.3.1, and has to be split into internally consistent parts. However, as a rule, the knowledge item (6.26), (6.27) does not determine the constructed pdf $f(\Theta|K)$ completely. It is reasonable to construct such a pdf $f(\Theta|K)$ that expresses just the considered information item. Thus, it makes sense to choose such a $f(\Theta|K)$ that is the nearest one to the flat pre-prior pdf $\bar{f}(\Theta)$. This prior estimate $f(\Theta|K)$ is searched for among all pdfs

$$f(\Theta|K) \in f_K^* \equiv \{\text{pdfs fulfilling (6.26), (6.27)}\}. \tag{6.29}$$

It motivates our option of $f(\Theta|K)$ respecting (6.26) and (6.27) as

$$f(\Theta|K) \in \operatorname{Arg} \min_{f \in f_K^*} \mathcal{D}(f||\bar{f}). \tag{6.30}$$

Proposition 6.3.1 (Knowledge of input-output characteristics) *The minimizing argument $f(\Theta|K)$ of (6.30) respecting the restriction (6.27) has the form*

$$\begin{aligned} f(\Theta|K) &= \frac{\bar{f}(\Theta)\exp[\mu' g(\psi,\Theta)]}{\int \bar{f}(\Theta)\exp[\mu' g(\psi,\Theta)]\, d\Theta}, \quad \textit{where} && (6.31)\\ g(\psi,\Theta) &= \int H(d,\psi) f(d|\psi,\Theta)\, dd \qquad \textit{and } \mu \textit{ solves} && (6.32)\\ h(\psi) &= \frac{\int g(\psi,\Theta)\exp[\mu' g(\psi,\Theta)\bar{f}(\Theta)]\, d\Theta}{\int \exp[\mu' g(\psi,\Theta)\bar{f}(\Theta)]\, d\Theta}. && (6.33) \end{aligned}$$

Proof: Inserting (6.26), determined by an arbitrary $f(\Theta) \equiv f(\Theta|K) \in f_K^*$, into the restriction (6.27), we get

$$h(\psi) = \int H(d,\psi) \left[\int f(d|\psi,\Theta) f(\Theta)\, d\Theta\right] dd = \int g(\psi,\Theta) f(\Theta)\, d\Theta. \tag{6.34}$$

The second equality is implied by Fubini theorem on multiple integration and by the definition (6.32). The equality (6.34), normalization $\int f(\Theta)\, d\Theta = 1$ and non-negativity $f(\Theta) \geq 0$ determine the convex set where the optimal $f(\Theta|K)$ is searched for. Thus, we optimize the convex KL distance on the convex set. The corresponding Lagrangian, given by the vector Lagrangian multiplier μ, has the form

$$\int f(\Theta)\left[\ln\left(\frac{f(\Theta)}{\bar{f}(\Theta)}\right) - \mu' g(\psi,\Theta)\right] d\Theta = \int f(\Theta)\left[\ln\left(\frac{f(\Theta)}{\bar{f}(\Theta)\exp[\mu' g(\psi,\Theta)]}\right)\right] d\Theta =$$

$$= \int f(\Theta)\left[\ln\left(\frac{f(\Theta)}{\frac{\bar{f}(\Theta)\exp[\mu' g(\psi,\Theta)]}{\int \bar{f}(\tilde{\Theta})\exp[\mu' g(\psi,\tilde{\Theta})]\, d\tilde{\Theta}}}\right)\right] d\Theta - \ln\left(\int \bar{f}(\Theta)\exp[\mu' g(\psi,\Theta)]\, d\Theta\right).$$

Properties of the KL distance imply that the first term is minimized by the pdf (6.31). It is the overall minimizer as the second term is not influenced by this choice at all. The multiplier μ has to be chosen so that the restrictions (6.27) are met. $\square$

Remark(s) 6.3.2

1. *Solution of the algebraic equation (6.33) is often a difficult task.*

2. *The obtained form differs generally from the conjugate prior pdf. For this reason, a specific optimization is needed when the conjugate prior pdf (3.12) is required.*

3. *Majority existing solutions is related to a single-input u single-output y linear system with the state in phase form $\psi_t = [u_t, \ldots, u_{t-\partial_u}, y_{t-1}, \ldots, y_{t-\partial_y}, 1]$.*

   Knowledge of the static gain g is the simplest example of expressing the knowledge pieces in terms of (6.28). The following correspondence holds: $\mathcal{E}[g] \approx \hat{g} = \hat{d}$, $cov[g] \approx r_g = r_d$ and

   $$\psi' \equiv [\underbrace{1,\ldots,1}_{(\partial_u+1)\, times}, \underbrace{\hat{g},\ldots,\hat{g}}_{\partial_y\, times}].$$

   Dominant time constant, knowledge of a point on frequency response, measured step response, smoothness of step response are other examples elaborated for this model [68, 69, 70].

6.3.3 Merging of knowledge pieces

A combination of results in Sections 6.3.1, 6.3.2 gives a collection of prior estimates $f(\Theta|K_k)$, $k \in k^* \equiv \{1,\ldots,\mathring{k}\}$ obtained by the processing of individual, internally consistent, data blocks and of individual knowledge items. We have to use them for constructing a single estimate $\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k}))$ of the posterior pdf $f(\Theta|d(\mathring{t}), K(\mathring{k}))$ that reflects all of them as well as the measured data $d(\mathring{t})$. This posterior pdf should be close to the unknown posterior pdf $f(\Theta|d(\mathring{t}), K(\mathring{k}))$ that arises from the application of the Bayes rule starting from an unknown ideal combination $f(\Theta|K(\mathring{k}))$. For the construction of the estimate $\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k}))$, we

- can use available real data $d(\mathring{t})$ and pdfs $f(\Theta|K_k)$, $k \in k^*$: they form our experience $\mathcal{P}$ for the addressed estimation task,

- do not know mutual relationships of the prior estimates $f(\Theta|K_k)$, $k \in k^*$,

- are aware that the quality of individual pdfs $f(\Theta|K_k)$, $k \in k^*$ may vary substantially but in an unknown manner.

We face the problem of estimating the unknown pdf $f(\Theta|d(\mathring{t}), K(\mathring{k})) \in f^*$. The estimation of this infinite-dimensional object is hard. We solve it by adopting the following approximation.

Agreement 6.3.2 (Approximation of the estimated pdf) *Let $f_k^* \subset f^*$ contain those posterior pdfs $f(\Theta|d(\mathring{t}), K(\mathring{k}))$ for which $f(\Theta|d(\mathring{t}), K_k)$ is the best approximating pdf among $\left\{f(\Theta|d(\mathring{t}), K_k)\right\}_{k\in k^*}$.*

On f_k^*, we approximate $f \equiv f(\Theta|d(\mathring{t}), K(\mathring{k}))$ by the known $f_k \equiv f(\Theta|d(\mathring{t}), K_k)$, i.e.

$$f \equiv f(\Theta|d(\mathring{t}), K(\mathring{k})) \approx f(\Theta|d(\mathring{t}), K_k) \equiv f_k,\ \forall f(\Theta|d(\mathring{t}), K(\mathring{k})) \in f_k^*. \tag{6.35}$$

Proposition 6.3.2 (Merging of information sources) *Let the approximation (6.35) be adopted, the natural conditions of decision-making (2.36) met and the sets f_j^* be a priori equally probable.*

Then, the optimal merger $\hat{f}$ minimizing the expected KL distance $\mathcal{E}\left[\mathcal{D}\left(\hat{f}||f\right)\right]$, exploiting the experience $\mathcal{P}$ consisting of $d(\mathring{t})$ and $\{f(\Theta|K_k)\}_{k\in k^*}$ has the form

$$\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k})) \propto \mathcal{L}(\Theta, d(\mathring{t})) \prod_{k\in k^*} \left[f(\Theta|K_k)\right]^{\beta_{k|d(\mathring{t})}}. \tag{6.36}$$

In (6.36), $\mathcal{L}(\Theta, d(\mathring{t}))$ is the likelihood corresponding to the considered parameterized factor and the posterior probabilities $\beta_{k|d(\mathring{t})}$ of sets f_k^* are

$$\beta_{k|d(\mathring{t})} \propto \int \mathcal{L}(\Theta, d(\mathring{t})) f(\Theta|K_k)\, d\Theta. \tag{6.37}$$

Thus, the probabilities $\beta_{k|d(\mathring{t})}$ are equal to the normalized v-likelihood corresponding to respective choices $f(\Theta|K_k)$ of the prior pdfs.

Proof: Under (6.35), the minimization of the approximate expected KL distance can be performed as follows

$$\begin{aligned}
\operatorname{Arg}\min_{\hat{f}} \mathcal{E}\left[\mathcal{D}\left(\hat{f}||f\right)\right] &= \operatorname{Arg}\min_{\hat{f}} \mathcal{E}\left\{\sum_{k\in k^*} \chi_{f_k^*}(f)\mathcal{E}\left[\mathcal{D}\left(\hat{f}||f\right)|\mathcal{P}\right]\right\} \approx \\
\underbrace{\approx}_{(6.35)} &\operatorname{Arg}\min_{\hat{f}} \mathcal{E}\left\{\sum_{k\in k^*} \mathcal{E}\left[\chi_{f_k^*}(f)\mathcal{D}\left(\hat{f}||f_k\right)|\right]\right\} = \\
\underbrace{=}_{Proposition\ 2.3.7} &\operatorname{Arg}\min_{\hat{f}} \sum_{k\in k^*} \mathcal{E}\left[\chi_{f_k^*}(f)\mathcal{D}\left(\hat{f}||f_k\right)|\mathcal{P}\right] = \\
\underbrace{=}_{\substack{\beta_{k|d(\mathring{t})} \equiv \mathcal{E}[\chi_{f_k^*}(f)|\mathcal{P}] \\ \text{definition of } \mathcal{D}}} &\operatorname{Arg}\min_{\hat{f}} \int \hat{f}(\Theta|d(\mathring{t}), K(\mathring{k})) \ln\left[\frac{\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k}))}{\prod_{k\in k^*}\left[f(\Theta|d(\mathring{t}), K_k)\right]^{\beta_{k|d(\mathring{t})}}}\right] d\Theta.
\end{aligned} \tag{6.38}$$

Normalization of the denominator in the last fraction to a pdf and the use of the second property of the KL distance, Proposition 2.4.3, imply the form (6.36) of the approximating pdf. To prove remaining statements, it is sufficient realize that the pdfs combined into the geometric mean have the common factor equal to the likelihood function $\mathcal{L}(\Theta, d(\mathring{t}))$. The weight $\beta_{k|d(\mathring{t})}$ is the posterior probability that $f_k \equiv f(\Theta|K_k)$ is the closest pdf to $f(\Theta|K(\mathring{k}))$, i.e. $f(\Theta|K(\mathring{k})) \in f_k^*$. Values of these posterior probabilities are simply obtained by the Bayes rule applied to hypotheses $f(\Theta|K(\mathring{k})) \in f_k^*,\ k \in k^*$, starting from the uniform prior on them. It proves the formula (6.37). □

Algorithm 6.3.2 (Merging knowledge pieces)

1. *Apply Algorithm 6.3.1 to get $f(\Theta|K_k),\ k \in k^*$ on internally-consistent data blocks.*
2. *Quantify individual knowledge pieces as described by Proposition 6.3.1.*
3. *Evaluate the likelihood function $\mathcal{L}(\Theta, d(\mathring{t}))$ corresponding to the considered parameterized model.*
4. *Evaluate the v-likelihood $f(d(\mathring{t})|K_k) = \int \mathcal{L}(\Theta, d(\mathring{t})) f(\Theta|K_k)\, d\Theta$ with prior pdfs $f(\Theta|K_k),\ k \in k^*$.*

5. *Evaluate weights*

$$\beta_{k|d(\mathring{t})} = \frac{f(d(\mathring{t})|K_k)}{\sum_{\tilde{k}\in k^*} f(d(\mathring{t})|K_{\tilde{k}})}, \quad k \in k^*.$$

6. *Determine the merger as the posterior pdf to be used*

$$\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k})) \propto \mathcal{L}(\Theta, d(\mathring{t})) \prod_{k\in k^*} [f(\Theta|K_k)]^{\beta_{k|d(\mathring{t})}}.$$

Note that the used "prior pdf" $\prod_{k\in k^*}[f(\Theta|K_k)]^{\beta_{k|d(\mathring{t})}}$ is seen in the last step of the merging algorithm.

6.4 Construction of the prior estimate

It is known [49] that mixtures cannot be identified uniquely. This means that the estimation results depend on the prior estimates of parameters of the individual factors, of the component weights as well as on the overall structure used. The need to use an approximate estimation, Section 6.5, makes the situation even harder. Thus, the *choice of the prior pdf* determines significantly quality of the results.

All clustering and mixture estimation techniques face a similar *initialization problem.* There is a lot of algorithms for approaching it, for instance [71]. We have, however, found no good solution suitable for estimation of mixtures with high-dimensional *dynamic components* and a large estimation horizon $\mathring{t}$. Our, probably novel, approach is gradually presented here.

6.4.1 Iterative construction of prior pdf

The Bayes rule specifies the posterior estimates $f(\Theta|d(\mathring{t})) \propto f(d(\mathring{t})|\Theta)f(\Theta)$ in a non-iterative fashion. At the same time, iterative data-based constructions of $f(\Theta)$ are practically and successfully used. They face the following common and often overlooked danger.

Let the pdf $f_n(\Theta|d(\mathring{t}))$ be the pdf obtained through the n-th formal repetition of the Bayes rule

$$f_n(\Theta|d(\mathring{t})) \propto f(d(\mathring{t})|\Theta)f_{n-1}(\Theta|d(\mathring{t}))$$

with $f_0(\Theta|d(\mathring{t})) \equiv f(\Theta)$. Then, $f_n(\Theta|d(\mathring{t})) \propto [f(d(\mathring{t})|\Theta)]^n f(\Theta)$. For $f(\Theta) > 0$ on Θ^*, the pdf $f_n(\Theta|d(\mathring{t}))$ concentrates on Θ maximizing globally the likelihood function $f(d(\mathring{t})|\Theta)$. This seems to be a good property. The convergence to a meaningful point is, however, likely to be spoiled when the involved likelihood $f(d(\mathring{t})|\Theta)$ is evaluated approximately only as it must be done for mixtures. Moreover, the pdf obtained in this way looses completely information on precision of such an estimate.

The following re-interpretation and modification of the iterative evaluations resolves the outlined problem.

Iterations are used when we are uncertain about the adequate prior pdf. Moreover, the likelihood function is evaluated approximately only and the quality of the approximation depends on the prior pdf used. Thus, the *likelihood function is uncertain*, too. According to the adopted Bayesian philosophy, any uncertainty should be treated as randomness. Thus, the joint pdf $f(d(\mathring{t}), \Theta)$ of data $d(\mathring{t})$ and parameters Θ becomes an infinite-dimensional unknown (hyper)parameter. The mixture estimation of interest specifies the set $f^*(d(\mathring{t}), \Theta)$ of the possible joint pdfs. The considered estimation task is then characterized by the following agreement.

Agreement 6.4.1 (Estimation of the joint pdf of data and parameters) *An estimate $\hat{f}(d(\mathring{t}), \Theta)$ of the joint pdf of data and parameters $f(d(\mathring{t}), \Theta)$ is searched for within the set*

$$\begin{aligned}
f^*(\cdot,\cdot) &\equiv \left\{f(d(\mathring{t}),\Theta) \equiv \mathcal{L}(\Theta, d(\mathring{t}))f(\Theta)\right\} \quad \textit{with the likelihood} \\
\mathcal{L}(\Theta, d(\mathring{t})) &= \prod_{t\in t^*} f(d_t|d(t-1),\Theta) \quad \textit{defined by the parameterized model} \qquad (6.39)\\
f(d_t|d(t-1),\Theta) &= \sum_{c\in c^*} \alpha_c f(d_t|d(t-1),\Theta_c,c). \quad \textit{Its components are products of factors}
\end{aligned}$$

$$f(d_t|d(t-1), \Theta_c, c) = \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)$$

and the prior mixture estimate $f(\Theta)$ is a product of (conjugate) prior pdfs

$$f(\Theta) = Di_\alpha(\kappa_0) \prod_{i \in i^*} f(\Theta_{ic}), \text{ see Agreement 6.1.1.}$$

The constructed estimator $d^*(\mathring{t}) \to f^*$ is required to specify a point estimate $\hat{f}(\cdot,\cdot) \in f^*(\cdot,\cdot)$ of the joint pdf $f(\cdot,\cdot)$ of data and parameters. The evaluations have to stop after a finite number of evaluation steps.

The application of the general decision-making theory needs a specification of the expected loss to be optimized. It calls for specification of probabilistic measure on $f^*(\cdot,\cdot)$. It is difficult both technically and practically. In order to avoid this difficulty, we interpret the maximum likelihood estimate as a point Bayesian estimate corresponding to a special prior distribution. This interpretation is correct under some technical conditions, e.g. [72]. We assume that these conditions are applicable in our case. Then, we have to describe the corresponding (hyper)likelihood function ${}^{\lfloor h}\mathcal{L}(f(\cdot,\cdot), d(\mathring{t}))$ only. It means that we have to specify the functional dependence of the distribution of the observed data $d(\mathring{t})$ on the estimated infinite-dimensional (hyper)parameter $f(\cdot,\cdot) \equiv f(d(\mathring{t}), \Theta)$. This likelihood equals to the marginal pdf of data $d(\mathring{t})$ computed for each unknown (hyper)parameter coinciding with the unknown joint pdf $f(\cdot,\cdot)$ of data and parameters. Thus,

$${}^{\lfloor h}\mathcal{L}(f(\cdot,\cdot), d(\mathring{t})) \equiv f(d(\mathring{t})) = \int f(d(\mathring{t}), \Theta)\, d\Theta, \text{ where} \tag{6.40}$$

the last equality is implied by the marginalization rule, Proposition 2.3.4.

These considerations lead us to the choice of the estimate $\hat{f}(d(\mathring{t}), \Theta)$ of the joint pdf $f(d(\mathring{t}), \Theta)$ of data and parameters as

$$\hat{f}(d(\mathring{t}), \Theta) \in \operatorname{Arg} \max_{\tilde{f} \in f^*(\cdot,\cdot)} \int \tilde{f}(d(\mathring{t}), \Theta)\, d\Theta. \tag{6.41}$$

For any $\tilde{f} \in f^*(\cdot,\cdot)$ (6.39), we are able to evaluate (approximately) the maximized functional (6.41). We face an "ordinary" optimization problem over the infinite dimensional domain (6.39). We tackle it by the branch and bound techniques prepared in Section 6.1.3. Moreover, we use the fact that for a chosen guess of the prior pdf $\hat{f}(\Theta)$ the approximation of the likelihood $\mathcal{L}$ in (6.39) is uniquely determined by the algorithm adopted for the approximate estimation, see Section 6.5.

Note that forgetting can be used during the search. The final estimates have to be done without forgetting as we need a time-invariant description of the o-system. We summarize the above thoughts.

Agreement 6.4.2 (Iterative search for the joint pdf of data and parameters)
We search for such a variant of $\hat{f}(\Theta)$ for which the joint pdf $\hat{f}(d(\mathring{t}), \Theta)$ in $f^*(\cdot,\cdot)$, Agreement 6.4.1, evaluated by an approximate algorithm, Section 6.5, with the unit forgetting exhibits the highest v-likelihood $\hat{f}(d(\mathring{t}))$. The dimensionality of the problem directs us towards the maximization by the branch and bound techniques, Section 6.1.3.

6.4.2 Common bounding mapping

The bounding mapping $\mathcal{U}$ (6.18) adopted further on is simple and universal. The arguments $X_{i(n+1|n)}$ with smallest values $F(X_{i(n+1|n)})$ are omitted when forming the new set of candidates X^*_{n+1}.

It is necessary to stress that for its application it is useful to take into account the following points.

- Values $F(X_{i(n+1|n)})$ for omitted arguments have to be known and compared with the best value $\bar{F}_{n+1|n}$. This trivial observation can be (and was) overlooked in a complex optimization setting.

- The number of preserved or omitted arguments is the only optional parameter of this mapping. It is mostly dictated by the adopted branching mapping that needs usually several arguments in X^*_n for generating new promising candidates to form $X^*_{n+1|n}$.

6.4.3 Flattening mapping

Our discussion on the choice of the prior estimate started with an inspection of the iterative application of the Bayes rule. Its basic idea – the posterior pdf $f(\Theta|d(\mathring{t}))$ gives a better clue about Θ than the prior pdf $f(\Theta)$ – is elaborated here.

The posterior estimate of parameters cannot be directly used as a new guess of the prior pdf as it is much more concentrated than a reasonable prior pdf. It is dangerous in the discussed context, as the original prior has been assumed unreliable and thus the posterior pdf may be concentrated at false subset of Θ^*. In order to avoid this danger, we have to apply a *flattening mapping* $\mathcal{G}$

$$\mathcal{G} : f(\Theta|d(\mathring{t})) \to \hat{f}(\Theta) \tag{6.42}$$

to get a new guess of $\hat{f}(\Theta)$ of the prior pdf $f(\Theta)$. It has to be designed so that a compromise between the following contradictory requirements is reached.

- $\hat{f}(\Theta)$ should resemble $f(\Theta|d(t))$,
- $\hat{f}(\Theta)$ should be flat enough.

We design $\mathcal{G}$ by solving a suitable optimization task.

For the fixed $d(\mathring{t})$, we denote $\tilde{f}(\Theta) \equiv f(\Theta|d(\mathring{t}))$. Moreover, we need a pdf $\bar{f}(\Theta)$ that serves us as a prototype of the flat distribution. Uniform pdf (even improper one) suits often to this purpose.

The value $\hat{f}$ of the flattening mapping $\mathcal{G}$ (6.42) is found as a minimizing argument of the functional

$$\mathcal{D}\left(\hat{f}||\tilde{f}\right) + q\mathcal{D}\left(\hat{f}||\bar{f}\right),\ q > 0. \tag{6.43}$$

The KL distance (2.26) $\mathcal{D}\left(\hat{f}||\tilde{f}\right)$ reflects the first requirement and $\mathcal{D}\left(\hat{f}||\bar{f}\right)$ the second one. The positive weight q is the design parameter that controls the compromise.

Proposition 6.4.1 (Optimal flattening mapping) *Let $\tilde{f}$ and $\bar{f}$ be a given pair of parameter estimates defined on Θ^*. Then, the pdf $\hat{f}$, defined on Θ^* minimizing the functional (6.43) has the form*

$$\hat{f} \propto \tilde{f}^{\Lambda}\bar{f}^{1-\Lambda} \quad \textit{with} \quad \Lambda = 1/(1+q) \in (0,1). \tag{6.44}$$

Proof: It holds

$$\begin{aligned}
\hat{f} &\in \operatorname{Arg}\min_{f\in f^*} \mathcal{D}\left(f||\tilde{f}\right) + q\mathcal{D}\left(f||\bar{f}\right) \underbrace{\equiv}_{definition\ of\ \mathcal{D}} \operatorname{Arg}\min_{f\in f^*} \int f\left[\ln\left(\frac{f}{\tilde{f}}\right) + q\ln\left(\frac{f}{\bar{f}}\right)\right] d\Theta \equiv \\
&\equiv \operatorname{Arg}\min_{f\in f^*} (1+q)\int f \ln\left(\frac{f}{\tilde{f}^{1/(1+q)}\bar{f}^{q/(1+q)}}\, d\Theta\right) \underbrace{\equiv}_{(6.44)} \hat{f}.
\end{aligned}$$

The last identity is implied by independence of the normalizing shift $\ln\left(\int \tilde{f}^{1/(1+q)}\bar{f}^{q/(1+q)}\, d\Theta\right)$ of the optional pdf f, by the positivity of $1+q$, definition of Λ, definition of $\hat{f}$ and Proposition 2.4.3, (6.44). □

The pdf $\hat{f}$ found in Proposition 6.4.1 coincides formally with a geometric mean of a pair of pdfs, see Proposition 3.1.1. The power Λ has, however, another interpretation. It is controlled by the optional weight $q > 0 \Leftrightarrow \Lambda \in (0,1)$ balancing the Bayes rule and the flattening.

The adopted Agreement 6.1.1 implies that the estimate of mixture parameters Θ is a product of the Dirichlet pdf describing component weights α and the product of pdfs describing parameters Θ_{ic} of individual factors. The KL distance of a pair of such products is simply sum of KL distances between the corresponding marginal pdfs creating the product. Consequently, Proposition 6.4.1 can be applied to each of them with its specific weight and thus its specific Λ.

Flattening was motivated by an iterative application of Bayes rule. The combination of the Bayes rule and flattening is described by the operator

$$\mathcal{A}_{\Lambda} : f(\Theta) \to \frac{\left[\mathcal{L}(\Theta, d(\mathring{t}))f(\Theta)\right]^{\Lambda}[\bar{f}(\Theta)]^{1-\Lambda}}{\int \left[\mathcal{L}(\Theta, d(\mathring{t}))f(\Theta)\right]^{\Lambda}[\bar{f}(\Theta)]^{1-\Lambda}\, d\Theta}. \tag{6.45}$$

Its use requires specification of the factor Λ. We inspect influence of Λ on results of the repetitive use of $\mathcal{A}_\Lambda$ in the ideal case when the likelihood $\mathcal{L}(\Theta, d(\mathring{t})) \equiv f(d(\mathring{t})|\Theta)$ is evaluated exactly in each iteration.

Proposition 6.4.2 (Asymptotic of the repetitive use of $\mathcal{A}_\Lambda$) *Let $\Lambda(n) \in [0, 1-\varepsilon]^n$, $\varepsilon \in (0,1)$, be a sequence of weights determining operators $\mathcal{A}_n \equiv \mathcal{A}_{\Lambda_n}$ (6.45) with a common likelihood $\mathcal{L}(\Theta, d(\mathring{t})) \equiv f(d(\mathring{t})|\Theta)$. Let the common support of the prior pdf $f(\Theta)$ and the (flat) pre-prior pdf $\bar{f}(\Theta)$ contain the support of $\mathcal{L}(\Theta, d(\mathring{t}))$.*

Then, the pdfs $f_n \equiv \mathcal{A}_n \mathcal{A}_{n-1} \ldots \mathcal{A}_1[f]$ have the following possible stationary value

$$f_\infty \equiv \limsup_{n\to\infty} f_n \propto \mathcal{L}^A \bar{f}^B, \quad \text{where the weight } A \in [0,1) \text{ equals to} \tag{6.46}$$

$$A \equiv \limsup_{n\to\infty} A_n, \; A_n \equiv \sum_{i=1}^{n} \prod_{j=i}^{n} \Lambda_j \equiv \Lambda_n(1 + A_{n-1}), \; A_0 = 0, \quad \text{and} \tag{6.47}$$

$$B \equiv \limsup_{n\to\infty} B_n, \; B_n \equiv \sum_{i=1}^{n} (1-\Lambda_i) \prod_{j=i+1}^{n} \Lambda_j \equiv (1-\Lambda_n) + \Lambda_n B_{n-1}, \; B_0 = 0. \tag{6.48}$$

The asymptotic pdf f_∞ equals to $\mathcal{L}^{\Lambda/(1-\Lambda)}\bar{f}$ for a constant Λ. For $\Lambda = 0.5$, the pdf f_∞ coincides with that resulting from the Bayes rule initiated by the pre-prior pdf $\bar{f}$.

The weight $\Lambda = 0.5$ is the desirable asymptotic value of varying Λ_n. For it, we get $A = B = 1$, leading to f_∞ coinciding with the posterior pdf for the prior pdf equal to $\bar{f}$.

Proof: The repetitive application of $\mathcal{A}_n$ gives $f_n = \mathcal{L}^{A_n} f^{\prod_{\tilde{n}=1}^{n} \Lambda_{\tilde{n}}} \bar{f}^{B_n}$. The non-negative sequence $\{A_n\}_{n\geq 1}$ is given by the formula (6.47) and is bounded from above by the sum of the geometric sequence with the first term and quotient equal to $\bar{\Lambda}$. Thus, the $\limsup_{n\to\infty} A_n$ exists and it is smaller than $1/\varepsilon - 1$.

The power of the prior pdf f converges quickly to zero as all factors $\Lambda_n \leq 1-\varepsilon$. The power B_n of the pre-prior pdf $\bar{f}$ evolves according to (6.48) and it is bounded by unity. Thus, its limes superior is smaller than one. For a constant Λ, A_n becomes sum of geometric sequence with value converging to $\Lambda/(1-\Lambda)$ and B_n to unity. □

Remark(s) 6.4.1

1. *Uniform separation of Λ_n from unity by $\varepsilon > 0$ corresponds with a uniformly positive penalty $q_n = q$ (6.43). It is desirable restriction as it prevents to apply the plain Bayes rule again and again. Thus, it avoids drawbacks discussed in Section 6.4.1.*
2. *The chosen "informative" prior pdf does not influence the asymptotic solution. Its preservation, however, may be of an extreme importance at the beginning of iterations as it may keep the approximately evaluated likelihood close to the correct one.*

A change of the alternative pdf used in the flattening during iterative learning offers a remedy of the problem mentioned in the above remark. The simplest possibility is described by the following algorithm.

Algorithm 6.4.1 (Iterative learning with the preserved prior pdf)
Initial mode

- *Select the upper bound $\mathring{n}$ on the number n of iterations.*
- *Set the iteration counter $n = 1$ and select the flattening rate Λ_1.*
- *Select a flat alternative pdf, typically, improper one $\bar{f} \propto 1$.*
- *Specify a proper prior pdf $f_0 = f$.*
- *Compute a new guess of the prior pdf f_1 using the approximate estimation and flattening*

$$f_1 \propto [\mathcal{L} f_0]^{\Lambda_1} \bar{f}^{1-\Lambda_1}.$$

- *Re-define the alternative pdf to* $\bar{f} = f_0$.

Iterative mode

1. *Stop if* $n \geq \mathring{n}$.
2. *Increase the iteration counter* $n = n + 1$ *and select the flattening rate* Λ_n.
3. *Compute a new guess of the pdf prior pdf* f_n *using approximate estimation and flattening*
$$f_n \propto [\mathcal{L} f_{n-1}]^{\Lambda_n} f_0^{1-\Lambda_n}.$$
4. *Go to the beginning of* Iterative mode.

Proposition 6.4.3 (Asymptotic of Algorithm 6.4.1) *Let* $\Lambda(n) \in [0, 1-\varepsilon]^n$, $\varepsilon \in (0,1)$, *be a sequence of weights determining operators* $\mathcal{A}_n \equiv \mathcal{A}_{\Lambda_n}$ *(6.45) with a common likelihood* $\mathcal{L}(\Theta, d(\mathring{t})) \equiv f(d(\mathring{t})|\Theta)$. *Let the common support of the prior pdf* $f(\Theta)$ *and the (flat) pre-prior pdf* $\bar{f}(\Theta)$ *contain the support of* $\mathcal{L}(\Theta, d(\mathring{t}))$.

Then, the pdfs f_n *generated by Algorithm 6.4.1 have the following possible stationary value*

$$\begin{aligned} f_\infty &\equiv \limsup_{n\to\infty} f_n \propto \mathcal{L}^A f_0^B, \quad \text{where the weight} && (6.49)\\ A &\equiv \limsup_{n\to\infty} A_n = \limsup_{n\to\infty} \Lambda_n(1 + A_{n-1}),\ A_1 = \Lambda_1, \quad \text{and} \\ B &\equiv \limsup_{n\to\infty} B_n = B_n \equiv (1 - \Lambda_n) + \Lambda_n B_{n-1},\ B_1 = \Lambda_1. \end{aligned}$$

The asymptotic pdf becomes $f_\infty = \mathcal{L}^{\Lambda/(1-\Lambda)} f_0$, *for a constant* $\Lambda = \Lambda_n$, $n > 1$. *For* $\Lambda = 0.5$, *the pdf* f_∞ *coincides with that resulting from the Bayes rule initiated by the prior pdf* f_0.

The value $\Lambda = 0.5$ *is the desirable limits of varying* Λ_n. *For it, we get* $A = B = 1$, *leading to* f_∞ *coinciding with the posterior pdf determined by the prior pdf* f_0.

Proof: Omitted as it essentially copies the proof of Proposition 6.4.2. □

Flattening is often applied as a part of branching. Then, it is effectively used just once and the asymptotic analysis gives no hint how to choose the flattening parameter Λ. Let us discuss this problem.

For the Dirichlet pdf $Di_\alpha(\kappa)$ (6.4) estimating component weights, the sum $\sum_{c\in c^*} \kappa_c$ has meaning of the number of observed and fictitious data, see Chapter 10. Thus, without forgetting, the prior and posterior values of this sum differ by the number of data $\mathring{t}$. This difference should be suppressed if the flattened pdf serves as the prior pdf. The flattened version should be based on a same number of data as the prior pdf. This gives the following recommendations.

Algorithm 6.4.2 (Recommended flattening of the Dirichlet pdf in branching) *Let the preprior pdf be Dirichlet* $Di_\alpha(\bar{\kappa})$, *the prior one be* $Di_\alpha(\kappa_0)$ *and the posterior one* $Di_\alpha(\kappa_{\mathring{t}})$. *Then, the new prior obtained by flattening should be* $Di_\alpha(\hat{\kappa}_0)$ *with*

$$\hat{\kappa}_0 = \Lambda \kappa_{\mathring{t}} + (1 - \Lambda)\bar{\kappa}, \quad \text{where } \Lambda = \frac{\sum_{c\in c^*}(\kappa_{c;0} - \bar{\kappa}_c)}{\sum_{c\in c^*}(\kappa_{c;\mathring{t}} - \bar{\kappa}_c)}. \tag{6.50}$$

Remark(s) 6.4.2

1. *Experiments indicate that* $\bar{\kappa} \approx 0$ *and* $\kappa_0 \approx 0.1\mathring{t}$ *are suitable options. Then,* $\Lambda \approx 0.1$.
2. *A positive flattening factor* Λ *is obtained iff* $\sum_{c\in c^*} \kappa_{c;0} > \sum_{c\in c^*} \bar{\kappa}_c$. *For the non-iterative flattening, it is sufficient and reasonable to take the lowest value* $\bar{\kappa} = 0$.
3. *Meaningful* $\Lambda < 1$ *is obtained if* $B_{\mathring{t}} \equiv \sum_{c\in c^*} \kappa_{c;\mathring{t}} > B_0 \equiv \sum_{c\in c^*} \kappa_{c;0}$. *It is naturally fulfilled when estimating without forgetting. This condition has to be checked whenever exponential forgetting is employed. For the constant forgetting rate with a flat alternative pdf, this condition is equivalent to the requirement*
$$\textit{forgetting\_factor} > 1 - \frac{1}{B_0 - \bar{B}},\ \bar{B} \equiv \sum_{c\in c^*} \bar{\kappa}_c. \tag{6.51}$$

Problem 6.4.1 (Various iterative-learning scenarios) *The adequate choice of the varying flattening rate depends on its specific use. For this reason, it would be reasonable to think over various iterative-learning scenarios. Ideally, a sort of feedback during iterations could and should be used.*

We assume that all pdfs in $\prod_{i\in i^*}\prod_{c\in c^*} f(\Theta_{ic})$ are conjugate pdfs of the parameterized factors belonging to exponential family, Section 3.2. In this family, the effective number of exploited, possibly fictitious, records is stored in counters ν_{ic} for each in $f(\Theta_{ic})$. All considered approximate estimation, see Section 6.5, add a fraction of the data mass to individual factors. We can guarantee at most that $\sum_{ic}\nu_{ic;\mathring{t}} = \mathring{d}\mathring{t} + \sum_{ic}\nu_{ic;0}$. Thus, we cannot flatten factor-wise and have to use a common flattening factor to all of them. It is chosen so that $\sum_{ic}\nu_{ic;\mathring{t}}$ decreases to $\sum_{ic}\nu_{ic;0}$. This gives the following algorithm.

Algorithm 6.4.3 (Recommended flattening of $\prod_{ic} f(\Theta_{ic}|d(\mathring{t}))$ in branching) *Let us consider factors in the exponential family and the product forms of*

$$\begin{array}{rll} \textit{the conjugate pre-prior pdf} & \prod_{i\in i^*,c\in c^*} & A^{\bar{\nu}_{ic}}(\Theta_{ic})\exp\left[\left\langle \bar{V}_{ic}, C(\Theta)\right\rangle\right]\chi_{\Theta^*_{ic}}(\Theta_{ic}) \\ \textit{the conjugate prior pdf} & \prod_{i\in i^*,c\in c^*} & A^{\nu_{ic;0}}(\Theta_{ic})\exp\left[\langle V_{ic;0}, C(\Theta)\rangle\right]\chi_{\Theta^*_{ic}}(\Theta_{ic}) \\ \textit{the posterior pdf} & \prod_{i\in i^*,c\in c^*} & A^{\nu_{ic;\mathring{t}}}(\Theta_{ic})\exp\left[\left\langle V_{ic;\mathring{t}}, C(\Theta)\right\rangle\right]\chi_{\Theta^*_{ic}}(\Theta_{ic}). \end{array} \tag{6.52}$$

Then, the prior pdf obtained by flattening is $\prod_{i\in i^*,c\in c^*c} A^{\hat{\nu}_{ic;0}}(\Theta)\exp\left[\left\langle \hat{V}_{ic;0}, C(\Theta)\right\rangle\right]\chi_{\Theta^*_{ic}}(\Theta_{ic})$ *with*

$$\begin{aligned} \hat{V}_{ic;0} &= \Lambda V_{ic;\mathring{t}} + (1-\Lambda)\bar{V}_{ic}, \quad \hat{\nu}_{ic;0} = \Lambda\nu_{ic;\mathring{t}} + (1-\Lambda)\bar{\nu}_{ic}, \quad \textit{where} \\ \Lambda &= \frac{\sum_{i\in i^*}\sum_{c\in c^*}(\nu_{ic;0}-\bar{\nu}_{ic})}{\sum_{i\in i^*}\sum_{c\in c^*}(\nu_{ic;\mathring{t}}-\bar{\nu}_{ic})}, \ \forall i\in\{1,\ldots,\mathring{d}\},\ c\in c^*. \end{aligned} \tag{6.53}$$

Remark(s) 6.4.3

1. *A positive flattening factor Λ is obtained iff $\sum_{i\in i^*}\sum_{c\in c^*}\nu_{ic;0} > \sum_{i\in i^*}\sum_{c\in c^*}\bar{\nu}_{ic}$.*
   It is guaranteed when we select $\bar{\nu}_c = 0,\ c\in c^*$.
2. *Meaningful $\Lambda < 1$ is obtained if $A_{\mathring{t}} \equiv \sum_{i\in i^*}\sum_{c\in c^*}\nu_{ic;\mathring{t}} > A_0 \equiv \sum_{i\in i^*}\sum_{c\in c^*}\nu_{ic;0}$. It is naturally fulfilled when the estimation is performed without forgetting. This condition has to be checked whenever the exponential forgetting is employed. For the constant forgetting rate with a flat alternative pdf, this condition is equivalent to the requirement*

$$\textit{forgetting\_factor} > 1 - \frac{1}{A_0 - \bar{A}},\ \bar{A} \equiv \sum_{i\in i^*}\sum_{c\in c^*}\bar{\nu}_{ic}. \tag{6.54}$$

3. *Use of stabilized forgetting avoids the problem discussed in the previous item. It can be used fully for the Dirichlet part of pdfs. For the respective factors in the exponential family, it is sufficient to select a positive degree $\nu_{ic;0}$ of the factor $A(\Theta_{ic})$ in the exponential family, cf. (3.5), serving as the alternative pdf. Then, the inspected sums do not fall and no special actions are needed during flattening.*

In the real iterative mode when the likelihood is evaluated approximately, it is reasonable to use a varying flattening rate Λ_n. We would like to have $A_{\mathring{n}} = 1$ after the specified number of iterations $\mathring{n} > 1$. For $\mathring{n} = 1$, the values of Λ are recommended in Algorithms 6.4.2, 6.4.3. We take them respectively as initial conditions $A_1 \equiv \Lambda_1$ for the Dirichlet (6.4) and exponential factors (6.52). If we specify, moreover, a function describing the desired increase of A_n to $A_{\mathring{n}} = 1$ we get the rule how to compute respective values of Λ_n. We do that for a simple linear function.

Proposition 6.4.4 (Flattening rate for linearly growing confidence into likelihood) *Let us repeat flattening for $\mathring{n} > 1$ steps and select*

$$\begin{aligned} A_1 &\equiv \Lambda_1 = \frac{\sum_{c\in c^*}(\kappa_{c;0}-\bar{\kappa}_c)}{\sum_{c\in c^*}(\kappa_{c;\mathring{t}}-\bar{\kappa}_c)} \quad \textit{for flattening of the Dirichlet factor in (6.3) and} \\ A_1 &\equiv \Lambda_1 = \frac{\sum_{i\in i^*,c\in c^*}(\nu_{ic;0}-\bar{\nu}_{ic})}{\sum_{i\in i^*,c\in c^*}(\nu_{ic;\mathring{t}}-\bar{\nu}_{ic})} \quad \textit{for the exponential factors (6.52) in (6.3).} \end{aligned}$$

Let us require $A_{\mathring{n}} = 1$ after a linear growth. Then, for $n > 1$

$$A_n = A_{n-1} + k,\ k \equiv \frac{1 - A_1}{\mathring{n} - 1},\quad \Lambda_n = \frac{A_n}{1 + A_{n-1}}. \tag{6.55}$$

Proof: The required linear growth and the fixed initial condition implies $A_n = A_1 + k(n-1)$. The requirement $A_{\mathring{n}} = 1$ gives $k = \frac{1-A_1}{\mathring{n}-1}$ and (6.47) implies

$$\Lambda_n = \frac{A_n}{1 + A_{n-1}} \equiv \frac{A_1 + k(n-1)}{1 + A_1 + k(n-2)}.$$

□

Asymptotic analysis of the flattening is made for the common exactly evaluated likelihood $\mathcal{L}(\Theta, d(\mathring{t}))$. Flattening is, however, justified by a violation of this assumption. Thus, it makes sense to model this violation and consequently to modify variations of the flattening rate. In this respect, we assume that in n-th flattening iteration

$$\mathcal{L}_n(\Theta, d(\mathring{t})) = \left[\mathcal{L}(\Theta, d(\mathring{t}))\right]^{\beta_n},\ \text{where} \tag{6.56}$$

$\beta_n \in (0, 1]$ is a known scalar. This model is motivated by experiments with estimation employing forgetting, see Section 6.4.7. With forgetting $\sum_{c \in c^*} \nu_{c;\mathring{t}} < \sum_{c \in c^*} \nu_{c;0} + \mathring{\mathring{d}t}$, cf. Remarks 6.4.3, and thus $\mathcal{L}_n(\Theta, d(\mathring{t}))$ cannot be equal to the exact $\mathcal{L}(\Theta, d(\mathring{t}))$. The likelihood is simply more flat than assumed up to now. The ratio

$$\beta_n \quad = \quad \frac{\sum_{c \in c^*} \nu_{nc;\mathring{t}}}{\sum_{c \in c^*} \nu_{nc;0} + \sum_{t=1}^{\mathring{t}} \lambda^t},\ \text{where } \lambda \text{ is used forgetting factor,} \tag{6.57}$$

indicates the degree of flattening in n-th iterative step. Let us repeat the asymptotic analysis of Algorithm 6.4.1 under the assumption (6.56).

Proposition 6.4.5 (Asymptotics of Algorithm 6.4.1 under (6.56)) *Let $\Lambda(n) \in [0, 1 - \varepsilon]^n$, $\varepsilon \in (0, 1)$, be a sequence of weights determining operators $\mathcal{A}_n \equiv \mathcal{A}_{\Lambda_n}$ (6.45) with the varying likelihood $\mathcal{L}_n(\Theta, d(\mathring{t})) = [\mathcal{L}(\Theta, d(\mathring{t}))]^{\beta_n} \equiv [f(d(\mathring{t})|\Theta)]^{\beta_n}$. Let the common support of the prior pdf $f(\Theta) = f_0(\Theta)$ and of the (flat) pre-prior pdf $\bar{f}(\Theta)$ contain the support of $\mathcal{L}(\Theta, d(\mathring{t}))$.*

Then, the pdfs f_n generated by Algorithm 6.4.1 have the following possible stationary value

$$\begin{aligned} f_\infty &= \limsup_{n\to\infty} f_n \propto \mathcal{L}^A f_0^B, \quad \textit{where} \\ A &= \limsup_{n\to\infty} A_n = \limsup_{n\to\infty} \Lambda_n(\beta_n + A_{n-1}),\ A_1 = \Lambda_1, \quad \textit{and} \\ B &= \limsup_{n\to\infty} B_n = B_n \equiv (1 - \Lambda_n) + \Lambda_n B_{n-1},\ B_1 = \Lambda_1. \end{aligned} \tag{6.58}$$

The desirable limit is $A = B = 1$ when f_∞ coincides with the posterior pdf given by the prior pdf f_0.

Proof: Let us assume that, for $n > 1$, $f_{n-1} = \mathcal{L}^{A_{n-1}} f_0^{B_{n-1}} \bar{f}^{C_{n-1}}$. The estimation and flattening steps, together with the last assumption and with (6.56), imply

$$\begin{aligned} f_n &= [\mathcal{L}_n f_{n-1}]^{\Lambda_n} f_0^{1-\Lambda_n} = \left[\mathcal{L}^{\beta_n + A_{n-1}} f_0^{B_{n-1}} \bar{f}^{C_{n-1}}\right]^{\Lambda_n} f_0^{1-\Lambda_n} \Rightarrow \\ A_n &= \Lambda_n(\beta_n + A_{n-1}),\ B_n - 1 = \Lambda_n(B_{n-1} - 1),\ C_n = \Lambda_n C_{n-1}. \end{aligned}$$

Thus, $B_n \to 1$ and $C_n \to 0$ whenever $\prod_{n=1}^{\mathring{n}} \Lambda_n \to 0$. For it, it is sufficient to have $\Lambda_n \in [0, 1-\varepsilon]$, $1 > \varepsilon > 0$. Then also the non-stationary difference equation for A_n is stable and the requirement $A_n \to 1$ is the only serious restriction on the sequence Λ_n. □

The discussed modification leads to the modification of Proposition 6.4.4.

Proposition 6.4.6 (Flattening rate for almost linearly growing confidence in $\mathcal{L}_n$ (6.56)) *Let us assume that β_n, β_{nD} are given and repeat flattening for $\mathring{n} > 1$ steps. Let us select*

$$A_{1D} \equiv \beta_{1D}\Lambda_{1D},\ \Lambda_{1D} = \frac{\sum_{c\in c^*}(\kappa_{c;0} - \bar{\kappa}_c)}{\sum_{c\in c^*}(\kappa_{c;\mathring{t}} - \bar{\kappa}_c)}$$

for flattening of Dirichlet factor (6.4) in (6.3) and

$$A_1 \equiv \beta_1\Lambda_1,\ \Lambda_1 = \frac{\sum_{i\in i^*, c\in c^*}(\nu_{ic;0} - \bar{\nu}_{ic})}{\sum_{i\in i^*, c\in c^*}(\nu_{ic;\mathring{t}} - \bar{\nu}_{ic})}$$

for the exponential factors (6.52) in (6.3). Let us require $A_{\mathring{n}} = 1$ after a linear growth under (6.56).

Then, for $n > 1$

$$\begin{aligned} A_n &= A_{n-1} + k_n,\ k_n \equiv \min\left(\beta_n, \frac{1 - A_{n-1}}{\mathring{n} - n + 1}\right),\quad \Lambda_n = \frac{A_n}{\beta_n + A_{n-1}} \\ A_{nD} &= A_{(n-1)D} + k_{nD},\ k_{nD} \equiv \min\left(\beta_{nD}, \frac{1 - A_{(n-1)D}}{\mathring{n} - n + 1}\right),\quad \Lambda_{nD} = \frac{A_{nD}}{\beta_n + A_{(n-1)D}} \end{aligned} \tag{6.59}$$

It may happen that either $A_{\mathring{n}} < 1$ or $A_{\mathring{n}D} < 1$. Then, $\mathring{n}$ must be increased by 1. The necessary number of such extensions for reaching the value $\max(A_{\mathring{n}}, A_{\mathring{n}D})$ is always finite.

Proof: It is sufficient to inspect the exponential-factor case as the Dirichlet case is identical.

Let us consider a generic step n with A_{n-1} and β_n given. We would like to have $A_{\mathring{n}} = 1$ after a linear growth. With $A_n = A_{n-1} + k_n$, we require $1 = A_{n-1} + \sum_{\tilde{n}=n}^{\mathring{n}} k_{\tilde{n}}$. Hoping for constant $k_{\tilde{n}}$ for $\tilde{n} \geq n$, we get

$$k_n = \frac{1 - A_{n-1}}{\mathring{n} - n + 1} \text{ and } \Lambda_n = \frac{k_n + A_{n-1}}{\beta_n + A_{n-1}}.$$

This is a legal step iff $\Lambda_n \leq 1 \Leftrightarrow k_n \leq \beta_n$. When this condition is met, we apply the proposed Λ_n. We use $\Lambda_n = 1$ otherwise, which corresponds with $k_n = \beta_n$. This will make slower than the intended linear growth but such a Λ_n will increase β_{n+1} at most to 1. Thus, it can be expected that the linear increase will be possible in the next iteration step.

It may well happen that, for the planned $\mathring{n}$, $A_{\mathring{n}}$ does not reach unity. Then, it has to be increased. Need for this extension will end up in a finite amount of attempts as the increase of A_n in each step is positive and the target value is finite. □

The proved proposition specifies a general flattening algorithm applicable both in iterative learning and in branching. It is presented for factors in the exponential family

$$\begin{aligned} f(d_i|\psi_{ic}, \Theta_{ic}, ic) &= A(\Theta_{ic})\exp\left[\langle B(\Psi_{ic}), C(\Theta_{ic})\rangle\right] \text{ and estimates} \\ f(\Theta_{ic}|d(\mathring{t})) &\propto A^{\nu_{ic;\mathring{t}}}(\Theta_{ic})\exp\left[\left\langle V_{ic;\mathring{t}}, C(\Theta_{ic})\right\rangle\right] \text{ resulting from the conjugate prior pdfs} \\ f(\Theta_{ic}) &\propto A^{\nu_{ic;0}}(\Theta_{ic})\exp\left[\langle V_{ic;0}, C(\Theta_{ic})\rangle\right] \text{ and the flat conjugate alternative} \\ \bar{f}(\Theta_{ic}) &\propto A^{\bar{\nu}_{ic}}(\Theta_{ic})\exp\left[\left\langle \bar{V}_{ic}, C(\Theta_{ic})\right\rangle\right]. \end{aligned}$$

Dirichlet pdfs $Di_\alpha(\bar{\kappa})$, $Di_\alpha(\kappa_0)$ are used as flat alternative pdf and prior pdf, respectively. The adopted approximate estimation preserves this form so that the posterior pdf is also $Di_\alpha(\kappa_{\mathring{t}})$.

Algorithm 6.4.4 (Universal flattening with almost linearly growing flattening rate)
Initial mode

- *Select upper bound $\mathring{n}$ on the number n of iterations and set $n = 1$.*
- *Select the statistics $\bar{\nu}, \bar{V}, \bar{\kappa}$ determining the pre-prior pdf $\bar{f}(\Theta)$.*

Iterative mode

1. *Perform approximate mixture estimation, see Section 6.5.*

2. *Evaluate*

$$\begin{aligned} T &= \begin{cases} \mathring{t} & \textit{if no forgetting is used} \\ \sum_{t=1}^{\mathring{t}} \lambda^t \approx \frac{1}{1-\lambda} & \textit{if forgetting with factor } \lambda < 1 \textit{ is used} \end{cases} \\ \beta_n &= \frac{\sum_{i\in i^*, c\in c^*} \nu_{ic;\mathring{t}}}{\mathring{d}T + \sum_{i\in i^*, c\in c^*} \nu_{ic;0}} \quad \textit{for the "factor" part of the posterior pdf} \\ \beta_{nD} &= \frac{\sum_{c\in c^*} \kappa_{c;\mathring{t}}}{T + \sum_{c\in c^*} \kappa_{c;0}} \quad \textit{for the Dirichlet part of the posterior pdf.} \end{aligned}$$

3. *Compute flattening rates Λ_n, Λ_{nD} and the powers of the likelihood A_n, A_{nD} for the factor as well as for Dirichlet parts.*

$$\begin{aligned} \textit{For } n = 1, \quad & \Lambda_1 \equiv \frac{\sum_{i\in i^*, c\in c^*} (\nu_{ic;0} - \bar{\nu}_{ic})}{\sum_{i\in i^*, c\in c^*} (\nu_{ic;\mathring{t}} - \bar{\nu}_{ic})} \textit{ and set } A_1 \equiv \beta_1 \Lambda_1 \\ & \Lambda_{1D} \equiv \frac{\sum_{c\in c^*} (\kappa_{c;0} - \bar{\kappa}_c)}{\sum_{c\in c^*} (\kappa_{c;\mathring{t}} - \bar{\kappa}_c)} \quad \textit{and set } A_{1D} \equiv \beta_{1D}\Lambda_{1D}. \\ \textit{For } n > 1, \quad & k_n \equiv \min\left(\beta_n, \frac{1 - A_{n-1}}{\mathring{n} - n + 1}\right), \ A_n = A_{n-1} + k_n, \ \Lambda_n = \frac{A_n}{\beta_n + A_{n-1}} \\ & k_{nD} \equiv \min\left(\beta_{nD}, \frac{1 - A_{(n-1)D}}{\mathring{n} - n + 1}\right), \ A_{nD} = A_{(n-1)D} + k_n, \ \Lambda_{nD} = \frac{A_{nD}}{\beta_{nD} + A_{(n-1)D}}. \end{aligned}$$

4. *Apply flattening as follows, $i = 1, \ldots, \mathring{d}$, $c \in c^*$.*

$$\begin{aligned} \textit{For } n = 1, \quad & V_{ic(n+1);0} = \Lambda_n V_{icn;\mathring{t}} + (1 - \Lambda_n)\bar{V}_{ic}, \ \nu_{ic(n+1);0} = \Lambda_n \nu_{icn;\mathring{t}} + (1 - \Lambda_n)\bar{\nu}_{ic} \\ & \kappa_{c(n+1);0} = \Lambda_{nD}\kappa_{cn;\mathring{t}} + (1 - \Lambda_{nD})\bar{\kappa}_c \\ \textit{For } n > 1, \quad & V_{ic(n+1);0} = \Lambda_n V_{icn;\mathring{t}} + (1 - \Lambda_n) V_{ic;0}, \ \nu_{ic(n+1);0} = \Lambda_n \nu_{icn;\mathring{t}} + (1 - \Lambda_n)\nu_{ic;0} \\ & \kappa_{c(n+1);0} = \Lambda_{nD}\kappa_{cn;\mathring{t}} + (1 - \Lambda_{nD})\kappa_{c;0}. \end{aligned}$$

5. *Stop if $\mathring{n} = 1$ and take the result as the newly generated prior pdf.*

6. *Increase $n = n + 1$.*

7. *Stop if $n > \mathring{n} > 1$ and $\max(A_{\mathring{n}}, A_{\mathring{n}D}) = 1$ and take the result as the newly generated prior pdf.*

8. *Go to the beginning of* Iterative mode *if $n \le \mathring{n}$.*

9. *Increase $\mathring{n} = \mathring{n} + 1$ and go to the beginning of* Iterative mode *if $n > \mathring{n}$ and $\max(A_{\mathring{n}}, A_{\mathring{n}D}) < 1$.*

Remark(s) 6.4.4

1. *It is possible to take the result of the first estimation started from a flat pre-prior pdf and followed by flattening as the prior pdf if there is no justified alternative option.*

2. *The powers $A_{\mathring{n}}$, $A_{\mathring{n}D}$ have to be simultaneously checked for reaching unity. Generally, it may not be feasible. We have selected conservative stopping leading to a flatter final pdf. It corresponds to the experience that over-confidence is more dangerous under-confidence.*

3. *Use of this flattening should allow us to use forgetting extensively as the presented algorithm avoids the check on values of the forgetting factor discussed in Remarks 6.4.3.*

Problem 6.4.2 (Improvements of flattening strategies) *Experiments indicate superiority of forgetting with linear growth. Seemingly more realistic "almost linear growth" provides poorer results. It indicates that the understanding of the problem is still limited and there is a space for a further improvements.*

6.4.4 Geometric mean as branching mapping

This section starts specialization of the general branch and bound technique, Section 6.1.3, to the problem summarized in Agreement 6.4.2.

The inspected estimates of the prior pdf $\hat{f}(\Theta)$ and the adopted approximate estimator, Section 6.5, determine the joint pdf of data and mixture parameters $X \equiv \hat{f}(\cdot,\cdot) \equiv \hat{f}(d(\mathring{t}),\Theta)$. This pdf is the argument of the maximized functional (6.40)

$$F(X) \equiv {}^{\lfloor h}\mathcal{L}\left(\hat{f}(\cdot,\cdot), d(\mathring{t})\right) \equiv \int \hat{f}(d(\mathring{t}),\Theta)\, d\Theta \equiv \hat{f}(d(\mathring{t})).$$

While searching for maximizing argument we always preserve the best argument found till the considered step, cf. Proposition 6.1.3. Thus, for describing the branching mapping, it is necessary to describe only how novel alternative guesses $\{\hat{f}(\Theta)\}$ of the best prior pdf are generated.

Here, we inspect the case when the set of candidates contains just a pair of pdfs $\hat{f}_1(\Theta)$, $\hat{f}_2(\Theta)$ and a new singleton is generated. These guesses of the prior pdf define the approximate posterior pdfs through an approximate estimation. We want to find the best point estimate of the posterior pdf. It is unknown because of the lack of complete knowledge about the adequate prior pdf. Formally, we are facing exactly the same situation as discussed in Section 6.3.3. Similarly to Agreement 6.3.2, we introduce the unknown probability β_1 that the unknown prior pdf $f(\Theta)$ belongs to the set f_1^* containing pdfs that are closer to the guess $\hat{f}_1(\Theta)$ than to the guess $\hat{f}_2(\Theta)$. Adopting the approximation $f(\Theta) \approx \hat{f}_1(\Theta)$ on f_1^* and $f(\Theta) \approx \hat{f}_2(\Theta)$ on the complement of f_1^*, we get the (approximate) minimizing argument $\hat{f}(\Theta|d(\mathring{t}))$ of the expected KL distance $\mathcal{E}\left[\mathcal{D}\left(f||\hat{f}\right)\right]$. It is a hot candidate for the value of branching operator. This motivates the definition of the *geometric branching mapping*

$$\begin{aligned}
\mathcal{A} \quad &: \quad \left\{\hat{f}_1(d(\mathring{t})), \hat{f}_2(d(\mathring{t})), \hat{f}_1(\Theta|d(\mathring{t})),\ \hat{f}_2(\Theta|d(\mathring{t}))\right\} \to \hat{f}(\Theta|d(\mathring{t})) \propto [\hat{f}_1\ (\Theta|d(\mathring{t}))]^{\lambda}[\hat{f}_2\ (\Theta|d(\mathring{t}))]^{1-\lambda} \\
\lambda \quad &\equiv \quad \beta_{1|d(\mathring{t})} = \frac{\hat{f}_1(d(\mathring{t}))}{\hat{f}_1(d(\mathring{t})) + \hat{f}_2(d(\mathring{t}))}.
\end{aligned} \tag{6.60}$$

The used values of v-likelihood $\hat{f}_i(d(\mathring{t})) \equiv \int \hat{f}_i(d(\mathring{t})|\Theta)\hat{f}_i(\Theta)\, d\Theta$, $i = 1,2$, are obtained as by-products of the quasi-Bayes algorithm, see Section 6.1.1, formula (6.7), or have to be approximated for a fixed predictor as discussed in Section 6.1.2.

Remark(s) 6.4.5

1. *We may use the geometric branching (almost) universally in conjunction with various initial guess of alternatives and various additional branching mappings $\mathcal{A}$. In this way, we get various versions of the general branch and bound Algorithm 6.1.1. These variants are discussed in subsequent sections.*

2. *It is straightforward to specify a geometric branching that projects more than two variants into a single pdf. Simply,*

$$\hat{f}(d(\mathring{t}),\Theta) \propto \prod_{i=1}^{\mathring{i}} [\hat{f}_i(d(\mathring{t}),\Theta)]^{\lambda_i}, \quad \lambda_i \propto \int \hat{f}_i(d(\mathring{t}),\Theta)\, d\Theta,\ i = 1,2,\ldots,\mathring{i}.$$

It does not seem that it makes sense to deal with this extension in our context.

Problem 6.4.3 (Likelihood assigned to the geometric mean) *It is not sure that the found geometric mean gives a better results. Generally, it must be checked by repeating the estimation after flattening. The change in the v-likelihood can be, however, predicted without repeating the estimation. It is implied by the Bayes rule (2.8)*

$$\hat{f}_i(\Theta|d(\mathring{t})) \quad = \quad \frac{\mathcal{L}(\Theta,d(\mathring{t}))\hat{f}_i(\Theta)}{\int \mathcal{L}(\Theta,d(\mathring{t}))\hat{f}_i(\Theta)\, d\Theta} = \frac{\mathcal{L}(\Theta,d(\mathring{t}))f_i(\Theta)}{\hat{f}_i(d(\mathring{t}))},\ i = 1,2 \quad \underbrace{\Rightarrow}_{(6.60)}$$

$$\hat{f}_3(\Theta|d(\mathring{t})) = \frac{[\hat{f}_1(\Theta|d(\mathring{t}))]^{\lambda}[\hat{f}_2(\Theta|d(\mathring{t}))]^{1-\lambda}}{\int[\hat{f}_1(\Theta|d(\mathring{t}))]^{\lambda}[\hat{f}_2(\Theta|d(\mathring{t}))]^{1-\lambda}\, d\Theta} \underbrace{\Rightarrow}_{(2.8)}$$

$$\hat{f}_3(\Theta|d(\mathring{t})) = \frac{\mathcal{L}(\Theta, d(\mathring{t}))\hat{f}_1^{\lambda}(\Theta)\hat{f}_2^{1-\lambda}(\Theta)}{\hat{f}_3(d(\mathring{t}))\int \hat{f}_1^{\lambda}(\Theta)\hat{f}_2^{1-\lambda}(\Theta)\, d\Theta} \Rightarrow$$

$$\hat{f}_3(d(\mathring{t})) = [\hat{f}_1(d(\mathring{t}))]^{\lambda}[\hat{f}_2(d(\mathring{t}))]^{1-\lambda}\frac{\int\left[\hat{f}_1(\Theta|d(\mathring{t}))\right]^{\lambda}\left[\hat{f}_2(\Theta|d(\mathring{t}))\right]^{1-\lambda} d\Theta}{\int\left[\hat{f}_1(\Theta)\right]^{\lambda}\left[\hat{f}_2(\Theta)\right]^{1-\lambda} d\Theta}.$$

The above formula is approximate one as the approximately evaluated likelihood function depends on the chosen prior pdf. The derived relationship has not been used yet in this case but its analogy is widely exploited for merging of components and factors, see Section 6.6.4. It is worth of trying as it would enhanced potential behind the geometric-mean mapping.

6.4.5 Random branching of statistics

The deterministic algorithms are prone to find just a local optimum. It is known that the global optimization calls for a random search [50]. Computational complexity prevents us to use the safe, completely random search. Thus, we should use a combination of deterministic and random searches. Here, we discuss possible random steps.

According to Agreement 6.1.1, all considered pdfs on unknown parameters of the mixture have the form (6.3). In learning, we have to consider the cases when the whole data history $d(\mathring{t})$ reduces to a finite-dimensional (almost) sufficient statistics, say $\{\mathcal{V}_{ic}\}$. It means that estimates are searched in the form $\hat{f}(\Theta_{ic}|d(\mathring{t})) = \hat{f}(\Theta_{ic}|\mathcal{V}_{ic})$, $i \in \{1, \ldots, \mathring{d}\}$, $c \in c^*$. The functional form of the right-hand sides as well as the set $\{\mathcal{V}_{ic}^*\}$ of possible values of $\{\mathcal{V}_{ic}\}$ are known.

Knowing this, we can generate random samples $\tilde{\mathcal{V}}_{ic}$ in $\mathcal{V}_{ic}^*$. It gives us a new guess $\tilde{f}(\Theta_{ic}|d(\mathring{t})) \equiv \hat{f}(\Theta_{ic}|\tilde{\mathcal{V}}_{ic})$. By flattening, see Proposition 6.4.1, we get a new guess of the prior pdf $\hat{f}_{new}(\Theta_{ic})$.

Remark(s) 6.4.6

1. *The random generating should be applied to selected factors only. It decreases computational load and, moreover, preserves those estimates that are satisfactory. Typically, the candidate factors are those that provide predictors similar to predictors constructed from the prior pdf only. The similarity may be measured by entropy rate of this predictor pair.*

2. *The random sample $\tilde{\mathcal{V}}_{ic}$ may or may not be related to the given $\mathcal{V}_{ic}$. The latter option allows us a less restricted search and increases chance for reaching the best possible result. Computationally it may be prohibitive.*

3. *Samples $\tilde{\mathcal{V}}_{ic}$ around $\mathcal{V}_{ic}$ should be searched for, if worries of the previous item apply. The Bayes rule implies that $f(\mathcal{V}_{ic}) = f(d(\mathring{t})) = \frac{f(d(\mathring{t})|\Theta_{icj})f(\Theta_{icj})}{f(\Theta_{icj}|d(\mathring{t}))}$ for any $\{\Theta_{icj}\}_{j\in j^*} \in \Theta^*$. Thus, we know functional form of $f(\mathcal{V}_{ic})$ and we are able to evaluate this pdf for any values $\mathcal{V}_{ic}$. Thus, we can get a clue how this pdf looks like and to use it potentially for generating of highly probable samples of the statistics.*

6.4.6 Prior-posterior branching

The posterior pdf flattened back to the space of potential prior pdfs provides an alternative to the original prior pdf. Thus, $\mathcal{A}_{\Lambda}$ (6.45) can be used as a promising branching mapping. Flattening combined with geometric branching, see (6.60), we call *prior-posterior branching*. It gives the following iterative Bayesian learning algorithm.

Algorithm 6.4.5 (Prior-posterior branching)
Initial mode

- *Select an upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 1$.
- *Select a sufficiently rich structure of the mixture, Agreement 5.2.1.*
- *Select the pre-prior* $\bar{f}(\Theta) =$ *a flat prior pdf used for flattening.*
- *Select the prior pdf* $\hat{f}_{1n}(\Theta) \equiv f(\Theta)$ *(not necessarily equal to* $\bar{f}(\Theta)$*).*
- *Compute the approximate posterior pdf* $\tilde{f}_{1n}(\Theta|d(\mathring{t})) \propto f(d(\mathring{t})|\Theta))\hat{f}_{1n}(\Theta)$ *using an approximate Bayesian estimation, see Section 6.5.*
- *Evaluate the v-likelihood* l_{1n} *resulting from the use of* $\hat{f}_{1n}(\Theta)$.
- *Apply the flattening operation to* $\tilde{f}_{1n}(\Theta|d(\mathring{t}))$ *according to Algorithms 6.4.2 (on Dirichlet marginal) and 6.4.3 (on pdfs describing factors). Denote the resulting pdf* $\hat{f}_{2n}(\Theta)$.
- *Compute the approximate posterior pdf* $\tilde{f}_{2n}(\Theta|d(\mathring{t})) \propto f(d(\mathring{t})|\Theta))\hat{f}_{2n}(\Theta)$ *using an approximate Bayesian estimation, see Section 6.5.*
- *Evaluate the v-likelihood* l_{2n} *resulting from the use of* $\hat{f}_{2n}(\Theta)$.
- *Set* $\bar{l}_n = \max(l_{1n}, l_{2n})$.

Iterative mode

1. *Apply geometric branching to the pair* $\tilde{f}_{1n}(\Theta|d(\mathring{t}))$, $\tilde{f}_{2n}(\Theta|d(\mathring{t}))$ *with the v-likelihood* l_{1n}, l_{2n}, *respectively. For* $\lambda \equiv \frac{l_{1n}}{l_{1n}+l_{2n}}$, *it gives*
$$\tilde{f}_{3n}(\Theta|d(\mathring{t})) \propto \left[\tilde{f}_{1n}(\Theta|d(\mathring{t}))\right]^{\lambda} \left[\tilde{f}_{2n}(\Theta|d(\mathring{t}))\right]^{1-\lambda}.$$
2. *Apply the flattening operation on* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$ *according to Algorithms 6.4.2 (on Dirichlet marginal) and 6.4.3 (on pdfs describing factors). Denote the resulting pdf* $\hat{f}_{3n}(\Theta)$.
3. *Compute the approximate posterior pdf* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$ *resulting from the use of* $\hat{f}_{3n}(\Theta)$ *as the prior pdf. Evaluate the v-likelihood* l_{3n} *related to the use of* $\hat{f}_{3n}(\Theta)$ *as the prior pdf.*
4. *Choose among* $\tilde{f}_{in}(\Theta|d(\mathring{t}))$, $i \in \{1, 2, 3\}$ *the pair with the highest v-likelihood* l_{in}. *Call them* $\tilde{f}_{1(n+1)}(\Theta|d(\mathring{t}))$, $\tilde{f}_{2(n+1)}(\Theta|d(\mathring{t}))$ *with the v-likelihood* $l_{1(n+1)}$, $l_{2(n+1)}$.
5. *Go to the beginning of* Iterative mode *with* $n = n + 1$ *if* $\bar{l}_{n+1} \equiv \max(l_{1(n+1)}, l_{2(n+1)}) > \bar{l}_n$, *or if* $\bar{l}_{n+1}, \bar{l}_n$ *are the same according to Proposition 6.1.2, and if* $n < \mathring{n}$.
6. *Stop and select among* $\hat{f}_{in}(\Theta)$, $i = 1, 2$, *the pdf leading to the higher value of* l_{in} *as the prior pdf constructed.*

Remark(s) 6.4.7

1. *No improvement can be expected when the values of* λ *stabilize around 0.5. This observation may serve as an additional stopping rule.*
2. *The algorithm is applicable out of the considered context of the mixture estimation.*
3. *The structure of the estimated mixture does not change during iterations. Thus, it has to be sufficiently reflected even in the prior pdf* $f(\Theta)$. *Techniques allowing changes of the structure are described later on.*

Problem 6.4.4 (Controlled prior-posterior branching) *Prediction of the v-likelihood after making geometric mean, see Problem 6.4.3, could be used for controlling the branching algorithm. The comparison of the predicted and evaluated the v-likelihood values indicates how much the likelihood varies due to the approximate estimation used.*

6.4.7 Branching by forgetting

Each iteration of the Bayes rule described in previous Section is performed over all data available $d(\mathring{t})$. It might be waste of the precious computational resources. Moreover, it cannot be used in on line mode. The technique described here helps to avoid these drawbacks. Essentially, the considered alternatives are generated through parallel quasi-Bayes estimations, Algorithm 6.5.2, differing in forgetting factors used.

For a given $\hat{f}(\Theta|d(t))$, two approximate estimations run in parallel. One with the unit forgetting and the other one with a forgetting $\lambda << 1$. Their predictive abilities are evaluated through their v-likelihood. Their geometric mean coincides with one of the compared pdfs if the absolute difference of their v-log-likelihood values is high enough. At this moment, the poorer variant should be reset to the better alternative. This description is formalized in the following algorithm.

Algorithm 6.4.6 (On line branching with forgetting on components)
Initial mode

- *Select a sufficiently rich structure of the mixture.*
- *Select a constant* $\rho \approx 3 - 5$ *defining the significant difference of* v*-log-likelihood values.*
- *Set the record counter* $t = 0$ *and select the prior pdf* $\hat{f}(\Theta|d(t)) \equiv f(\Theta)$.
- *Choose a fixed, relatively small forgetting factor* $\lambda < 1$.
- *Select a fixed pre-prior alternative pdf used in the stabilized forgetting, see Proposition 3.1.1. The alternative is either flat pre-prior pdf or the chosen prior pdf.*

Data processing mode

1. *Set* $\hat{f}_1(\Theta|d(t)) = \hat{f}_\lambda(\Theta|d(t)) = \hat{f}(\Theta|d(t))$.
2. *Initialize* v*-log-likelihood values assigned to considered alternatives* $l_{1;t} = 0$, $l_{\lambda;t} = 0$.
3. *Collect new data* d_{t+1}.
4. *Update* $\hat{f}_1(\Theta|d(t))$ *to* $\hat{f}_1(\Theta|d(t+1))$ *using approximate estimation, Section 6.5, with the forgetting factor 1.*
5. *Re-compute the* v*-log-likelihood* $l_{1;t}$ *to* $l_{1;t+1}$ *by adding the logarithm of the mixture prediction* $\ln(f(d(t+1)|d(t)))$ *obtained for the "prior" pdf* $\hat{f}_1(\Theta|d(t))$.
6. *Update* $\hat{f}_\lambda(\Theta|d(t))$ *to* $\hat{f}_\lambda(\Theta|d(t+1))$ *using approximate estimation with the forgetting factor* λ *and the chosen alternative pdf.*
7. *Re-compute the* v*-log-likelihood* $l_{\lambda;t}$ *to* $l_{\lambda;t+1}$ *by adding logarithm of the mixture prediction* $\ln(f(d(t+1)|d(t)))$ *obtained for the "prior" pdf* $\hat{f}_\lambda(\Theta|d(t))$.
8. *Go to the step 3 with* $t = t+1$ *if* $|l_{1;t+1} - l_{\lambda;t+1}| < \rho$.
9. *Set* $\hat{f}(\Theta|d(t+1)) = \hat{f}_1(\Theta|d(t+1))$ *if* $l_{1;t+1} > l_{\lambda;t}$, *otherwise set* $\hat{f}(\Theta|d(t+1)) = \hat{f}_\lambda(\Theta|d(t+1))$.
10. *Set* $t = t+1$ *and go to the beginning of* Data processing mode *if* $t \leq \mathring{t}$.
11. *Take* $\hat{f}(\Theta|d(\mathring{t}))$ *as the final estimate of the posterior pdf.*

Remark(s) 6.4.8

1. *The decision on the appropriate model, based on the difference of the involved* v*-likelihood, seems to be sensitive to the choice of* ρ. *An insufficient reliability of the posterior pdfs seems to be a real reason of this behavior. This hints the appropriate remedy: Decide on the model quality after checking that the probability of the branch with no forgetting stabilizes in the time course.*

2. *Speeding up of the learning is the main expectation connected with this algorithm. The model with no forgetting is expected to be winner in a long run. Thus, the estimation with forgetting can be switched off when the estimation without forgetting is better for majority of time. The rule for switching off the estimation with forgetting might look like as follows.*

   Switch off the estimation with forgetting if the estimation without it is better for a sufficiently large portion of time.

3. *There is a danger of a numerical breakdown when the update with forgetting wins for a longer time. This case indicates either a very poor start or a significantly wrong model structure. The use of a proper alternative in forgetting removes this danger.*

4. *The algorithm can be extended to the factor level. This extension will be added in the future. This extension looks quite promising and worth of trying.*

5. *The estimation with no forgetting should be always included as we search for a time invariant model of the o-system.*

6. *The technique can be directly combined with prior-posterior branching.*

7. *It will be necessary to get experience with choice of the fixed forgetting factor λ. At present, the option $\lambda \approx 0.6$ seems to be satisfactory.*

8. *The necessary condition needed for standard flattening $\sum_{i\in i^*,c\in c^*} \nu_{ic;\mathring{t}} > \sum_{i\in i^*,c\in c^*} \nu_{ic;0}$ may be violated when forgetting $\lambda < 1$ wins to often and non-informative alternative pdf is used. Use of the stabilized forgetting with a simplified alternative that preserves the degrees of freedom is the recommended way how to avoid the problem.*

6.4.8 Branching by factor splitting

The design of the branching mapping (6.17) presented in this section suits to cases when there is no clue on mixture structure, i.e. structure of factors, presence of common factors and the number of components. The remaining, hopefully less significant, estimation of the component structure is unsolved yet.

Basic idea of the branching by factor splitting is simple. Let us assume that the mixture structure, Agreement 5.2.1, and the estimate of the prior pdf $\hat{f}_n(\Theta)$ are selected in an n-th iteration. The approximate estimation, Section 6.5, gives the posterior pdf $\tilde{f}_n(\Theta|d(\mathring{t})) \propto f(d(\mathring{t})|\Theta)\hat{f}_n(\Theta)$.

During approximate estimation, Section 6.5, we can check whether a particular factor covers two modes of the pdf predicting $d_{ic;t}$. We split each two-modes covering factor into a new pair that preserves some properties of the original factor but has a chance to fit the processed data better.

Structure of factors is estimated and possibly modified during splitting. We get a new, substantially larger, amount of components that are re-estimated after flattening. Then, similar components are merged or spurious ones cancelled. Consequently a simpler mixture is obtained. The whole process is repeated while the v-likelihood increases.

For reference purposes, we list steps that have to be elaborated in detail.

Algorithm 6.4.7 (Steps of branching by splitting)

1. *Selection of the very initial mixture to be split.*
2. *Selection of factors for splitting.*
3. *Splitting of factors.*
4. *Estimation of the factor structure, see subsection 6.6.1.*
5. *Construction of a new mixture.*
6. *Merging of components, see subsection 6.6.4.*
7. *Cancelling of components, see subsection 6.6.4.*

The steps without reference are discussed in this section. The structure estimation has a wider use and its parts are discussed in the referred subsections of Section 6.6.

Selection of the initial mixture: step 1 in Algorithm 6.4.7

The very initial mixture, corresponding to the selection of the initial set of alternatives in branch-and-bound Algorithm 6.1.1, has to selected first. It should

- give a chance to find the objective model,
- exclude part of the data space that contains (almost) no data.

The first condition requires selection of sufficiently rich structures of factors involved. For reference purposes, we formulate it formally.

Requirement 6.4.1 (Over-parameterized factors)
The regression vectors $\psi_{ic;t}$ *of parameterized factors* $f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1), \Theta_{ic}, c) \equiv f(d_{i;t}|\psi_{ic;t}, \Theta_{ic}, c)$ *include regression vectors* $^{\llcorner o}\psi_{i;t}$ *of the objective pdf* $^{\llcorner o}f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1)) \equiv {}^{\llcorner o}f(d_{i;t}|\, ^{\llcorner o}\psi_{i;t})$*, cf. Section 2.3.1.*

This requirement is relatively simply fulfilled due to the slow dynamics of the inspected o-system, cf. Section 5.2. It must, however, be respected when reducing the model structure, see Section 6.6.

To meet the second wish on the very initial mixture, we used the single-component mixture. It safely excludes the space out of the union of support of the objective components. It cannot exclude emptiness of the inter-component space. This causes often bad behavior of the search. Typically, the attempts to improve the one-component mixture fail and the initialization stops prematurely. A few splitting performed without the check of the v-likelihood brings practical remedy of the problem. The depth $\mathring{m}$ of this initial splitting is usually restricted by the problem complexity so that its choice is not difficult. This uncontrolled splitting relies on our ability to merge similar components, see Section 6.6.4.

Algorithm 6.4.8 (Selection of the very initial mixture)
Initial mode

- *Select variables in* d_t *to be modelled.*
- *Select a structure of the richest regression vector.*
- *Select an initial-splitting depth* $\mathring{m} > 1$ *and set the depth counter* $m = 0$.
- *Select a (flat) prior pdf* $f(\Theta)$ *on the single component mixture.*
- *Select an initial value of the statistics* κ_0 *describing Dirichlet pdf (10.2). It serves for flattening according to Algorithm 6.4.2.*

Evaluation mode

1. *Increase* $m = m + 1$.
2. *Perform (approximate) estimation of the mixture, see Section 6.5.*
3. *Split factors as described in subsequent sections.*
4. *Apply flattening operation, Algorithms 6.4.2, 6.4.3, so that a new initial mixture estimate is obtained.*
5. *Go to the beginning of* Evaluation mode *if* $m < \mathring{m}$. *Otherwise take the resulting mixture as the very initial one for further evaluations.*

Remark(s) 6.4.9

Other choices, like random starts, are possible and should be considered in specific cases. A random start, however, does not guarantee that we shall not stay in a weakly populated part of the data space. Moreover, available generators of this type care mostly about initial positions only. They do not provide a complete description of the prior pdf.

Hierarchical selection of split factors and their split: steps 2 and 3 in Algorithm 6.4.7

The *hierarchical splitting* described here solves both referred steps. It relies on our ability to estimate three factors instead of a single one.

During the approximate estimation, Section 6.5, the i-th factor within the c-th component is assigned a weight $w_{ic;t} \in [0,1]$ that expresses the expectation that the data item $d_{ic;t}$ is generated by this factor. Then, the modified parameterized factor

$$f_w(d_{ic;t}|d_{i+1\_\mathring{d};t}, d(t-1)) \equiv f_w(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) \propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic})]^{w_{ic;t}}$$

is used in the "ordinary" Bayes rule for updating of parameter estimates of this factor. Thus, for estimation purposes, we can inspect this factor without taking into account other ones. It allows us to simplify notation and drop temporarily the subscripts w, c.

We want to check whether the discussed factor explains data that should be modelled by more factors while assuming that Requirement 6.4.1 is met. We formulate hypothesis H_0 that the objective pdf has two components

$$^{\llcorner o}f(d_t|\psi_t) = \beta f(d_t|\psi_t, \Theta_1) + (1-\beta) f(d_t|\psi_t, \Theta_2),\ \beta \in (0,1),\ \text{where} \tag{6.61}$$

Θ_1, Θ_2 have the same structure as the parameter Θ in the split factor.

The alternative hypothesis H_1 assumes that $^{\llcorner o}f(d_t|\psi_t) = f(d_t|\psi_t,\ ^{\llcorner o}\Theta)$, i.e. denies the need for splitting.

If we can afford it, we estimate parameters of the mixture (6.61) together with the factor in question in order to decide on the need to split. With $f(H_0) = f(H_1)$, modelling no prejudice, the Bayes rule gives

$$Probability(\text{split is needed}|d(\mathring{t})) \equiv f(H_0|d(\mathring{t})) = \frac{f(d(\mathring{t})|H_0)}{f(d(\mathring{t})|H_0) + f(d(\mathring{t})|H_1)}. \tag{6.62}$$

The v-likelihood values $f(d(\mathring{t})|H_0)$, $f(d(\mathring{t})|H_1)$ are obtained as byproduct of the approximate estimation, Section 6.5.

The factor is split if the probability (6.62) is high enough. The estimated factors of the mixture (6.61) are natural candidates for its replacement.

For reference purposes, let us write down the corresponding algorithm.

Algorithm 6.4.9 (Hierarchical split of factors)
Initial mode

- *Construct the very initial estimate $f(\Theta)$ of the mixture, Algorithm 6.4.8.*
- *Select significance level $\bar{P} \in (0,1)$, $\bar{P} \approx 1$ controlling the need to split.*
- *Assign to each factor $i, c,$, $i = 1, \ldots, \mathring{d}$, $c \in c^*$, a prior pdf $f(\beta_{ic}, \Theta_{1ic}, \Theta_{2ic})$ on parameters of the mixture (6.61). Indices ic stress the "membership" of this mixture to the factor ic.*

Sequential mode, *running for $t = 1, 2, \ldots,$*

1. *Acquire data d_t and complement the data vectors $\Psi_{ic;t} = [d_{ic;t}, \psi'_{ic;t}]'$, $i = 1, \ldots, \mathring{d}$, $c \in c^*$.*
2. *Perform single step of an approximate estimation, i.e. update $f(\Theta|d(t-1)) \rightarrow f(\Theta|d(t))$. It generates weights $w_{ic;t}$ measuring the degree with which the data d_t are assigned for respective factors.*
   Update at the same time the respective values of the v-likelihood $l_{ic;t|H_1}$
3. *Weight new data by respective $w_{ic;t}$, i.e. in the exponential family (3.5) replace unit power of $A(\Theta_{ic})$ by $w_{ic;t}$ and $B(\Psi_{ic;t})$ by $w_{ic;t}B(\Psi_{ic;t}) \equiv B(\Psi_{wic;t})$. Update estimates of parameters of the models (6.61)*
   $$f(\beta_{ic}, \Theta_{1ic}, \Theta_{2ic}|d(t-1)) \rightarrow f(\beta_{ic}, \Theta_{1ic}, \Theta_{2ic}|d(t)),\quad \textit{using } \Psi_{wic;t}.$$
   Update also values of the v-likelihood $l_{ic;t|H_0}$ using the predictors $f(d_{wic;t}|\psi_{wic;t}, d(t-1))$ constructed from the estimated two-component mixture (6.61).

4. Go to the beginning of Sequential mode *while* $t \leq \mathring{t}$.

Selection mode *running for* $i = 1, \ldots, \mathring{d},\ c \in c^*$
Split $f(\Theta_{ic}|d(\mathring{t})) \rightarrow \left(f(\Theta_{1ic}|d(\mathring{t})), f(\Theta_{2ic}|d(\mathring{t}))\right)$ *if*

$$\frac{l_{ic;\mathring{t}|H_0}}{l_{ic;\mathring{t}|H_0} + l_{ic;\mathring{t}|H_1}} \geq \bar{P}.$$

Heuristic choice of split factors: an alternative for the step 2 in Algorithm 6.4.7

The hierarchically-based splitting depends heavily on our ability to face the computational demands caused by the need to estimate 3 factors instead of a single one. In this and subsequent paragraph, less demanding ways are discussed.

Simplification is reached by leaving out the full estimation of sub-factors hidden in the factor to be split. It is done by

- a heuristic step, for instance, by selecting all or the most flat factors in the most populated component,
- a hypothesis testing based on some statistics collected during estimation,
- exploiting a reduced version of the hierarchical split, see Chapter 8.

Other variants were also considered. Generally, experience is rather mixed but the last item seems to be mostly preferable.

Optimization-based factor splitting: an alternative of the step 3 in Algorithm 6.4.7

Simplified splitting of a selected factor is discussed here.

The first variant takes new factors as flattened or sharpened versions of the factor in question.

Proposition 6.4.7 (Flattened/sharpened factors) *Let pdfs* $f, \bar{f}$ *have a common support* Θ^*. *Let the pdf* $\hat{f}$ *minimize the KL distance* $\mathcal{D}\left(\hat{f}||f\right)$ *and have the pre-specified KL distance*

$$\mathcal{D}\left(\hat{f}||\bar{f}\right) = \omega \mathcal{D}\left(f||\bar{f}\right),\ \omega \neq 1. \text{ Then, it has the form } \hat{f} \propto f^{\lambda}\bar{f}^{1-\lambda}. \tag{6.63}$$

The scalar λ *chosen so that the restriction in (6.63) is met.*

Proof: We minimize convex functional and minimizing argument of the Lagrangian is searched for. It coincides, however, with the functional (6.43) with q in the role of the Lagrangian multiplier. It gives flattened version if $q > 0 \Leftrightarrow \lambda < 1$ and sharpened version if $q < 0 \Leftrightarrow \lambda > 1$. □

Flatten/sharpened version of a factor has a limited use as it preserves the original-factor mode. Moreover, the choice of the optional value ω is unsolved.

More promising is the version that enforces explicitly a shift of the original-factor mode. The mode is characterized through the expected value of a function $g(\Theta)$ of the estimated parameters Θ.

Proposition 6.4.8 (Shifted factor) *Let pdfs* $\hat{f}, f$ *have a common support* Θ^* *and let* g *be a vector function defined on* Θ^*. *Let the pdf* $\hat{f}$ *minimize the KL distance* $\mathcal{D}\left(\hat{f}||f\right)$ *and have the prescribed norm of the shift in expectation of the function* $g(\Theta)$

$$\omega = \left\langle \int g(\Theta)(\hat{f}(\Theta) - f(\Theta))\, d\Theta, \int g(\Theta)(\hat{f}(\Theta) - f(\Theta))\, d\Theta \right\rangle,\ \omega > 0. \text{ Then, it has the form}$$
$$\hat{f}(\Theta) \propto f(\Theta)\exp[s\langle \mu, g(\Theta)\rangle],\ s \in \{-1, 1\}, \tag{6.64}$$

determined by the constant vector μ. *The finite-dimensional vector* μ *is compatible with the scalar product* $\langle \cdot, \cdot \rangle$ *and solution of the considered task is in the set of functions (6.64) "indexed" by* μ.

Proof: The Lagrangian of this optimization task is

$$\mathcal{J} \equiv \mathcal{D}\left(\hat{f}||f\right) + \nu \int \hat{f}(\Theta)\, d\Theta + \rho \left\langle \int g(\Theta)\left(\hat{f}(\Theta) - f(\Theta)\right) d\Theta, \int g(\Theta)\left(\hat{f}(\Theta) - f(\Theta)\right) d\Theta \right\rangle.$$

It is determined by scalars ν, ρ that have to be chosen so that restrictions, normalization and the restriction on the shift norm, are met. Let us denote $\mu \equiv \rho \int g(\Theta)(\hat{f}(\Theta) - f(\Theta))\, d\Theta$. Then, the Lagrangian can be written in the form

$$\mathcal{J} = \int \hat{f}(\Theta) \ln\left(\frac{\hat{f}(\Theta)}{\frac{f(\Theta)\exp[-\mu' g(\Theta)]}{\int f(\Theta)\exp[-\mu' g(\Theta)]\, d\Theta}}\right) d\Theta - \ln\left[\int f(\Theta)\exp[-\mu' g(\Theta)]\, d\Theta\right].$$

Basic properties of the KL distance imply the claimed form of the minimizer. The freedom in the choice of s is implied by the even nature of the restriction in (6.64). □

Remark(s) 6.4.10

1. *The proposed shifting may fail if the over-parameterized factor is split. The shift tends to directions, where the split factor is less concentrated. In the over-parameterized case, such a shift brings nothing positive. Thus, it is necessary to perform* structure estimation *on the factor selected for splitting and get rid off superfluous parameters.*
2. *Restrictions in Propositions 6.4.7, 6.4.8 can be used simultaneously giving the candidate pdfs in the predictable form*
$$\hat{f}(\Theta) \propto f^{\lambda}(\Theta)\bar{f}^{1-\lambda}(\Theta)\exp\left[\langle \mu, g(\Theta)\rangle\right] \tag{6.65}$$
parameterized by a scalar λ and an array μ. The optimum can be searched for them only. This variant is not used yet.
3. *A proper choice of the function $g(\Theta)$ helps us to stay within a desirable class pdfs. Typically, we try to stay within the computationally advantageous class of conjugate pdfs, (3.12).*
4. *A linear term in $\int \tilde{\mu}' g(\Theta)(\hat{f}(\Theta) - f(\Theta))\, d\Theta$ can be included into the restriction without changing form of the results. It adds a freedom that may be used for obtaining a suitable form of $\hat{f}$.*
5. *Restriction $\hat{f} \geq 0$ was not explicitly considered while optimizing. Luckily enough, the solution meets it automatically.*

The initialization results based on Proposition 6.4.8 were found too much sensitive with respect to a specific choice of the optional parameters $\langle \cdot, \cdot \rangle$, $g(\Theta)$, ω. For this reason, an alternative formulation was tried. The following proposition reflects a step made in this direction. It gets rid off the choice of the product $\langle \cdot, \cdot \rangle$ and supports intuitive reasons for selecting $g(\Theta) = \Theta$.

Proposition 6.4.9 (Alternative shifted factor) *Let us consider pdfs $\hat{f}, f$ with a common support Θ^*. Let the pdfs $\hat{f}, f$ have finite gradients on Θ^*. Let moreover a function $g : \Theta^* \to \Theta^*$ and scalars $\beta, b \in (0,1)$ be chosen. We search for such a pdf $\hat{f}$ that*

- *minimizes the KL distance $\mathcal{D}\left(\hat{f}||f\right)$,*
- *has the prescribed shift in expectation of $g(\Theta)$*
$$f\left(\int g(\Theta)\hat{f}(\Theta)\, d\Theta\right) = \beta f\left(\int g(\Theta) f(\Theta)\, d\Theta\right),\ \beta \in (0,1), \tag{6.66}$$
- *has the prescribe ratio of "peaks"*
$$\hat{f}\left(\int g(\Theta)\hat{f}(\Theta)\, d\Theta\right) = b^{-1} f\left(\int g(\Theta) f(\Theta)\, d\Theta\right),\ b \in (0,1). \tag{6.67}$$

Then, it has the form

$$\hat{f}(\Theta) \propto f(\Theta)\exp[\mu' g(\Theta)] \tag{6.68}$$

determined by a vector μ chosen so that the restrictions (6.66), (6.67) are met.

Proof: The Lagrangian of this optimization task is

$$\mathcal{J} \equiv \mathcal{D}(\hat{f}||f) + \nu \int \hat{f}(\Theta)\, d\Theta + \rho f\left(\int g(\Theta)\hat{f}(\Theta)\, d\Theta\right) + \lambda \hat{f}\left(\int g(\Theta)\hat{f}(\Theta)\, d\Theta\right).$$

It is determined by scalars ν, ρ and λ that have to be chosen so that restrictions – normalization, (6.66) and (6.67) – are met. Its variation $\delta\mathcal{J}$, that should equal to zero for the optimal $\hat{f}$, reads

$$\delta\mathcal{J} = \int \hat{\delta}\left[\ln(\hat{f}/f) + \nu + 1 + g'(\Theta)\mu\right] d\Theta, \text{ with } \mu = \rho\frac{\partial f(\Theta)}{\partial\Theta} + \lambda\frac{\partial \hat{f}(\Theta)}{\partial\Theta}$$
$$\text{evaluated at the point } \int g(\Theta)\hat{f}(\Theta)\, d\Theta.$$

This implies the claimed results. □

Remark(s) 6.4.11

1. *The requirement (6.66) has the following meaning. The positions of the new factor $\hat{f}(\Theta)$, characterized by $\int g(\Theta)\hat{f}(\Theta)\, d\Theta$ should be still sufficiently probable when judged from the view point of the split pdf. It should, however, differ from the original position of $f(\Theta)$ characterized by $\int g(\Theta)f(\Theta)\, d\Theta$. The selection $\beta < 1$ guarantees it.*

2. *As a rule, there is many pdfs $\hat{f}$ that minimize the distance $\mathcal{D}\left(\hat{f}||f\right)$ under the restriction (6.66). We can use this freedom for selecting a more concentrated pdf than the split one so that the corresponding parameter uncertainty projected into the data space is smaller. The requirement (6.67) expresses this wish. The requirement fixes ratio of "peaks" of the involved pdfs. The pdf values at expectations determining respective positions are taken as the compared peaks. The selection $b < 1$ guarantees that the new pdf has a higher peak and thus is more narrow than the split factor f.*

3. *Intuitively, for each pair $(\beta, b) \in (0,1)\times(0,1)$ there are pdfs fulfilling both conditions (6.66), (6.67). Non-emptiness of this set has to be of course checked. Those pdfs that minimize the distance to the split pdf while meet these conditions are taken as a result of splitting. Additional conditions can be formulated if the formulated task has more solutions.*

4. *The specific choices β, b have to be studied in order to find a compromise between the contradictory requirements. We want*
   - *make the shift as large as possible,*
   - *stay within the domain of attraction of the factor split,*
   - *make the resulting pdf reasonably narrow.*

Intuition and experiments indicate that the choice $g(\Theta) = \Theta$ is reasonable. The results were, however, found sensitive to the optional parameters β, b. This stimulated attempts to search for a more objective choice restrictions. Proposition 2.5.3 provides a guide. According to it, the posterior pdf $f(\Theta|d(t))$ concentrates on minimizing arguments of entropy rate (2.48). For motivating thoughts, we assume that it converges to its expected value and that we deal with the parameterized model from the exponential family, Section 3.2. Thus, the posterior pdf $f(\Theta|d(t))$, conjugated to this family, concentrates on

$$\lim_{\mathring{t}\to\infty} \operatorname{supp}\left[f(\Theta|d(\mathring{t}))\right] = \operatorname{Arg}\min_{\underline{\Theta}\in\Theta^*}\left\{-\ln(A(\underline{\Theta})) - {}^{\lfloor o}\mathcal{E}\left[\langle B([d,\psi']'), C(\underline{\Theta})\rangle\right]\right\} \equiv \operatorname{Arg}\min_{\underline{\Theta}\in\Theta^*}\mathcal{H}_\infty\left({}^{\lfloor o}f||\underline{\Theta}\right). \tag{6.69}$$

Here, $\underline{\Theta}$ is a non-random argument of the posterior pdf $f(\Theta = \underline{\Theta}|d(t))$ and ${}^{\lfloor o}\mathcal{E}[\cdot]$ is the expectation assigned to the objective pdf ${}^{\lfloor o}f(d,\psi)$. The weighting used in connection with the approximate estimation, Section 6.5, guarantees that we deal with data vectors in a relatively narrow set. Thus, we can assume that the mixture character of the objective pdf is reflected in the conditional pdf (6.61) only, i.e. we assume

$$ {}^{\lfloor o}f(d,\psi) = [\beta f(d|\psi,\Theta_1) + (1-\beta) f(d|\psi,\Theta_2)]\, f(\psi), \text{ where} \tag{6.70} $$

the pdf $f(\psi)$ brings no information on $\beta, \Theta_1, \Theta_2$. In other words, a sort of natural conditions of decision making (2.36) is adopted.

Then, the identity (6.69) relates points in supp $\left[f(\Theta|d(\mathring{t}))\right]$, to the parameters β, Θ_1, Θ_2 and can motivate a restriction on the split factors. In Chapter 8 this possibility is elaborated so that the split into a pair of pdfs $f(\Theta_1|d(\mathring{t}))$, $f(\Theta_2|d(\mathring{t}))$ is made without a full estimation of the mixture (6.61). It decreases significantly the computational burden connected with the hierarchical splitting.

Construction of a new mixture: step 5 in Algorithm 6.4.7

Splitting of factors provides potentially huge amount of components. A prospective component is formed from new factors with parameters described by pdfs $\hat{f}_{j_i}(\Theta_{ic}|d(\mathring{t}))$, $j_i \in \{1,2\}$. A deductive selection of a reasonably small subset of such components is impossible. For this reason, a non-deductive rule has to be chosen. At present, we let each component split at most into a pair of component by selecting respective factors in a predefined way. Random choice seems to be a better alternative and tests confirmed it. Also, the methodology selecting the most advantageous variants in an extensive set, proposed in connection with so called shadow cancelling problem, see Section 12.3.2, is worth of elaborating.

6.4.9 Techniques applicable to static mixtures

Majority of available techniques, including those considered at early stage of the project, are applicable to static mixtures that have constant regression vectors. The following subsection demonstrates that these techniques are approximately applicable to dynamic mixtures. Then, promising but not elaborated techniques suitable for static mixtures are outlined.

Static clustering of dynamic systems

Cluster algorithms are often used for an approximate dynamic clustering. They simply neglect mutual dependence of data vectors $\Psi(\mathring{t})$ and treat them as independent records. Thus, it makes sense to analyze properties of this approximation. We adopt the Bayesian viewpoint, having background in [28], to tackle this problem. For presentation simplicity, we assume a restricted but important special case of discrete-valued data and state in the phase form, Remark 6.3.2, with an over-estimated order ∂. Thus, our analysis is made under the following restrictions.

Requirement 6.4.2 (On analysis of static treatment of dynamic mixtures) *We assume that*

- *data d_t have discrete values,*
- *data vectors Ψ_t are modelled by a static mixture*

$$ f(\Psi_t|\Psi(t-1),\Theta) \equiv f(\Psi_t|\Theta); \tag{6.71} $$

- *the chosen prior pdf $f(\Theta)$ is positive on Θ^*,*
- *data vectors are in the phase form $\Psi_t = [d_t', \ldots, d_{t-\partial}', 1]' \equiv [d_t', \phi_{t-1}']'$ with a finite order ∂,*
- *the state of the objective pdf of data ${}^{\lfloor o}f(d(\mathring{t}))$, cf. Chapter 2, is uniquely determined by the model state ϕ_{t-1}, $t \in t^*$,*

$$ {}^{\lfloor o}f(d_t|d(t-1)) = {}^{\lfloor o}f(d_t|\phi_{t-1}) \text{ and } {}^{\lfloor o}f(d(\mathring{t})) = \prod_{t\in t^*} f(d_t|\phi_{t-1}). \tag{6.72} $$

Proposition 6.4.10 (On independently treated data vectors) *Let Requirement 6.4.2 be met. If a unique* ${}^{\lfloor o}\Theta \in \Theta^* \cap \mathrm{supp}\,[\,f(\Theta)]$ *exists such that*

$$f(\Psi_t|\,{}^{\lfloor o}\Theta) = {}^{\lfloor o}f(\Psi_t),\ \forall \Psi_t \in \Psi_t^*, \tag{6.73}$$

then the formal posterior pdf $\prod_{t\in t^*} f(\Psi_t|\Theta)f(\Theta)$ *concentrates on* ${}^{\lfloor o}\Theta$. *Otherwise, the posterior pdfs* $\prod_{t\in t^*} f(\Psi_t|\Theta)f(\Theta)$ *concentrates on minimizing arguments of the entropy rate*

$$\mathrm{Arg}\min_{\Theta\in\Theta^*} \overline{\lim}_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}\ln\left(\frac{{}^{\lfloor o}f(\Psi_\tau)}{f(\Psi_\tau|\Theta)}\right).$$

The statement holds in spite of the neglected dependence of data vectors $\Psi(\mathring{t})$.

Proof: The <u>formal</u> approximate posterior pdf can be expressed as follows

$$f(\Theta|\Psi(t)) \propto f(\Theta)\exp\left\{-t\frac{1}{t}\sum_{\tau=1}^{t}\ln\left(\frac{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}{f(\Psi_\tau|\Theta)}\right)\right\}. \tag{6.74}$$

Let ${}^{\lfloor o}\mathcal{E}$ denote objective expectation corresponding to the objective pdf ${}^{\lfloor o}f(d_t|\phi_{t-1})$. Then, for any $\Theta\in\Theta^*$ and it holds

$$\begin{aligned}
&{}^{\lfloor o}\mathcal{E}\left[\prod_{\tau=1}^{t}\frac{f(\Psi_\tau|\Theta)}{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}\right] = \int\left[\prod_{\tau=1}^{t}\frac{f(\Psi_\tau|\Theta)}{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}\,{}^{\lfloor o}f(d_\tau|d(\tau-1))\right]dd(\mathring{t}) \equiv\\
\underbrace{\equiv}_{(6.72)}\ &\int\left[\prod_{\tau=1}^{t}\frac{f(\Psi_\tau|\Theta)}{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}\,{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})\right]dd(\mathring{t}) \equiv\\
\underbrace{\equiv}_{canceling}\ &\int\prod_{\tau=1}^{t}f(\Psi_\tau|\Theta)\,dd(\mathring{t}) \underbrace{=}_{chain\ rule} \int\prod_{\tau=1}^{t}f(d_\tau|\phi_{\tau-1},\Theta)f(\phi_{\tau-1}|\Theta)\,dd(\mathring{t}) \leq\\
\underbrace{\leq}_{discrete\ \phi_{t-1}}\ &\int\prod_{\tau=1}^{t}f(d_\tau|\phi_{\tau-1},\Theta)\,dd(\mathring{t}) \underbrace{=}_{normalisation} 1.
\end{aligned} \tag{6.75}$$

In summary, for the discrete-valued ϕ_{t-1}, it holds

$$0 < {}^{\lfloor o}\mathcal{E}\left[\prod_{\tau}^{t}\frac{f(\Psi_\tau|\Theta)}{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}\right] \leq 1. \tag{6.76}$$

Similar manipulations with the conditional version of ${}^{\lfloor o}\mathcal{E}$ pdf reveals that

$$\zeta_t \equiv \prod_{\tau}^{t}\frac{f(\Psi_\tau|\Theta)}{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}$$

is non-negative super-martingale. Thus, [27] it converges almost surely to a non-negative variable smaller than one. Consequently, the inversion ζ_t^{-1} converges almost surely either to infinity or to the variable greater than one. Consequently,

$$\overline{\lim}_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}\ln\left(\frac{{}^{\lfloor o}f(d_\tau|\phi_{t-1})}{f(\Psi_\tau|\Theta)}\right) \geq 0$$

almost surely for $t\to\infty$. The form of (6.74) – notice the factor $-t$ – implies that the formal posterior pdf concentrates on

$$\begin{aligned}
&\mathrm{Arg}\min_{\Theta\in\Theta^*}\overline{\lim}_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}\ln\left(\frac{{}^{\lfloor o}f(d_\tau|\phi_{\tau-1})}{f(\Psi_\tau|\Theta)}\right) =\\
=\ &\mathrm{Arg}\min_{\Theta\in\Theta^*}\overline{\lim}_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}\ln\left(\frac{{}^{\lfloor o}f(\Psi_\tau)}{f(\Psi_\tau|\Theta)}\right) = \mathrm{Arg}\min_{\Theta\in\Theta^*}\overline{\lim}_{t\to\infty}\mathcal{H}_t({}^{\lfloor o}f(\cdot)||f(\cdot|\Theta)).
\end{aligned}$$

The first identity is implied by the chain rule ${}^{\lfloor o}f(\Psi_\tau) = {}^{\lfloor o}f(d_\tau|\phi_{\tau-1})\, {}^{\lfloor o}f(\phi_{\tau-1})$ and by the fact that the added term $\frac{1}{t}\sum_{\tau=1}^{t} \ln({}^{\lfloor o}f(\phi_{\tau-1}))$ does not influence minimization. The last identity is just definition of the entropy rate.

If the unique ${}^{\lfloor o}\Theta \in \Theta^*$ exists such that ${}^{\lfloor o}f(\Psi_\tau) = f\left(\Psi_\tau|\, {}^{\lfloor o}\Theta\right)$, then it is the unique minimizing argument.

□

Remark(s) 6.4.12

1. *The result is not surprising: a dynamic model introduces a specific dependence structure among data records $\{d_t\}$. The use of data vectors Ψ_t instead of d_t offers a similar but less structured freedom. Hence, the asymptotic coincidence is possible whenever $\Psi_t,\ t \in t^*$ "overlap", the dependence structure of the correct dynamic model.*

2. *The previous statement can be nicely illustrated on the special case of normal components.*

3. *The performed analysis indicates that the algorithms for static mixtures are applicable in the dynamic case, at least asymptotically.*

4. *The parameterized mixture can be interpreted as a superposition of two random generators. First, a pointer $c_t \in c^*$ is selected with probability α_c. Then, data are randomly generated from this component. In the considered mixtures, the generated pointers $\{c_t\}_{t\in t^*}$ are mutually independent. This independence is a bit illogical in the dynamic context. modelling of pointers by Markov chains would be more adequate. With such an option, the estimation becomes much harder. The pointers could be, however, approximately treated as pairs c_t, c_{t-1} with dependence on c_{t-1}, c_{t-2} neglected. The presented result makes this extension of the mixture model worth of considering.*

Problem 6.4.5 (Weakening of the adopted conditions in Proposition 6.4.10) *The conditions of Proposition 6.4.10 can be almost surely made weaker. It would be worthwhile to make inspection in this respect as there is a chance for a better understanding of the result. Also, estimates of the convergence rate could be and should be searched for. The latter item would have obviously a direct practical impact.*

Some non-elaborated ideas

Here, potential directions are mentioned without a detailed elaboration. The initialization can be based on the following ideas.

Splitting of data space The idea of this technique is similar to that of branching by factor splitting. All operations are, however, made in the data space. The technique is applicable to static mixtures. The technique can be applied to dynamic mixtures when data vectors Ψ are treated as independent data records.

Gradual extension of low-dimensional mixtures The approximate Bayesian mixture estimation for low-dimensional data records is a computationally feasible task. For them, even the initial locations and widths can be equally spaced within the region, where data and thus their static mean values occur. Then, the approximate Bayesian estimation is expected to provide good results within a reasonable computational time. This speculation has motivated the first solution we have ever used. The solution is described in terms of data records $\{d_t\}$ but it is directly applicable to the data vectors $\{\Psi_t\}$.

Algorithm 6.4.10 (Extension of low-variate static mixtures)
Initial mode

- *Select the upper bound $\mathring{c}_i$ on the number c_i of components on each scalar axis $i \in \{1, \ldots, \mathring{d}\}$.*
- *Specify prior pdfs $f(\Theta_i)$ related to single-dimensional mixtures assigned to each entry of d_i.*
- *Estimate the individual mixtures $f(\Theta_i|d_i(\mathring{t}))$, $i = 1, \ldots, \mathring{d}$, by some approximate technique, Section 6.5.*

- *Select the most significant components (coinciding with factors), see Section 6.6.*
- *Apply branching versions of the flattening operation, Algorithms 6.4.2, 6.4.3. This gives good single-variate prior pdfs* $f(\Theta_i)$.
- *Set the dimension of the data space to be extended* $\mathring{\underline{d}} = 1$.
- *Denote* $f(\Theta_{1\_\mathring{\underline{d}}}) = f(\Theta_1)$.

Extension mode

1. *Set* $\mathring{\underline{d}} = \mathring{\underline{d}} + 1$.
2. *Assume the mixture defined on data* $d_{1\_\mathring{\underline{d}}}$ *as* $f\left(d_{1\_\mathring{\underline{d}}}|\Theta_{1\_\mathring{\underline{d}}}\right) \equiv f\left(d_{1\_(\mathring{\underline{d}}-1)}|\Theta_{1\_(\mathring{\underline{d}}-1)}\right) f\left(d_{\mathring{\underline{d}}}|\Theta_{\mathring{\underline{d}}}\right)$.
3. *Define the prior pdf on its parameters as* $f\left(\Theta_{1\_\mathring{\underline{d}}}\right) = f\left(\Theta_{1\_(\mathring{\underline{d}}-1)}) f(\Theta_{\mathring{\underline{d}}}\right)$.
4. *Perform an approximate parameter estimation using data* $d_{1\_\mathring{\underline{d}}}(\mathring{t})$ *to get* $f\left(\Theta_{1\_\mathring{\underline{d}}}|d_{1\_\mathring{\underline{d}}}(\mathring{t})\right)$.
5. *Select the most significant components, see Section 6.6.*
6. *Apply branching versions of the flattening operation, Algorithms 6.4.2, 6.4.3. This gives good lower-variate prior pdfs* $f\left(\Theta_{1\_\mathring{\underline{d}}}\right)$.
7. *Go to the beginning of* Extension mode *if* $\mathring{\underline{d}} < \mathring{d}$. *Otherwise stop and take the resulting prior pdf as the prior on the whole parameter space.*

Remark(s) 6.4.13

The experimental results obtained are reasonable. They exhibit a dependence on the order in which the extension is performed. This may be compensated by starting the procedure from the most technologically important variables or by a random permutation of the ordering used.

A direct extension of single variate components The following variant of the extension provides another possibility of the iterative Bayesian estimation.

Again, we perform estimation of single variate mixtures for all variables d_i (or Ψ_i). Each variable d_i (Ψ_i) is individually described by a mixture

$$f(d_i|\Theta_i) = \sum_{c_i \in c_i^*} \alpha_{c_i} f(d_i|\Theta_{ic_i}, c_i),\ i = 1, \ldots, \mathring{d}. \tag{6.77}$$

Its components are identical with factors. This provides a set of factors for building multi-variate components. We take the single-variate components as independent factors forming multi-variate components. We have insufficient reasons for other choice at this stage of the construction, see also Proposition 12.3.1. The potential multi-variate components have structures determined by the one-to-one composition rule π

$$\begin{aligned} \pi : \quad & c^* \rightarrow [c_1^*, \ldots, c_{\mathring{d}}^*] \Leftrightarrow \\ \pi : \quad & \text{the multi-variate component } c \text{ is the product of factors} \\ & \text{that coincide with components } \pi(c)_1 \in c_1^*, \ldots, \pi(c)_{\mathring{d}} \in c_{\mathring{d}}^* \text{ in (6.77).} \end{aligned} \tag{6.78}$$

The number of such mappings π is very large. Thus, we cannot inspect them fully and suitable candidates have to guessed from the available single-variate results. The selection of promising composition rules is called *shadow cancelling problem.* Essentially, we want to avoid the combinations that contain little data and appeared just due to the non-uniqueness of reconstructing multi-variate objects from their low-dimensional "shadows". A solution is proposed in Chapter 12.

Clustering of pointers The troubles with the discussed shadow cancelling problem, led to the idea to record weights of individual single-variate components generated in approximate estimation and to cluster these pointers. This clustering is of a similar complexity as the original one. We expect, however, a better separation of the "weight" clusters. Moreover, a coupling with MT algorithm [11], Chapter 12, suits ideally to it as the static clustering is adequate and the box width determining it can simply be chosen as the probabilistic weights are known to be in the interval $[0, 1]$.

Combination of solutions Feasible solutions are mostly non-deductive and thus susceptible to failures or to stopping at local optimum. Thus, a combination of techniques is the must. Generally,

- random element in the adopted search increases the chance for reaching the global optimum,
- quality of the results can hardly be monotonous function of the problem parameters thus, floating search has to be considered: both extensions of low dimensional solutions as well as reductions of high dimensional ones have to be used,
- the v-likelihood, Section 6.1.2, seems to be the only viable universal tool for comparing quality of the alternative results.

Problem 6.4.6 (Extension of the model set) *The exponential family contains a wider set of pdfs than the dynamic exponential family. Thus, the static treatment of dynamic components opens possibility to extend the set of feasible models. This possibility should be widely exploited.*

Problem 6.4.7 (Bring new and elaborate above ideas) *This section opens a lot of possible directions to be followed. Even conclusions that some of the sketched ideas are crazy would be invaluable.*

6.5 Approximate parameter estimation

The estimation described by the Proposition 2.5.2 is quite formal for the mixture model (6.1) since the resulting posterior pdf consists of a product of sums of functions of unknown parameters. The number of terms that should be stored and handled blows up exponentially. An approximation is therefore required.

The existing algorithms do not support estimation of dynamic mixtures. This has motivated our search for an adequate algorithm. A slight extension of a known algorithm [49], labelled quasi-Bayes mixture estimation, was proposed in [73] and its properties illustrated on simulation examples. It is discussed in Section 6.5.1.

The gained insight into the approximate learning has allowed us to extend the classical *expectation-maximization algorithm (EM)* to the dynamic mixtures, see Section 6.5.2. It serves us as an auxiliary initialization tool and as a golden standard suitable for comparisons. Note that Markov Chain Monte Carlo (MCMC) techniques are also worth of inspecting but the known applications deal still with much lower dimensions than we need [74].

The quasi-Bayes estimation has good properties, Bayesian motivation and interpretation as well as predictable and tractable computational complexity. It is of a recursive nature so that it may serve for obtaining the adaptive advisory system. These reasons justify our preference for it. At the same time, its construction implies that its results depend on the order of the data processing. This drawback was found significant in some applications dealing with static mixtures. This made us to design a hybrid of the quasi-Bayes and EM algorithms. This algorithm, called *batch quasi-Bayes estimation*, is described in Section 6.5.3. It preserves the essence of the Bayesian paradigm needed in other tasks like in structure estimation, see Section 6.6. At the same time, its results are independent of the processing order.

The *quasi-EM algorithm* is another hybrid proposed. It can be viewed as a rather simple quasi-Bayes point estimation and allows us to created universal approximate estimation covering all four cases discussed, Section 6.5.4.

6.5.1 Quasi-Bayes estimation

For the design and description of the quasi-Bayes estimation, we introduce discrete random pointers $c(\mathring{t}) = (c_1, \ldots, c_{\mathring{t}})$, $c_t \in c^*$, to particular components. They are assumed to be mutually independent with time-invariant probabilities $f(c_t = c|d(t-1), \Theta) = \alpha_c$. With these pointers, the parameterized model (6.1) can be interpreted as the marginal pdf $f(d_t|d(t-1), \Theta)$ of the joint pdf, cf. Agreement 5.1.1,

$$f(d_t, c_t|d(t-1), \Theta) = \prod_{c\in c^*} \left[\alpha_c f(d_t|\phi_{c;t-1}, \Theta_c, c)\right]^{\delta_{c,c_t}} \underbrace{=}_{(6.2)} \prod_{c\in c^*} \left[\alpha_c \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)\right]^{\delta_{c,c_t}}, \text{ where} \tag{6.79}$$

δ is the *Kronecker symbol* defined by the formula

$$\delta_{c,\tilde{c}} \equiv \begin{cases} 1 & \text{if } c = \tilde{c} \\ 0 & \text{otherwise} \end{cases}.$$

Indeed, the marginal pdf $f(d_t|d(t-1), \Theta)$ of the pdf $f(d_t, c_t|d(t-1), \Theta)$, that is found by summing (6.79) over $c_t \in c^*$, has the form (6.1), (6.2).

Let the approximate pdf $f(\Theta|d(t-1))$ be of the product form, cf. Agreement 6.1.1,

$$f(\Theta|d(t-1)) = Di_\alpha(\kappa_{t-1}) \prod_{c\in c^*} \prod_{i\in i^*} f(\Theta_{ic}|d(t-1)) \propto \prod_{c\in c^*} \alpha_c^{\kappa_{c;t-1}-1} \prod_{i\in i^*} f(\Theta_{ic}|d(t-1)). \tag{6.80}$$

The posterior pdfs $f(\Theta_{ic}|d(t-1))$ describe parameters of individual factors and $Di_\alpha(\kappa_{t-1})$ is the pdf describing component weights. It is determined by the vector κ_{t-1} with positive entries $\kappa_{c;t-1} > 0$. They form statistic, i.e. they are known values of scalar functions of $d(t)$. Recall that the *Dirichlet pdf* $Di_\alpha(\kappa)$ estimating component weights $\alpha = [\alpha_1, \ldots, \alpha_{\mathring{c}}]$ is defined

$$Di_\alpha(\kappa) \equiv \frac{\prod_{c\in c^*} \alpha_c^{\kappa_c - 1}}{\mathcal{B}(\kappa)}, \ \mathcal{B}(\kappa) \equiv \frac{\prod_{c\in c^*} \Gamma(\kappa_c)}{\Gamma\left(\sum_{c\in c^*} \kappa_c\right)}.$$

Its properties are discussed in detail in Chapter 10.

The assumption (6.80), the formula (6.79) and the Bayes rule (2.8) give

$$f(\Theta, c_t|d(t)) \propto \prod_{c\in c^*} \alpha_c^{\kappa_{c;t-1}+\delta_{c,c_t}-1} \prod_{i\in i^*} \left[f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)\right]^{\delta_{c,c_t}} f(\Theta_{ic}|d(t-1)). \tag{6.81}$$

In order to obtain an approximation of the desired pdf $f(\Theta|d(t))$, we have to eliminate c_t from (6.81). The correct marginalization with respect to c_t destroys the feasible product form (6.80). It is preserved if δ_{c,c_t} is replaced by its point estimate. We approximate δ_{c,c_t} by its conditional expectation

$$\delta_{c,c_t} \approx \mathcal{E}[\delta_{c,c_t}|d(t)] = f(c_t = c|d(t)). \tag{6.82}$$

By integrating (6.81) over the parameters Θ, cf. Proposition 10.1.1, it can be found that this probability has the form

$$\begin{aligned} w_{c;t} &\equiv f(c_t = c|d(t)) = \frac{\hat{\alpha}_{c;t-1} \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c};t-1} \prod_{i\in i^*} f(d_{i\tilde{c};t}|\psi_{i\tilde{c};t}, d(t-1), \tilde{c})} = \\ &= \frac{\kappa_{c;t-1} \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t-1} \prod_{i\in i^*} f(d_{i\tilde{c};t}|\psi_{i\tilde{c};t}, d(t-1), \tilde{c})}. \end{aligned} \tag{6.83}$$

Here,

$$f(d_{ic;t}|\psi_{ic;t}, d(t-1), c) \equiv \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) f(\Theta_{ic}|d(t-1))\, d\Theta_{ic} \tag{6.84}$$

$$\hat{\alpha}_{c;t-1} = \frac{\kappa_{c;t-1}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t-1}}$$

are the Bayesian prediction (2.38) for a single factor identified by indices i, c and $\hat{\alpha}_{c;t-1}$ the conditional expectations of α_c, cf. (6.5) and Chapter 10. The formula (6.83) can be interpreted as the Bayes rule applied to the discrete unknown random variable $c_t \in c^*$ with the prior probability $\hat{\alpha}_{c;t-1} \propto \kappa_{c;t-1}$.

By inserting the approximation (6.82), (6.83) into (6.81), the approximately updated posterior pdf has the same functional product form as in (6.80) with

$$\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t} \tag{6.85}$$
$$f(\Theta_{ic}|d(t)) \propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{c;t}} f(\Theta_{ic}|d(t-1)). \tag{6.86}$$

This step completes the design of the estimation algorithm.

Algorithm 6.5.1 (Quasi-Bayes estimation without common factors in components)

Initial (off line) mode

- *Select the complete structure of the mixture.*
- *Select the prior pdfs* $f(\Theta_{ic})$ *of the individual factors, ideally, in the conjugate form (3.12) with respect to the parameterized factors* $f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)$.
- *Select initial values* $\kappa_{c;0} > 0$ *of statistics describing prior pdf* α. *Intuitively motivated values about* $0.1\mathring{t}$ *were found reasonable.*
- *Compute the initial estimates of the component weights* $\hat{\alpha}_{c;t} = \frac{\kappa_{c;t}}{\sum_{\tilde{c}c^*} \kappa_{\tilde{c};t}}$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots$,

1. *Acquire the data record* d_t.
2. *Compute the values of the predictive pdfs*
$$f(d_{ic;t}|\psi_{ic;t}, d(t-1), c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) f(\Theta_{ic}|d(t-1))\, d\Theta_{ic} = \frac{\mathcal{I}(d(t)|ic)}{\mathcal{I}(d(t-1)|ic)},$$
see (2.47), for each individual factor $i \in i^* = \{1, \ldots, \mathring{d}\}$ *in all components* $c \in c^*$ *and the measured data record* d_t.
The adaptive, one-stage-ahead predictor, see Section 6.1.2, is thus used.
3. *Compute values of the predictive pdfs* $f(d_t|d(t-1), c) = \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)$ *for the measured data* d_t *and each individual component* $c \in c^*$.
4. *Compute the probabilistic weights* $w_{c;t}$ *using the formula (6.83).*
5. *Update the scalars* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}$ *according to (6.85).*
6. *Update the Bayesian parameter estimates* $f(\Theta_{ic}|d(t-1))$ *of individual factors according to the weighted Bayes rule(6.86)*
$$f(\Theta_{ic}|d(t)) \propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{c;t}} f(\Theta_{ic}|d(t-1)).$$
7. *Evaluate, the point estimates of the mixing weights* $\mathcal{E}[\alpha_c|d(t)] = \frac{\kappa_{c;t}}{\sum_{\tilde{c} \in c^*} \kappa_{\tilde{c};t}} \equiv \hat{\alpha}_{c;t}$ *and if need be characteristics of* $f(\Theta_{ic}|d(t))$ *describing other parameters* Θ_{ic}.
8. *Go to the beginning of* Sequential mode *while data are available, while* $t \leq \mathring{t}$.

Remark(s) 6.5.1

1. *The algorithm is applicable whenever the Bayesian estimation and prediction for each factor can be simply calculated. This singles out parameterized models admitting finite-dimensional statistics as the candidates among which factors should be chosen. This class consists essentially of the* exponential family *augmented by the uniform distribution with unknown boundaries, Section 3.2. Such models are known to have conjugate (self-reproducing) prior pdfs for which only finite-dimensional statistics need to be stored and updated. Normal regression models, Chapter 8, or Markov chains, Chapter 10, are prominent examples of such dynamic factors. Other cases are more or less restricted to static factors.*

2. *It is worth of noticing that the predictions of individual factors without data weighting is performed for evaluation of the weights* $w_{c;t+1}$. *The parameter estimation algorithm performs almost the same algebraic operations but with weighted data. This simple observation is exploited in implementation of the algorithm. Moreover, it is widely used in judging of estimation quality, Section 6.1.2.*

Problem 6.5.1 (Alternative approximation of unknown δ_{c,c_t}**)** *The proposed solution can be improved by searching for a specific product-form-preserving approximation* $\hat{f}(\Theta|d(t+1))$ *(containing no* c_{t+1}*), which is the nearest one to (6.81) in terms of the KL distance [23]. This task is feasible. It is not clear whether it can bring a significant difference from the presented solution but it is worth of being inspected. It may provide a structurally different form of updating with weighting both the parameterized model and corrected pdf on unknown parameters.*

The described quasi-Bayes algorithm 6.5.1 does not take into account possibility that some factors are common to various components. The extension is, however, straightforward.

Algorithm 6.5.2 (Quasi-Bayes estimation with common factors in components)
Initial (off line) mode

- *Select the complete structure of the mixture.*
- *Select prior pdfs* $f(\Theta_{ic})$ *of the individual factors, ideally, in the conjugate form (3.12) with respect to the parameterized factors* $f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c)$.
- *Select initial values* $\kappa_{c;0} > 0$, *say, about* $0.1\mathring{t}$, *describing prior pdf of the component weights* α.
- *Compute the initial point estimates of the component weights* $\hat{\alpha}_{c;0} = \frac{\kappa_{c;0}}{\sum_{\tilde{c}c^*} \kappa_{\tilde{c};0}}$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_t.
2. *Compute values of the predictive pdfs*

$$f(d_{ic;t}|\psi_{ic;t}, d(t-1), c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) f(\Theta_{ic}|d(t-1))\, d\Theta_{ic} = \frac{\mathcal{I}(d(t)|ic)}{\mathcal{I}(d(t-1)|ic)},$$

   see (2.47), for each individual factor $i \in i^* = \{1, \ldots, \mathring{d}\}$ *in all components* $c \in c^*$ *and the measured data record* d_t.

   The adaptive, one-stage-ahead predictor, see Section 6.1.2, is thus used.

3. *Compute the values of predictive pdfs* $f(d_t|d(t-1), c) = \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)$ *for the measured data* d_t *and each individual component* $c \in c^*$.
4. *Compute the probabilistic weights* $w_{c;t}$ *using the formula (6.83).*
5. *Update the scalars* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}$ *according to (6.85).*
6. *Update the Bayesian parameter estimates of different factors*

$$\begin{aligned} f(\Theta_{ic}|d(t)) &\propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{ic;t}} f(\Theta_{ic}|d(t-1)), \quad \text{where} \\ w_{ic;t} &= \sum_{\tilde{c} \in c_i^*} w_{\tilde{c};t}. \end{aligned} \tag{6.87}$$

   The set c_i^* *includes pointers to components that contain* i*-th factor.*

   This step distinguishes this Algorithm from Algorithm 6.5.1.

7. *Evaluate, the point estimate of the mixing weights* $\mathcal{E}[\alpha_c|d(t)] = \frac{\kappa_{c;t}}{\sum_{\tilde{c} \in c^*} \kappa_{\tilde{c};t}} \equiv \hat{\alpha}_{c;t}$ *and, if need be, characteristics of* $f(\Theta_{ic}|d(t))$ *describing other parameters* Θ_{ic}.
8. *Go to the beginning of* Sequential mode *while data are available, while* $t \leq \mathring{t}$.

6.5.2 EM estimation

Expectation-maximization (EM) algorithm is a classical and successful way of the mixture estimation [49]. The EM algorithm provides a local maximum of the posterior pdf (2.44) over Θ^*.

Its convergence is known to be monotonic but slow. The generic multi-modality of the posterior pdf calls for a repetitive search using various initial conditions. AutoClass [75] is a typical and successful implementation of the EM algorithm. It combats the multi-modality by using random initial conditions. It offers a high chance of finding the global maximum but it is computationally expensive.

The essence of the EM algorithm is described here.

We are given the joint pdf of the observed data $d(\mathring{t})$ and non-measured pointers $c(\mathring{t})$ to individual components conditioned on unknown parameters $\Theta \in \Theta^*$. For fixed measured data $d(\mathring{t})$ and chosen structure of parameterized models, they specify the logarithmic likelihood function

$$\mathcal{L}(\Theta, d(\mathring{t}), c(\mathring{t})) \equiv \ln(f(d(\mathring{t}), c(\mathring{t})|\Theta)) = \sum_{t \in t^*} \sum_{c \in c^*} \delta_{c,c_t} \ln\left(\alpha_c f(d_t|\phi_{t-1}, c, \Theta)\right). \tag{6.88}$$

Similarly as in the quasi-Bayes estimation, a reasonable approximation of δ_{c,c_t} is searched for. Unlike for the quasi-Bayes estimation, a point estimate $\hat{\Theta}$ of Θ is searched for and batch processing is supported.

Algorithm 6.5.3 (EM algorithm for mixture estimation)
Initial mode

- *Select the complete structure of the mixture.*
- *Select the upper bound $\mathring{n}$ on the number n of iterations and set $n = 0$.*
- *Select the initial point estimate $\hat{\Theta}_n$ of Θ.*

Iterative mode

1. *Approximate in batch all δ_{c,c_t}, $t \in t^*$ using all measured data $d(\mathring{t})$ by taking*

$$\delta_{c,c_t} \approx \mathcal{E}\left[\delta_{c,c_t}|d(\mathring{t}), \hat{\Theta}_n\right] \equiv f\left(c_t = c|d(\mathring{t}), \hat{\Theta}_n\right), \ t \in t^*. \tag{6.89}$$

   This is called the E(xpectation) step.

2. *Insert the approximations (6.89) into the log-likelihood function (6.88).*
3. *Find the new point estimate $\hat{\Theta}_{n+1}$ of Θ as the maximizing argument of this approximate pdf. This is called the M(aximization) step.*
4. *Evaluate the value of the approximate log-likelihood attained by the maximizing argument.*
5. *Stop if the value of the approximate log-likelihood has not increased or $n \geq \mathring{n}$. Otherwise, set $n = n + 1$ and go to the beginning of* Iterative mode.

To make the EM algorithm applicable we need to evaluate the estimate of the Kronecker symbol.

Proposition 6.5.1 (Batch estimate of pointers to components) *Let*

$$f(d_t, c_t|d(t-1), c(t-1), \Theta) \equiv f(d_t|d(t-1), c_t, \Theta_{c_t})\alpha_{c_t}, \ t \in t^*,$$

and $\hat{\Theta} \equiv (\hat{\alpha}_1, \ldots, \hat{\alpha}_{\mathring{c}}, \hat{\Theta}_1, \ldots, \hat{\Theta}_{\mathring{c}}) \in \Theta^*$ be a fixed instance of $\Theta \equiv (\alpha_1, \ldots, \alpha_{\mathring{c}}, \Theta_1, \ldots, \Theta_{\mathring{c}})$. Then, the probability of the random pointer $c_t \in c^*$, conditioned on all data $d(\mathring{t})$ and $\hat{\Theta}$, is given by the formula

$$w_{c_t;t} \equiv f(c_t|d(\mathring{t}), \hat{\Theta}) = \frac{\hat{\alpha}_{c_t} f(d_t|\psi_t, c_t, \hat{\Theta})}{\sum_{c \in c^*} \hat{\alpha}_c f(d_t|\psi_t, c, \hat{\Theta})} \propto \hat{\alpha}_{c_t} f(d_t|\phi_{t-1}, c_t, \hat{\Theta}). \tag{6.90}$$

Proof: The chain rule and assumed dependencies imply $f(d(\mathring{t}), c(\mathring{t})|\hat{\Theta}) = \prod_{t \in t^*} f(d_t|\phi_{t-1}, c_t, \hat{\Theta}_c)\hat{\alpha}_{c_t}$. The pdf $f(d(\mathring{t}), c_t|\hat{\Theta})$ is obtained by marginalization

$$f(d(\mathring{t}), c_t|\hat{\Theta}) = \sum_{\{c_\tau \in c^*\}_{\tau \in t^* \setminus t}} f(d(\mathring{t}), c(\mathring{t})|\hat{\Theta}) = f\left(d_t|\phi_{t-1}, c_t, \hat{\Theta}_{c_t}\right)\hat{\alpha}_{c_t} \times \text{term\_independent\_of\_}c_t.$$

The desired conditional pdf is proportional to the above pdf with the proportion factor implied by normalization. In it, the complex term independent of c_t cancels and we get the claimed form. $\square$

Remark(s) 6.5.2

1. *The guaranteed monotonic increase of the likelihood [49] to a local maximum is the key advantage of EM algorithm and justifies the stopping rule used.*
2. *As a rule, EM algorithm is described and used for static components. Here, the needed dynamic version is described.*
3. *The adopted interpretation of the EM algorithm differs from a standard interpretation. It is close to the description of the quasi-Bayes algorithm. It helped us to create a novel batch quasi-Bayes algorithm 6.5.5.*
4. *It is straightforward to adopt the idea of the quasi-Bayes algorithm for creating recursive version of EM algorithm. It is sufficient to correct the parameter estimate used for computing of data weights immediately after acquiring new data.*

EM algorithm 6.5.3 is described at the component level. Its applicability is significantly extended when it is used at factor level, possibly with common factors.

Algorithm 6.5.4 (EM algorithm for mixture estimation with common factors)
Initial mode

- *Select the complete structure of the mixture.*
- *Select the upper bound $\mathring{n}$ on the number n of iterations and set $n = 0$.*
- *Select initial point estimates $\hat{\Theta}_{icn}$ of parameters Θ_{ic} characterizing i-th factor within c-th component.*
- *Select point estimates $\hat{\alpha}_{cn}$ of the components weights α_c, typically, as the uniform pf.*

Iterative mode

1. *Set $\kappa_{n;t} = 0$ and use the current point estimates $\hat{\Theta}_n$ in the following evaluations.*
2. *Set the approximate log-likelihood functions $\mathcal{L}_n(\Theta_{ic}, d(0))$ to be accumulated to zero. They correspond to individual factors within individual components.*

   Sequential mode, *running for $t = 1, 2, \ldots,$*

   (a) *Construct the data vectors $\Psi_{ic;t}$.*

   (b) *Compute the values of the predictive pdfs $f(d_{ic;t}|\psi_{ic;t}, \hat{\Theta}_{icn}, c)$ for each individual factor $i \in i^* = \{1, \ldots, \mathring{d}\}$ in all components $c \in c^*$ using the parameter estimates $\hat{\Theta}_{icn}$ that are constant during the time cycle of the sequential mode.*
   Thus, the approximate, fixed one-stage-ahead predictor with certainty-equivalent approximation is used.

   (c) *Compute the values of predictive pdfs $f(d_t|\psi_{c;t}, \hat{\Theta}_{cn}, c) \equiv \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, \hat{\Theta}_{icn}, c)$ for each individual component $c \in c^*$.*

(d) *Compute the probabilities* $w_{cn;t}$ *approximating the Kronecker symbol* δ_{c,c_t}

$$w_{cn;t} = \frac{\hat{\alpha}_{cn} f(d_t|\psi_{c;t}, \hat{\Theta}_{cn}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}n} f(d_t|\psi_{\tilde{c};t}, \hat{\Theta}_{\tilde{c}n}, \tilde{c})}.$$

(e) *Update the log-likelihood functions describing different factors*

$$\begin{aligned} \mathcal{L}_n(\Theta_{ic}, d(t)) &= w_{icn;t} \ln\left[f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)\right] + \mathcal{L}_n(\Theta_{ic}, d(t-1)), \quad \textit{where} \quad (6.91) \\ w_{icn;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c}n;t}. \end{aligned}$$

Update also the statistics $\kappa_{cn;t} = \kappa_{cn;t-1} + w_{cn;t}$. (6.92)

The set c_i^* *includes pointers to components that contain* i*-th factor.*

(f) *Go to the beginning of* Sequential mode *if* $t \leq \mathring{t}$. *Otherwise continue.*

3. *Find the new point estimates* $\hat{\Theta}_{n+1}$ *of* Θ *as the maximizing arguments of the approximate log-likelihood*

$$\sum_{c\in c^*} \left[\kappa_{cn;\mathring{t}} \ln(\alpha_c) + \sum_{i=1}^{\mathring{d}} \mathcal{L}_n(\Theta_{ic}, d(\mathring{t}))\right].$$

4. *Evaluate the value of the approximate log-likelihood attained by the maximizing argument.*

5. *Stop if the value of the approximate log-likelihood has not increased or* $n \geq \mathring{n}$. *Otherwise, set* $n = n + 1$ *and go to the beginning of* Iterative mode.

Remark(s) 6.5.3

1. *The algorithm can be used for searching maximum a posterior probability estimate (MAP) by selecting non-trivial initial likelihood function* $\mathcal{L}(\Theta, d(0))$ *and positive* $\kappa_{n;0}$.

2. *Obviously, EM algorithm is applicable iff the considered log-likelihood function can be represented in computer. It essentially means that we have to deal with factors belonging to the exponential family, Section 3.2.*

3. *The algorithm is intentionally written in the style of the quasi-Bayes algorithm, Algorithm 6.5.2. It indicates their similarity that enriches both of them. For instance, the maximizing argument can be searched and immediately used within the time cycle. It leads to the adaptive* quasi EM algorithm. *It can be used in on line learning and its convergence can be faster. The loss of the monotonic convergence and of the processing-order independence is the price paid for it.*

6.5.3 Batch quasi-Bayes estimation

The construction of the quasi-Bayes estimation implies that its results depend on the order in which data are processed. Experiments have shown that this dependence may be significant. EM algorithm avoids this drawback but its orientation to the point estimation prevents us to combine it with the advantageous Bayesian framework. For instance, the construction of the prior pdf by splitting, see Section 6.4.8, cannot be exploited. The presented interpretation of the EM algorithm allows us to create a novel batch quasi-Bayes estimation algorithm that is processing-order independent. Essentially, it estimates pointers to components within n-th iteration by using approximation of pdfs of the mixture parameters that is fixed within the n-th stage of the iterative search for the posterior pdf.

Algorithm 6.5.5 (BQB algorithm for mixture estimation with common factors)
Initial mode

- *Select the complete structure of the mixture.*

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations set* $n = 0$.
- *Set maximum of the v-log-likelihood* $\bar{l} = -\infty$.
- *Select the pre-prior pdfs* $\bar{f}(\Theta_{ic})$ *of the individual factors, ideally, in the conjugate form with respect to the parameterized factors* $f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c)$.
- *Select the pre-prior values* $\bar{\kappa}_c > 0$ *used in flattening.*
- *Select the initial guess of the prior pdfs* $f_n(\Theta_{ic})$ *of the individual factors, ideally, in the conjugate form with respect to the parameterized factors* $f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c)$.
- *Select initial values* $\kappa_{cn} > 0$ *determining a prior Dirichlet pdf on component weights.*

Iterative mode

1. *Make copies* $f(\Theta_{ic}|d(0)) = f_n(\Theta_{ic})$ *and* $\kappa_{c;0} = \kappa_{cn}$.
2. *Set the value of v-log-likelihood* $l_{n;0} = 0$.
3. *Compute the point estimates of the components weights* $\hat{\alpha}_{cn} = \frac{\kappa_{cn}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c}n}}$.

   Sequential mode, *running for* $t = 1, 2, \ldots,$

   (a) *Construct the data vectors* $\Psi_{ic;t}$.

   (b) *Compute the values of the predictive pdfs* $f_n(d_{ic;t}|\psi_{ic;t}, c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c) f_n(\Theta_{ic})\, d\Theta_{ic}$ *for each individual factor* $i \in i^* = \{1, \ldots, \mathring{d}\}$ *in all components* $c \in c^*$.
   The prior pdf $f_n(\Theta)$ *is constant during the time cycle. The fixed one-step-ahead predictor, see Section 6.1.2, is thus used.*

   (c) *Compute the values of the predictive pdfs* $f_n(d_t|\psi_{c;t}, c) \equiv \prod_{i\in i^*} f_n(d_{ic;t}|\psi_{ic;t}, c)$ *for each individual component* $c \in c^*$.

   (d) *Update v-log-likelihood* $l_{n;t} = l_{n;t-1} + \ln\left(\sum_{c\in c^*} \hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)\right)$.

   (e) *Compute the probabilistic weights* $w_{cn;t}$ *approximating* δ_{c,c_t}

   $$w_{cn;t} = \frac{\hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}n} f_n(d_t|\psi_{\tilde{c};t}, \tilde{c})}.$$

   (f) *Update copies of the prior pdfs and statistics* κ

   $$\begin{aligned} f(\Theta_{ic}|d(t)) &\propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{icn;t}} f(\Theta_{ic}|d(t-1)), \\ \kappa_{c;t} &= \kappa_{c;t-1} + w_{cn;t} \\ w_{icn;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c}n;t}. \end{aligned} \tag{6.93}$$

   The set c_i^* *includes pointers to components that contain* i*-th factor.*

   (g) *Go to the beginning of* Sequential mode *if* $t \le \mathring{t}$. *Otherwise continue.*

4. *Stop if* $l_{n;\mathring{t}} < \bar{l}$ *or the compared valued are taken as equal, see Proposition 6.1.2, or* $n \ge \mathring{n}$. *Otherwise set* $\bar{l} = l_{n;\mathring{t}}$, *increase the iteration counter* $n = n + 1$ *and apply the flattening operation to*

   $$Di_\alpha(\kappa_{c;\mathring{t}}) \prod_{c\in c^*} \prod_{i\in i^*} f(\Theta_{ic}|d(\mathring{t})).$$

   It provides the new estimate of the prior pdf (6.3) given by $f_n(\Theta_{ic})$, κ_{cn}.

   The flattening factor corresponding to the iterative Bayesian learning, cf. Proposition 6.4.2, has to be used.

5. *Go to the beginning of* Iterative mode.

Remark(s) 6.5.4

The BQB algorithm uses Bayesian predictors for estimating the Kronecker symbol δ_{c,c_t}. They, among other, respect uncertainty of the current estimates of unknown parameters. Predictions become too cautious if this uncertainty is too high. It may break down the algorithm completely. Knowing it, the remedy is simple. Essentially, predictions used in the EM algorithm that ignore these uncertainties have to be used in several initial iterative steps of the algorithm.

6.5.4 Universal approximate estimation

The similarity of descriptions of particular approximate estimations led to the following universal approximate estimation algorithm.

Algorithm 6.5.6 (Universal mixture estimation with common factors)
Initial mode

- *Select the complete structure of the mixture.*
- *Select the upper bound $\mathring{n}$ on the number n of iterations and set $n = 0$.*
- *Set maximum of the v-log-likelihood $\bar{l} = -\infty$.*
- *Select the pre-prior pdfs $\bar{f}(\Theta_{ic})$ of the individual factors, ideally, in the conjugate form with respect to the parameterized factors $f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c)$. They serve for flattening.*
- *Select the initial guess of the prior pdfs $f_n(\Theta_{ic})$ of the individual factors, ideally, in the conjugate form with respect to the parameterized factors $f(d_{ic;t+1}|\psi_{ic;t+1}, \Theta_{ic}, c)$.*
- *Select the values $\bar{\kappa}_c > 0$ used in flattening.*
- *Select the initial values $\kappa_{cn} > 0$.*
- *Select the algorithmic variant **av**.*

Iterative mode

1. Make copies $f_o(\Theta_{ic}) = f_n(\Theta_{ic})$ *and* $\kappa_{oc} = \kappa_{cn}$ *if **av** is a* batch variant.
2. *Compute the point estimates of the components weights* $\hat{\alpha}_{cn} = \frac{\kappa_{cn}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c}n}}$ *and the point estimates* $\hat{\Theta}_{icn}$ *of parameters* Θ_{ic} *of the individual parameterized factors.*
3. *Set the value of v-log-likelihood* $l_{n;0} = 0$. Sequential mode, *running for* $t = 1, 2, \ldots,$
   (a) *Construct the data vectors* $\Psi_{ic;t}$.
   (b) *Compute the values of the predictive pdfs* $f_n(d_{ic;t}|\psi_{ic;t}, c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c) f_n(\Theta_{ic})\, d\Theta_{ic}$ *for each individual factor* $i \in i^* = \{1, \ldots, \mathring{d}\}$ *in all components* $c \in c^*$.[1]
   (c) *Compute values of the predictive pdfs* $f_n(d_t|\psi_{c;t}, c) \equiv \prod_{i\in i^*} f_n(d_{ic;t}|\psi_{ic;t}, c)$ *for each component* $c \in c^*$.[1]
   (d) *Update v-log-likelihood* $l_{n;t} = l_{n;t-1} + \ln\left(\sum_{c\in c^*} \hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)\right)$.[1]
   (e) *Compute the probabilistic weights* $w_{cn;t}$ *approximating* δ_{c,c_t} *according to the variant **av** used*

   av= *-Bayes $\quad w_{cn;t} \equiv \frac{\hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}n} f_n(d_t|\psi_{\tilde{c};t}, \tilde{c})}$

   av=*-EM $\quad w_{cn;t} \equiv \frac{\hat{\alpha}_{cn} \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t}, \hat{\Theta}_{icn}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}n} \hat{\alpha}_{\tilde{c}n} \prod_{\tilde{i}=1}^{\mathring{d}} f(d_{\tilde{i}\tilde{c};t}|\psi_{\tilde{i}\tilde{c};t}, \hat{\Theta}_{\tilde{i}\tilde{c}n}, \tilde{c})}$

[1] This evaluation is not necessary for EM versions but suits for a use in a wider Bayesian context.

(f) *Perform data updating according to the variant* ***av*** *used*

av=quasi-*

$$\begin{aligned} \kappa_{cn} &= \kappa_{cn} + w_{cn;t} \\ f_n(\Theta_{ic}) &\propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{ic;t}} f_n(\Theta_{ic}) \end{aligned}$$

Correct $\hat{\alpha}_{cn} = \frac{\kappa_{cn}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c}n}}$ *and the point estimates* $\hat{\Theta}_{icn}$ *of parameters* Θ_{ic} *of the individual parameterized factors.*

av=batch-* *Update the copies*

$$\begin{aligned} \kappa_{oc} &= \kappa_{oc} + w_{cn;t} \\ f_o(\Theta_{ic}) &\propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{ic;t}} f_o(\Theta_{ic}). \end{aligned}$$

In all cases, $w_{icn;t} = \sum_{\tilde{c}\in c_i^*} w_{\tilde{c};t}$, *where the set* c_i^* *includes pointers to the components that contain* i*-th factor.*

(g) *Go to the beginning of* Sequential mode *if* $t \leq \mathring{t}$ *otherwise continue.*

4. *Stop if* $l_{n;\mathring{t}} < \bar{l}$ *or* $l_{n;\mathring{t}}$, $\bar{l}$ *are not equal, see Proposition 6.1.2, and increase the iteration counter* $n = n + 1$.

5. *Set*

$$\begin{aligned} \kappa_{cn} &= \kappa_{oc} \\ f_n(\Theta_{ic}) &= f_o(\Theta_{ic}) \text{ and refresh point estimates } \hat{\Theta}_{ic} \end{aligned}$$

for ***av=batch-****.

6. *Apply the flattening operation to*

$$f_{n-1}(\Theta) = Di_\alpha(\kappa_{c(n-1)}) \prod_{c\in c^*} \prod_{i\in i^*} f_{n-1}(\Theta_{ic}).$$

It provides the new n*-th estimate of the prior pdf (6.3).*

The flattening factor corresponding to the iterative Bayesian learning, cf. Proposition 6.4.2, has to be used. The factor has to be almost zero if the standard EM algorithm should be obtained.

7. *Go to the beginning of* Iterative mode *if* $n \leq \mathring{n}$.

6.6 Structure estimation

The structure of the mixture is determined by the number and composition of components. Structure of individual factors is included in it, see Agreement 5.2.1. We can list the set of potential candidates for such a selection and label promising sub-selections by a "structural" pointer $S \in S^*$. As its best value is unknown, it becomes a part of the unknown mixture parameter Θ. Its estimation essentially reduces to evaluation of the corresponding marginal probability $f(S|d(\mathring{t}))$. For it, the predictive pdfs $f(d_t|d(t-1), S)$, are computed in parallel for the competing structures $S \in S^*$. The predictive pdfs are needed for computing the posterior pf

$$f(S|d(\mathring{t})) \propto \prod_{t\in t^*} f(d_t|d(t-1), S)\, f(S) = \frac{\mathcal{I}(d(\mathring{t})|S)}{\mathcal{I}(d(0)|S)} f(S). \tag{6.94}$$

Recall that $\mathcal{I}(d(\mathring{t})|S)$ is the normalization factor (2.46) with $\mathcal{P}_{a_{\mathring{t}+1}} = d(\mathring{t})$, computed within S-th structure. $f(S)$ is the optional prior probability assigned to the competitive structures.

It is clear that the complexity of the optimal solution increases with the number of compared structures. This number blows up exponentially with the dimensionality of data space, lengths of regression vectors and the number of components. Thus, we can compare only a small sub-selection of competitive structures. At this moment, the following feasible iterative procedure seems to be promising.

Algorithm 6.6.1 (Conceptual structure estimation)

1. *Mixture is estimated with the richest acceptable structure, both with respect to the number of components and complexity of the involved factors.*
2. *Structure of the individual factors is estimated, see Section 6.6.1.*
3. *Group of variables non-influencing markers are excluded.*
4. *Structures of the individual components are estimated, see Section 6.6.3.*
5. *Similar components are merged, see Section 6.6.4.*
6. *Superfluous components are cancelled, see Section 6.6.4.*
7. *The resulting mixture is flattened and the whole evaluation sequence is iterated till convergence.*

Problem 6.6.1 (Feasibility and reliability of structure estimation) *Structure estimation is known to be hard task even in the uni-modal case. The known problems are surely enhanced in the mixture estimation. Sufficient amount of data seems to be only advantage of the addressed problem. On the other hand, it increases computational load.*

6.6.1 Estimation of factor structure

In this case, structures of uni-modal models with single outputs have to be estimated. This is a classical, well supported problem. For normal factors, efficient numerical procedures exist [36, 76]. For other factors (Markov, MT), see Chapters 10, 12, the adequate procedures have to be prepared but the basic ideas can be "stolen" from the normal case.

6.6.2 Structure estimation in factor splitting

Structure estimation before splitting was found indispensable when splitting is made in the direction of largest uncertainty of the posterior pdf $f(\Theta|d(\mathring{t}))$. This happens in the case optimization based splitting.

Originally, superfluous parameters were excluded when recognized. This action is, however, dangerous when the components are still too far from the correct positions. Requirement 6.4.1 is then easily violated and the parameters originally assumed superfluous may play a significant role. The following way of use of the structure estimation provides the remedy.

Let us apply structure estimation. It splits $\Theta = (\Theta_a, \Theta_b) \equiv$ <u>temporarily</u> (superfluous,significant) parameters.

Using some technique described above, the marginal pdf $f(\Theta_b|d(\mathring{t}))$ is split into a pair $\hat{f}_j(\Theta_b|d(\mathring{t}))$, $j = 1, 2$. The part, which is temporarily taken as superfluous, remains unchanged. It defines the following pair of pdfs on the original parameter space

$$\hat{f}_j(\Theta|d(\mathring{t})) = f(\Theta_a|\Theta_b, d(\mathring{t}))\hat{f}_j(\Theta_b|d(\mathring{t})),\ j = 1, 2. \tag{6.95}$$

It is at disposal for further iterations while preserving possibility to move temporarily superfluous parameter entries among significant ones and vice versa.

6.6.3 Estimation of component structure

A component is a product of factors, see Agreement 5.2.1,

$$f(d_t|\phi_{t-1}, \Theta_c, c) = \prod_{i\in i^*} f(d_{ic;t}|d_{1c;t}, \ldots, d_{(i-1)c;t}, \phi_{t-1}, \Theta_{ic}) \equiv \prod_{i\in i^*} f(d_{ic;t}|, \psi_{ic;t}, \Theta_{ic}). \tag{6.96}$$

The permutation π_c (5.9) of d_t to $d_{c;t}$ influences the overall description of the parameterized component. Thus, it makes sense to search for the "best" ordering, to search for the structure of the component. It can be done by the following conceptual algorithm.

Algorithm 6.6.2 (MAP estimate of a component structure)
Specify the set of compared structures of the component and for all its entries make the following actions.

1. *Estimate structures of the involved factors according to Section 6.6.1.*
2. *Evaluate the likelihood value assigned to the inspected component structure and the best structures of factors.*

Compute the posterior probability of the component structures within the considered set and select the MAP estimate.

The algorithm is parameterized by the set of compared component structures. This set is always somehow restricted. The following restrictions are common.

- Discrete valued entries have to be placed at the end of d-entries in order to get Markov-chain factors.
- Structures in which "privileged variables", i.e. markers, depend on the smallest amount of variables are preferred.

In spite of these restrictions, the complete set is again too large to be inspected fully. Thus, a similar strategy to the factor-structure estimation has to be adopted: a full local search around the best current guess of the structure is to be performed while an improvement is observable.

6.6.4 Merging and cancelling of components

The initialization by the factor splitting, Section 6.4.8, is almost always employed for determining the number of components. During splitting, the number of components increases rather quickly leading to mixtures with too many components. It calls for algorithms reducing the *number of components*. They are discussed here.

We deal with mixtures with different number of components. In order to distinguish them, we attach the left superscript that stresses the number of components involved.

Two types of reductions of number of components are possible. The *merging of components* deals with mixtures

$$
{}^{\lfloor \mathring{c}}f(d_t|d(t-1),\Theta) = \sum_{c=1}^{\mathring{c}} \alpha_c f(d_t|d(t-1),\Theta_c,c)
$$

that contain almost identical components. We merge them and estimate the mixture with less components.

The *cancelling of components* deals with mixtures containing negligible components. We drop them and estimate the mixture with less components.

For a proper treatment of both cases, subsequent paragraphs inspect

- how to modify pdfs on parameters after reducing the number of components,
- how to find the component to be modified,
- when to stop reductions.

How to modify pdfs on parameters?

The mixture is invariant with respect to permutations of its components. Thus, we can inspect merging of the components $\mathring{c}-1, \mathring{c}$ and cancelling of the component $\mathring{c}$.

The verbal description of the component merging implies that

$$
\begin{aligned}
{}^{\lfloor \mathring{c}-1}f(\alpha_1,\ldots,\alpha_{\mathring{c}-1},\Theta_1,\ldots,\Theta_{\mathring{c}-1}) \quad &\propto \quad {}^{\lfloor \mathring{c}}f(\alpha_1,\ldots,\alpha_{\mathring{c}},\Theta_1,\ldots,\Theta_{\mathring{c}}|\alpha_{\mathring{c}-1}=\alpha_{\mathring{c}},\Theta_{\mathring{c}-1}=\Theta_{\mathring{c}}) \underbrace{\propto}_{(6.3)} \qquad (6.97) \\
&\propto \quad \prod_{c=1}^{\mathring{c}-2}\left[\alpha^{\,{}^{\lfloor \mathring{c}}\kappa_c-1}\,{}^{\lfloor \mathring{c}}f(\Theta_c)\right]\alpha_{\mathring{c}-1}^{{}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1}+{}^{\lfloor \mathring{c}}\kappa_{\mathring{c}}-2}\,{}^{\lfloor \mathring{c}}f(\Theta_{\mathring{c}-1})\,{}^{\lfloor \mathring{c}}f(\Theta_{\mathring{c}}=\Theta_{\mathring{c}-1}).
\end{aligned}
$$

It gives the merged component in the form

$$\begin{aligned} {}^{\lfloor \mathring{c}-1}f(\Theta_{\mathring{c}-1}) &\propto {}^{\lfloor \mathring{c}}f(\Theta_{\mathring{c}-1})\,{}^{\lfloor \mathring{c}}f(\Theta_{\mathring{c}} = \Theta_{\mathring{c}-1}) \qquad (6.98)\\ &\quad \text{and its weight is characterized by the statistics}\\ {}^{\lfloor \mathring{c}-1}\kappa_{\mathring{c}-1} &= {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}} - 1 \quad \text{while}\\ &\quad \text{pdfs describing the remaining components and weights are unchanged.} \end{aligned}$$

Note that merging leads to a proper pdf if ${}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}} - 1 > 0$.

The verbal description of the component cancelling implies that

$$\begin{aligned} {}^{\lfloor \mathring{c}-1}f(\alpha_1,\ldots,\alpha_{\mathring{c}-1},\Theta_1,\ldots,\Theta_{\mathring{c}-1}) &= {}^{\lfloor \mathring{c}}f(\alpha_1,\ldots,\alpha_{\mathring{c}},\Theta_1,\ldots,\Theta_{\mathring{c}-1},\Theta_{\mathring{c}}|\alpha_{\mathring{c}}=0,\Theta_{\mathring{c}} \text{ has no influence}) \propto\\ &\underbrace{\propto}_{(6.3)} \prod_{c=1}^{\mathring{c}-1}\left[\alpha_c^{{}^{\lfloor \mathring{c}}\kappa_c - 1}\,{}^{\lfloor \mathring{c}}f(\Theta_c)\right]. \qquad (6.99) \end{aligned}$$

The spurious component is completely neglected and statistics of remaining pdfs are unchanged.

Remark(s) 6.6.1

1. *The considered prior and posterior pdfs have the common form (6.3) so that the inspected question is answered both for prior and posterior pdfs.*
2. *The cancelling implies*

$$ {}^{\lfloor \mathring{c}-1}\mathcal{E}[\alpha_c] \equiv \frac{{}^{\lfloor \mathring{c}}\kappa_c}{\sum_{\tilde{c}=1}^{\mathring{c}-1}{}^{\lfloor \mathring{c}}\kappa_{\tilde{c}}} \geq \frac{{}^{\lfloor \mathring{c}}\kappa_c}{\sum_{\tilde{c}=1}^{\mathring{c}}{}^{\lfloor \mathring{c}}\kappa_{\tilde{c}}} \equiv {}^{\lfloor \mathring{c}}\mathcal{E}[\alpha_c],\ c = 1,\ldots,\mathring{c}-1.$$

   It indicates that there is a chance for an increase of the v-likelihood by omitting a non-negative term in the parameterized mixture.
3. *A similar inequality holds in merging for components $c = 1,\ldots,\mathring{c}-2$ as the normalizing denominator in expression for ${}^{\lfloor \mathring{c}-1}\mathcal{E}[\alpha_c]$ is decreased by 1, cf. (6.98).*
4. *The v-likelihood need not be recomputed after component merging and cancelling if the error of the approximate estimation can be neglected. This fact demonstrated below decreases significantly computational burden.*

What are candidates for merging

We may select the candidate by measuring the KL distance between the posterior pdf ${}^{\lfloor \mathring{c}}f(\Theta|d(\mathring{t}))$ describing the original mixture to that obtained by merging.

In ${}^{\lfloor \mathring{c}-1}f(\Theta|d(\mathring{t}))$ of the form (6.3), ${}^{\lfloor \mathring{c}-1}f(\Theta_c|d(\mathring{t})) = {}^{\lfloor \mathring{c}}f(\Theta_c|d(\mathring{t}))$, ${}^{\lfloor \mathring{c}-1}\kappa_{c;\mathring{t}} = {}^{\lfloor \mathring{c}}\kappa_{c;\mathring{t}}$, $c = 1,\ldots,\mathring{c}-2$. The prospective merged component has the form(6.98)

$$ {}^{\lfloor \mathring{c}-1}f(\Theta_{\mathring{c}-1}|d(\mathring{t})) \propto {}^{\lfloor \mathring{c}}f(\Theta_{\mathring{c}-1}|d(\mathring{t}))\,{}^{\lfloor \mathring{c}}f(\Theta_{\mathring{c}} = \Theta_{\mathring{c}-1}|d(\mathring{t})),\ \ {}^{\lfloor \mathring{c}-1}\kappa_{\mathring{c}-1;\mathring{t}} = {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}} - 1.$$

The pdf ${}^{\lfloor \mathring{c}-1}f(\Theta|d(\mathring{t}))$ acts on the reduced space of parameters. For judging the distance of ${}^{\lfloor \mathring{c}-1}f$ and ${}^{\lfloor \mathring{c}}f$, we have to stay within the original parameter space corresponding to $\mathring{c}$ components. We extend the merger into this space by using the identical estimates $\tilde{f}(\Theta_{\mathring{c}-1}|d(\mathring{t}))$, $\tilde{f}(\Theta_{\mathring{c}}|d(\mathring{t}))$ of the last two components. It means that both these pdfs are determined by the identical statistics. The statistics are selected so that the merger (6.98) is obtained after their merging of both pdfs $\tilde{f}(\cdot)$. Taking into account that the form (6.3) applies to factors, we set

$$\begin{aligned} \tilde{f}(\Theta_{i(\mathring{c}-1)} = \Theta_i|d(\mathring{t})) &\equiv \tilde{f}(\Theta_{i\mathring{c}} = \Theta_i|d(\mathring{t})) \propto \sqrt{{}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)} = \Theta_i|d(\mathring{t}))\,{}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_i|d(\mathring{t}))}\\ \tilde{\kappa}_{\mathring{t}} &\equiv 0.5\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}}\right). \qquad (6.100) \end{aligned}$$

The components $\mathring{c}-1$, $\mathring{c}$ are taken candidate for merging if the following KL distance is small

$$
\begin{aligned}
\mathcal{D}_{(\mathring{c}-1)\mathring{c}} &\equiv \mathcal{D}\left({}^{\lfloor \mathring{c}}f(\cdot|d(\mathring{t}))||\, {}^{\lfloor \mathring{c}}\tilde{f}(\cdot|d(\mathring{t}))\right) = \qquad (6.101)\\
&= \sum_{i=1}^{\mathring{d}} \left[\mathcal{D}\left({}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))||\tilde{f}(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\right) + \mathcal{D}\left({}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}|d(\mathring{t}))||\tilde{f}(\Theta_{i\mathring{c}}|d(\mathring{t}))\right)\right] +\\
&+ \mathcal{D}\left(Di_{\alpha_{\mathring{c}-1},\alpha_{\mathring{c}}}\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}},\, {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}}\right)||Di_{\alpha_{\mathring{c}-1},\alpha_{\mathring{c}}}\left(\tilde{\kappa}_{\mathring{t}},\tilde{\kappa}_{\mathring{t}}\right)\right).
\end{aligned}
$$

This allows us to order candidates for merging. We take into account that

- a difference in the structure of components prevents their merging as they are infinitely far,
- the distance matrix (6.101) is symmetric,
- the numerical construction of the trial merger is necessary for finding components whose merging makes the smallest change of the posterior pdf.

Algorithm 6.6.3 (Selection and merging of a pair of components)
Initial mode

- *Perform estimation of the mixture with a sufficient amount of components $\mathring{c}$ so that the pdfs ${}^{\lfloor \mathring{c}}f(\Theta_{ic}|d(t))$ and statistics ${}^{\lfloor \mathring{c}}\kappa_t$, $t \in \{0,\mathring{t}\}$, $c \in c^*$, $i = 1,\ldots,\mathring{d}$, are at disposal.*
- *Set the minimum value of the KL distance $\underline{\mathcal{D}} = +\infty$ and initialize the indices $(\underline{c},\underline{\tilde{c}}) = (0,0)$ of the component pair to be merged.*

Evaluation mode

1. *Select the pair whose merging causes the smallest change of the KL distance as follows.*

   For $c = 1,\ldots,\mathring{c}-1$
   For $\tilde{c} = c+1,\ldots,\mathring{c}$
   Set the indicator of the common structure $cs = 0$.
   For $i = 1,\ldots,\mathring{d}$
   Set $cs = -1$ and break the cycle over i if the structures of Θ_{ic} and $\Theta_{i\tilde{c}}$ differ.
   Create the pdf $\tilde{f}$ (6.100) needed in the distance measuring with $\mathring{c}-1 \leftrightarrow c$, $\mathring{c} \leftrightarrow \tilde{c}$.
   end of the cycle over i
   Do if $cs = 0$
   Complete definition of the pdf (6.100) used in the KL distance $\tilde{\kappa}_{\mathring{t}} = \frac{1}{2}\left(\kappa_{c;\mathring{t}} + \kappa_{\tilde{c};\mathring{t}}\right)$.
   Evaluate the KL distance $\tilde{\mathcal{D}} \equiv \mathcal{D}_{(\mathring{c}-1)\mathring{c}}$ according to (6.101) with $\mathring{c}-1 \leftrightarrow c$, $\mathring{c} \leftrightarrow \tilde{c}$
   If $\tilde{\mathcal{D}} < \underline{\mathcal{D}}$
   Set $\underline{\mathcal{D}} = \tilde{\mathcal{D}}$, $(\underline{c},\underline{\tilde{c}}) = (c,\tilde{c})$ and store the pdf $\tilde{f}$, $\tilde{\kappa}$.
   end of the test on the value $\tilde{\mathcal{D}}$ of the KL distance
   end of the condition $cs = 0$
   end of the cycle over $\tilde{c}$
   end of the cycle over c

2. *Stop and announce that no merging is possible if $(\underline{c},\underline{\tilde{c}}) = (0,0)$.*
3. *Rename components so that $(\mathring{c}-1,\mathring{c}) = (\underline{c},\underline{\tilde{c}})$.*

4. Finish the merging of the best components found and apply the same operation to the prior pdf

$$\begin{aligned}
{}^{\lfloor \mathring{c}-1}\kappa_{\mathring{c}-1;\mathring{t}} &\equiv 2\tilde{\kappa}_{\mathring{t}} - 1 \equiv {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}} - 1, \quad {}^{\lfloor \mathring{c}-1}\kappa_{\mathring{c}-1;0} \equiv {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;0} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};0} - 1. \\
\textit{For} \quad & i = 1,\ldots,\mathring{d} \\
{}^{\lfloor \mathring{c}-1}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t})) &\equiv \frac{{}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\, {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(\mathring{t}))}{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\, {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\, d\Theta_{i(\mathring{c}-1)}} \propto \left[\tilde{f}(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\right]^2 \\
{}^{\lfloor \mathring{c}-1}f(\Theta_{i(\mathring{c}-1)}|d(0)) &\equiv \frac{{}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(0))\, {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(0))}{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(0))\, {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(0))\, d\Theta_{i(\mathring{c}-1)}}. \qquad (6.102) \\
\textit{end} \quad & \textit{of the cycle over } i.
\end{aligned}$$

The remaining components and statistics are unchanged by the considered merging. It completes the definition of the prior and posterior pdf on the mixture with $\mathring{c}-1$ *components.*

The necessary condition for obtaining the proper pdfs $\kappa_{\mathring{c}-1;t} + \kappa_{\mathring{t};t} > 1,\ t \in \{0, \mathring{t}\}$ should be checked. Such components should not be merged but they are hot candidates for cancelling.

Complexity of the underlying evaluations is better visible on a practical version of this algorithm. It is written for parameterized factors in the exponential family

$$f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) = A(\Theta_{ic}) \exp\left[\langle B(\Psi_{ic;t}), C(\Theta_{ic})\rangle\right],$$

with $A(\Theta_{ic}) \geq 0$ $\Psi_{ic;t} = [d_{ic;t}, \psi'_{ic;t}]'$ and $\langle\cdot,\cdot\rangle$ being a scalar product of vector values of functions $B(\cdot), C(\cdot)$, see Section 3.2. Its conjugate pdfs have the form, $t \in \{0, \mathring{t}\}$,

$$ {}^{\lfloor \mathring{c}}f\left(\Theta|\, {}^{\lfloor \mathring{c}}V_t,\, {}^{\lfloor \mathring{c}}\nu_t,\, {}^{\lfloor \mathring{c}}\kappa_t\right) = \mathcal{B}^{-1}\left({}^{\lfloor \mathring{c}}\kappa_t\right) \prod_{c=1}^{\mathring{c}} \alpha_c^{{}^{\lfloor \mathring{c}}\kappa_{c;t}-1} \prod_{i=1}^{\mathring{d}} \frac{A(\Theta_{ic})^{{}^{\lfloor \mathring{c}}\nu_{ic;t}} \exp\left[\left\langle {}^{\lfloor \mathring{c}}V_{ic;t}, C_{ic}(\Theta_{ic})\right\rangle\right]}{\underbrace{\int A(\Theta_{ic})^{{}^{\lfloor \mathring{c}}\nu_{ic;t}} \exp\left[\left\langle {}^{\lfloor \mathring{c}}V_{ic;t}, C_{ic}(\Theta_{ic})\right\rangle\right] d\Theta_{ic}}_{\mathcal{I}\left({}^{\lfloor \mathring{c}}V_{ic;t},\, {}^{\lfloor \mathring{c}}\nu_{ic;t}\right)}}. \qquad (6.103)$$

Algorithm 6.6.4 (Selection and merging of a pair of components in exponential family)
Initial mode

- *Perform estimation of a mixture with a sufficient amount of components* $\mathring{c}$ *so that statistics describing them* ${}^{\lfloor \mathring{c}}\kappa_t,\ {}^{\lfloor \mathring{c}}\nu_{ic;t},\ {}^{\lfloor \mathring{c}}V_{ic;t},\ t \in \{0, \mathring{t}\},\ c \in c^*,\ i = 1,\ldots,\mathring{d}$, *are at disposal.*
- *Set the minimum value of the KL distance* $\underline{\mathcal{D}} = +\infty$ *and initialize the indices* $(\underline{c}, \underline{\tilde{c}}) = (0,0)$ *of the component pair to be merged.*

Evaluation mode

1. *Select the pair whose merging causes the smallest change of the KL distance as follows.*

For $c = 1,\ldots,\mathring{c}-1$
  For $\tilde{c} = c+1,\ldots,\mathring{c}$
    Set the indicator of the common structure $cs = 0$.
  For $i = 1,\ldots,\mathring{d}$
    Set $cs = -1$ *and break the cycle over* i *if the structures of* Θ_{ic} *and* $\Theta_{i\tilde{c}}$ *differ.*
    Create the statistics $\tilde{\nu}_{i;\mathring{t}} = 0.5(\nu_{ic;\mathring{t}} + \nu_{i\tilde{c};\mathring{t}}),\quad \tilde{V}_{i;\mathring{t}} = 0.5(V_{ic;\mathring{t}} + V_{i\tilde{c};\mathring{t}})$.
  end *of the cycle over* i
    Do if $cs = 0$
    Complete creating the statistics for the distance measuring $\tilde{\kappa}_{\mathring{t}} = 0.5(\kappa_{c;\mathring{t}} + \kappa_{\tilde{c};\mathring{t}})$.
    Evaluate the common part of the KL distance

$\tilde{\mathcal{D}} = \mathcal{D}\left(Di_{\alpha_c,\alpha_{\tilde{c}}}(\kappa_{c;\mathring{t}},\kappa_{\tilde{c};\mathring{t}})||Di_{\alpha_c,\alpha_{\tilde{c}}}(\tilde{\kappa}_{\mathring{t}},\tilde{\kappa}_{\mathring{t}})\right).$

Complete the evaluation of the KL distance

For $i = 1,\ldots,\mathring{d}$

$$\tilde{\mathcal{D}} = \tilde{\mathcal{D}} + \mathcal{D}\left(f(\Theta_{ic}|V_{ic;\mathring{t}},\nu_{ic;\mathring{t}})||f(\Theta_{ic}|\tilde{V}_{i;\mathring{t}},\tilde{\nu}_{i;\mathring{t}})\right) + \mathcal{D}\left(f(\Theta_{i\tilde{c}}|V_{i\tilde{c};\mathring{t}},\nu_{i\tilde{c};\mathring{t}})||f(\Theta_{ic}|\tilde{V}_{i;\mathring{t}},\tilde{\nu}_{i;\mathring{t}})\right).$$

end of the cycle over i

If $\tilde{\mathcal{D}} < \underline{\mathcal{D}}$

Set $\underline{\mathcal{D}} = \tilde{\mathcal{D}},\ (\underline{c},\underline{\tilde{c}}) = (c,\tilde{c})$

and store values of the trial statistics $\tilde{\kappa}_{\mathring{t}},\ \tilde{\nu}_{i;\mathring{t}},\ \tilde{V}_{i;\mathring{t}},\ \forall i \in i^*$.

end of the test on the value $\tilde{\mathcal{D}}$ *of the KL distance.*

end of the condition $cs = 0$

end of the cycle over $\tilde{c}$

end of the cycle over c

2. *Stop and announce that no merging is possible if* $(\underline{c},\underline{\tilde{c}}) = (0,0)$.
3. *Rename components so that* $(\mathring{c}-1,\mathring{c}) = (\underline{c},\underline{\tilde{c}})$.
4. *Finish the merging of the best components found and apply the same operation to the prior pdf*

$$^{\lfloor \mathring{c}-1}\kappa_{\mathring{c}-1;\mathring{t}} \equiv 2\tilde{\kappa}_{\mathring{t}} - 1 \equiv {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}} - 1,\ {}^{\lfloor \mathring{c}-1}\kappa_{\mathring{c}-1;0} \equiv {}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;0} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};0} - 1 \quad (6.104)$$

For $i = 1,\ldots,\mathring{d}$

$$^{\lfloor \mathring{c}-1}\nu_{i(\mathring{c}-1);\mathring{t}} \equiv 2\tilde{\nu}_{i;\mathring{t}} \equiv {}^{\lfloor \mathring{c}}\nu_{i(\mathring{c}-1);\mathring{t}} + {}^{\lfloor \mathring{c}}\nu_{i\mathring{c};\mathring{t}}$$

$$^{\lfloor \mathring{c}-1}V_{i(\mathring{c}-1);\mathring{t}} \equiv 2\tilde{V}_{i;\mathring{t}} \equiv {}^{\lfloor \mathring{c}}V_{i(\mathring{c}-1);\mathring{t}} + {}^{\lfloor \mathring{c}}V_{i\mathring{c};\mathring{t}}$$

$$^{\lfloor \mathring{c}-1}\nu_{i(\mathring{c}-1);0} \equiv {}^{\lfloor \mathring{c}}\nu_{i(\mathring{c}-1);0} + {}^{\lfloor \mathring{c}}\nu_{i\mathring{c};0},\ \ {}^{\lfloor \mathring{c}-1}V_{i(\mathring{c}-1);0} \equiv {}^{\lfloor \mathring{c}}V_{i(\mathring{c}-1);0} + {}^{\lfloor \mathring{c}}V_{i\mathring{c};0}.$$

end of the cycle over i

The remaining statistics are unchanged by the considered merging. It completes the definition of the conjugate prior and posterior pdfs on the mixture with $\mathring{c}-1$ *components.*

Remark(s) 6.6.2

Merging can be performed at factor level. It is done in an independent paragraph. Factor merging could reveal common factors that reflect a common physical nature of some modelled relationships.

Merging of a group of components

We could make trial merging of a pair components as described in previous paragraph then apply flattening operation and repeat estimation. The v-likelihood obtained as byproduct would indicate success or failure of this merging. This way is extremely costly for the considered large data sets. Thus, it makes sense to merge several pairs of components before re-estimation. It calls for a rule determining how many components should be merged. In other words, the estimation results obtained for $\mathring{c}$ components have to be used for prediction of the value of the v-likelihood after group merging. This problem is addressed here.

The merging provides both ${}^{\lfloor \mathring{c}-1}f(\Theta|d(\mathring{t}))$ and ${}^{\lfloor \mathring{c}-1}f(\Theta)$. We also know ${}^{\lfloor \mathring{c}}f(\Theta|d(\mathring{t}))$ and ${}^{\lfloor \mathring{c}}f(\Theta)$. Parameter estimates of both mixtures, labelled by $\tilde{c} \in \{\mathring{c}-1,\mathring{c}\}$, obey approximately the Bayes rule

$$\overbrace{{}^{\lfloor \tilde{c}}f(\Theta|d(\mathring{t}))}^{\text{known}} = \frac{{}^{\lfloor \tilde{c}}f(d(\mathring{t})|\Theta)\ \overbrace{{}^{\lfloor \tilde{c}}f(\Theta)}^{\text{known}}}{{}^{\lfloor \tilde{c}}f(d(\mathring{t}))}. \quad (6.105)$$

For the mixture with $\mathring{c}$ terms, we even know the v-likelihood ${}^{\lfloor \mathring{c}}f(d(\mathring{t}))$. Let us consider pdfs describing parameters of both mixtures on the subset ${}^{\lfloor \mathring{c}}\Theta^*$ determined by the merging conditions

$$\alpha_{\mathring{c}-1} = \alpha_{\mathring{c}}, \ \Theta_{i(\mathring{c}-1)} = \Theta_{i\mathring{c}}, \ i = 1, \ldots, \mathring{d}. \tag{6.106}$$

On this subset, ${}^{\lfloor \mathring{c}}f(d(\mathring{t})|\Theta) = {}^{\lfloor \mathring{c}-1}f(d(\mathring{t})|\Theta)$ and this factor cancels in the ratio of expressions (6.105)

$$\begin{aligned}
\frac{{}^{\lfloor \mathring{c}-1}f(\Theta|d(\mathring{t}))}{{}^{\lfloor \mathring{c}}f(\Theta|d(\mathring{t}))} &= \frac{\mathcal{B}\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{t}}\right)}{\mathcal{B}\left({}^{\lfloor \mathring{c}-1}\kappa_{\mathring{t}}\right)} \prod_{i=1}^{\mathring{d}} \frac{{}^{\lfloor \mathring{c}-1}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))}{{}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,{}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(\mathring{t}))} = \\
&\underbrace{=}_{(6.102),(6.4),(6.106)} \frac{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}}\right)\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}}\right)\Gamma\left(\left(\sum_{c=1}^{\mathring{c}} {}^{\lfloor \mathring{c}}\kappa_{c;\mathring{t}}\right)-1\right)}{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}} - 1\right)\Gamma\left(\sum_{c=1}^{\mathring{c}} {}^{\lfloor \mathring{c}}\kappa_{c;\mathring{t}}\right)} \\
&\prod_{i=1}^{\mathring{d}} \frac{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,d\Theta_{i(\mathring{c}-1)} \int {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}|d(\mathring{t}))\,d\Theta_{i\mathring{c}}}{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,{}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,d\Theta_{i(\mathring{c}-1)}} = \\
&\underbrace{=}_{(6.105),(6.102),(6.106)} \frac{{}^{\lfloor \mathring{c}}f(d(\mathring{t}))}{{}^{\lfloor \mathring{c}-1}f(d(\mathring{t}))} \frac{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;0}\right)\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c};0}\right)\Gamma\left(\left(\sum_{c=1}^{\mathring{c}} {}^{\lfloor \mathring{c}}\kappa_{c;0}\right)-1\right)}{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;0} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};0} - 1\right)\Gamma\left(\sum_{c=1}^{\mathring{c}} {}^{\lfloor \mathring{c}}\kappa_{c;0}\right)} \\
&\prod_{i=1}^{\mathring{d}} \frac{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(0))\,d\Theta_{i(\mathring{c}-1)} \int {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}|d(0))\,d\Theta_{i\mathring{c}}}{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(0))\,{}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}} = \Theta_{i(\mathring{c}-1)}|d(0))\,d\Theta_{i(\mathring{c}-1)}}.
\end{aligned} \tag{6.107}$$

The last equality in (6.107) gives the identity that allows us to evaluate the v-likelihood ${}^{\lfloor \mathring{c}-1}f(d(\mathring{t}))$ without repeating the whole estimation. Using the formula $\Gamma(x+1) = x\Gamma(x)$, we can write it explicitly

$$\begin{aligned}
&{}^{\lfloor \mathring{c}-1}f(d(\mathring{t})) = {}^{\lfloor \mathring{c}}f(d(\mathring{t})) \times \\
&\times \frac{\dfrac{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;0}\right)\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c};0}\right)\left(\left(\sum_{c=1}^{\mathring{c}} {}^{\lfloor \mathring{c}}\kappa_{c;0}\right)-1\right)}{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;0} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};0} - 1\right)} \prod_{i=1}^{\mathring{d}} \dfrac{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(0))\,d\Theta_{i(\mathring{c}-1)} \int {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}|d(0))\,d\Theta_{i\mathring{c}}}{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(0))\,{}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}=\Theta_{i(\mathring{c}-1)}|d(0))\,d\Theta_{i(\mathring{c}-1)}}}{\dfrac{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}} + {}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}} - 1\right)}{\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c}-1;\mathring{t}}\right)\Gamma\left({}^{\lfloor \mathring{c}}\kappa_{\mathring{c};\mathring{t}}\right)\left(\left(\sum_{c=1}^{\mathring{c}} {}^{\lfloor \mathring{c}}\kappa_{c;\mathring{t}}\right)-1\right)} \prod_{i=1}^{\mathring{d}} \dfrac{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,d\Theta_{i(\mathring{c}-1)} \int {}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}|d(\mathring{t}))\,d\Theta_{i\mathring{c}}}{\int {}^{\lfloor \mathring{c}}f(\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,{}^{\lfloor \mathring{c}}f(\Theta_{i\mathring{c}}=\Theta_{i(\mathring{c}-1)}|d(\mathring{t}))\,d\Theta_{i(\mathring{c}-1)}}}.
\end{aligned} \tag{6.108}$$

We can judge whether trial merging of a component pair increases the v-likelihood. Thus, we have chance to merge a group components without costly re-estimation.

We write the resulting algorithm for practically significant case of components in the exponential family. For this model, the evaluation (6.108) gets the form written in terms of the statistics of the original mixture with the superscript $\mathring{c}$ drop.

$$\begin{aligned}
{}^{\lfloor \mathring{c}-1}f(d(\mathring{t})) &= {}^{\lfloor \mathring{c}}f(d(\mathring{t})) \frac{\Gamma\left(\kappa_{\mathring{c}-1;\mathring{t}} + \kappa_{\mathring{c};\mathring{t}} - 1\right)\Gamma\left(\kappa_{\mathring{c}-1;0}\right)\Gamma\left(\kappa_{\mathring{c};0}\right)\left(\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;0}\right)-1\right)}{\Gamma\left(\kappa_{\mathring{c}-1;0} + \kappa_{\mathring{c};0} - 1\right)\Gamma\left(\kappa_{\mathring{c}-1;\mathring{t}}\right)\Gamma\left(\kappa_{\mathring{c};\mathring{t}}\right)\left(\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;\mathring{t}}\right)-1\right)} \\
&\prod_{i=1}^{\mathring{d}} \frac{\mathcal{I}(V_{i(\mathring{c}-1);\mathring{t}} + V_{i\mathring{c};\mathring{t}}, \nu_{i(\mathring{c}-1);\mathring{t}} + \nu_{i\mathring{c};\mathring{t}})}{\mathcal{I}(V_{i(\mathring{c}-1);0} + V_{i\mathring{c};0}, \nu_{i(\mathring{c}-1);0} + \nu_{i\mathring{c};0})} \frac{\mathcal{I}(V_{i(\mathring{c}-1);0}, \nu_{i(\mathring{c}-1);0})\mathcal{I}(V_{i\mathring{c};0}, \nu_{i\mathring{c};0})}{\mathcal{I}(V_{i(\mathring{c}-1);\mathring{t}}, \nu_{i(\mathring{c}-1);\mathring{t}})\mathcal{I}(V_{i\mathring{c};\mathring{t}}, \nu_{i\mathring{c};\mathring{t}})}.
\end{aligned} \tag{6.109}$$

The last formula allows us to enrich Algorithm 6.6.4 so that promising candidates are gradually merged up to the moment when the v-likelihood is predicted to drop.

Algorithm 6.6.5 (Merging of a group of components in exponential family)
Initial mode

- *Perform estimation of a mixture with components in exponential family and conjugate prior pdfs. Let us do that for over-estimates amount of components* $\mathring{c} \geq 2$. *Thus, the statistics* ${}^{\lfloor \mathring{c}}\kappa_t$, ${}^{\lfloor \mathring{c}}\nu_{ic;t}$, ${}^{\lfloor \mathring{c}}V_{ic;t}$, $t \in \{0, \mathring{t}\}$, $c = 1, \ldots, \mathring{c}$, $i = 1, \ldots, \mathring{d}$, *are disposal.*

- *Set pointers $c = 1, \tilde{c} = 2$ to trial components to be merged.*

Evaluation mode

Set the indicator of the common structure $cs = 0$.

For $i = 1, \ldots, \mathring{d}$

Set $cs = -1$ and break the cycle over i if the structures of Θ_{ic} and $\Theta_{i\tilde{c}}$ differ.

end of the cycle over i

Do if $cs = 0$

Evaluate the common part of the trial merger

$\tilde{\kappa}_{\mathring{t}} = \kappa_{c;\mathring{t}} + \kappa_{\tilde{c};\mathring{t}} - 1, \ \tilde{\kappa}_0 = \kappa_{c;0} + \kappa_{\tilde{c};0} - 1.$

Evaluate and store the factor-related parts of trial merger

For $i = 1, \ldots, \mathring{d}$

$\tilde{\nu}_{i;\mathring{t}} = \nu_{ic;\mathring{t}} + \nu_{i\tilde{c};\mathring{t}}, \quad \tilde{V}_{i;\mathring{t}} = V_{ic;\mathring{t}} + V_{i\tilde{c};\mathring{t}}.$

end of the cycle over i

Evaluate the change $\tilde{l}$ of log-v-likelihood expected after the definite merging

$$\tilde{l} = + \left\{ -\ln\left(\Gamma(\kappa_{c;\mathring{t}})\right) - \ln\left(\Gamma(\kappa_{\tilde{c};\mathring{t}})\right) + \ln\left(\Gamma(\tilde{\kappa}_{\mathring{t}})\right) - \ln\left(\left(\sum_{c=1}^{\mathring{c}-1} \kappa_{c;\mathring{t}}\right) - 1\right)\right\} -$$
$$- \left\{ -\ln\left(\Gamma(\kappa_{c;0})\right) - \ln\left(\Gamma(\kappa_{\tilde{c};0})\right) + \ln\left(\Gamma(\tilde{\kappa}_0)\right) - \ln\left(\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;0}\right) - 1\right)\right\}.$$

For $i = 1, \ldots, \mathring{d}$

(factor parts)

$$\tilde{l} = \tilde{l} + \left\{ \ln(\mathcal{I}(\tilde{V}_{i;\mathring{t}}, \tilde{\nu}_{i;\mathring{t}})) - \ln(\mathcal{I}(V_{ic;\mathring{t}}, \nu_{ic;\mathring{t}})) - \ln(\mathcal{I}(V_{i\tilde{c};\mathring{t}}, \nu_{i\tilde{c};\mathring{t}})) \right\} -$$
$$- \left\{ \ln(\mathcal{I}(\tilde{V}_{i;0}, \tilde{\nu}_{i;0})) - \ln(\mathcal{I}(V_{ic;0}, \nu_{ic;0})) - \ln(\mathcal{I}(V_{i\tilde{c};0}, \nu_{i\tilde{c};0})) \right\}.$$

end of the cycle over i

end of the condition $cs = 0$

Do if $\tilde{l} \leq 0$ or $cs < 0$

Set $\tilde{c} = \tilde{c} + 1$.

Go to the beginning of Evaluation mode *if $\tilde{c} \leq \mathring{c}$. Otherwise continue.*

Set $c = c + 1$ and $\tilde{c} = c + 1$.

Go to the beginning of Evaluation mode *if $c < \mathring{c}$. Otherwise stop.*

else replace statistics related to the component c by $\tilde{\kappa}_{\mathring{t}}, \ \tilde{\kappa}_0, \ \left\{\tilde{V}_{i;\mathring{t}}, \ \tilde{V}_{i;0}, \ \tilde{\nu}_{i;\mathring{t}}, \ \tilde{\nu}_{i;0}\right\}_{i=1}^{\mathring{d}}$.

Swap the components $\mathring{c}$ and $\tilde{c}$.

Decrease $\mathring{c} = \mathring{c} - 1$, i.e. omit the component $\tilde{c}$.

Set $\tilde{c} = c + 1$ if $\tilde{c} > \mathring{c}$.

end of the test on improvement of v-likelihood and of $cs < 0$

Stop if $\mathring{c} = 1$. Otherwise go to the beginning of Evaluation mode.

Factor-based merging

Here we merge superfluous factors by inspecting presence of common factors. It decreases computational load, can reveal relationships of a common nature and – if a pair of component will consist of common factors only – to reduce the number of components. We proceed similarly as with the merging of components.

Candidates for merging The pdf on parameters has the form (6.3) $f(\Theta) \equiv Di_{\alpha}(\kappa) \prod_{c \in c^*} \prod_{i=1}^{\mathring{d}} f(\Theta_{ic})$. For presentation simplicity, we assume that the order items $d_{i;t}$ in the data record d_t is common for all components.

The pdf ${}^{\lfloor ic\tilde{c}}f(\Theta)$ describing the parameter estimate – under hypothesis that estimates of the i-th factor in components $c \neq \tilde{c} \in c^*$ equal – has the form

$$ {}^{\lfloor ic\tilde{c}}f(\Theta) \equiv f(\Theta) \frac{\tilde{f}(\Theta_{ic})\tilde{f}(\Theta_{i\tilde{c}})}{f(\Theta_{ic})f(\Theta_{i\tilde{c}})}, \text{ where} \tag{6.110} $$

$f(\Theta)$ is the original pdf whose factors should be merged and $f(\Theta_{ic})$, $f(\Theta_{i\tilde{c}}$ are its marginal pdfs. The pdf $\tilde{f}(\Theta_{ic})$ is the constructed common factor, i.e. $\tilde{f}(\Theta_{ic}) \equiv \tilde{f}(\Theta_{i\tilde{c}} = \Theta_{ic})\ \forall \Theta_{ic} \in \Theta^*_{ic}$. Creation of a common factor makes sense only when the structures of Θ_{ic} and $\Theta_{i\tilde{c}}$ are the same.

The KL distance of the mixture with the common factor (6.110) to the original mixture is (with our usual abuse of notation)

$$ {}^{\lfloor ic\tilde{c}}\mathcal{D} \equiv \mathcal{D}\left(\tilde{f}(\Theta_{ic})||f(\Theta_{ic})\right) + \mathcal{D}\left(\tilde{f}(\Theta_{ic})||f(\Theta_{i\tilde{c}})\right). \tag{6.111} $$

It is minimized by the geometric mean of the inspected factor estimates

$$ \tilde{f}(\Theta_{ic}) \propto \sqrt{f(\Theta_{ic})f(\Theta_{i\tilde{c}} = \Theta_{ic})}. \tag{6.112} $$

Thus, for all $i = 1, \ldots, \mathring{d}$, we can independently evaluate distances ${}^{\lfloor ic\tilde{c}}\mathcal{D}$, $c \in c^*$, $\tilde{c} > c$, for the merger (6.112). The restriction to the upper triangular part only exploits the obvious symmetry of matrices ${}^{\lfloor i\cdot\cdot}\mathcal{D}$, $i = 1, \ldots, \mathring{d}$. The prospective common factors should cause the smallest change of the original parameter estimate. This guides us in selecting the prospective merger.

Algorithm 6.6.6 (Selection and merging of pairs of factors)
Initial mode

- *Estimate the mixture with a sufficient amount of components $\mathring{c}$ so that the pdfs ${}^{\lfloor \mathring{c}}f(\Theta_{ic}|d(t))$ and statistics ${}^{\lfloor \mathring{c}}\kappa_t$, $t \in \{0, \mathring{t}\}$, $c \in c^*$, $i = 1, \ldots, \mathring{d}$, are at disposal.*

Evaluation mode

For $i = 1, \ldots, \mathring{d}$

Set the smallest value of the KL distance $\underline{\mathcal{D}} = +\infty$ and initiate indices of the component pair $(\underline{c}, \underline{\tilde{c}}) = (0, 0)$ in which the merged factors are.

For $c = 1, \ldots, \mathring{c} - 1$

For $\tilde{c} = c + 1, \ldots, \mathring{c}$

Go to the end of the cycle over $\tilde{c}$ if the structures of Θ_{ic} and $\Theta_{i\tilde{c}}$ differ.

Create the trial merger (6.112)

Evaluate the KL distance $\tilde{\mathcal{D}} \equiv {}^{\lfloor ic\tilde{c}}\mathcal{D}$ according to (6.111).

If $\tilde{\mathcal{D}} < \underline{\mathcal{D}}$

Set $\underline{\mathcal{D}} = \tilde{\mathcal{D}}$, $(\underline{c}, \underline{\tilde{c}}) = (c, \tilde{c})$ and store the trial merger.

end of the test on the value of the KL distance

end of the cycle over $\tilde{c}$

end of the cycle over c

Go to the end of cycle over i if $(\underline{c}, \underline{\tilde{c}}) = (0, 0)$.

Create merger of prior pdfs corresponding to factors with indices $i\underline{c}$ and $i\underline{\tilde{c}}$.

Replace prior and posterior estimates of factors with indices $i\underline{c}$, $i\underline{\tilde{c}}$ by the trial merger.

end of the cycle over i

Merging of a group of factors We could make trial merging of a pair factors as described in previous paragraph then apply flattening operation and repeat the estimation. The v-likelihood obtained as its byproduct would indicate success or failure of this merging. This evaluation is extremely costly for the considered large data sets. Thus, it makes sense to perform several merging steps at one shot. It needs, however, to find a rule when to stop the merging. Similarly, as in the component-based merging the values of the v-likelihood values expected after it are predicted.

The proposed merging provides both ${}^{\lfloor ic\tilde{c}}f(\Theta|d(\mathring{t}))$ and ${}^{\lfloor ic\tilde{c}}f(\Theta)$. We also know $f(\Theta|d(\mathring{t}))$, $f(\Theta)$ and, moreover, the original v-likelihood $f(d(\mathring{t}))$. We know that for values $\Theta_{ic} = \Theta_{i\tilde{c}}$ the corresponding likelihood functions coincide. Assuming the ratio $f(\Theta|d(\mathring{t}))/\,{}^{\lfloor ic\tilde{c}}f(\Theta|d(\mathring{t}))$ at such points, using the Bayes rule and exploiting the form of the merger, we get

$$\begin{aligned}\frac{f(\Theta|d(\mathring{t}))}{{}^{\lfloor ic\tilde{c}}f(\Theta|d(\mathring{t}))} &\equiv \frac{\int f(\Theta_{ic}|d(\mathring{t}))f(\Theta_{i\tilde{c}}=\Theta_{ic}|d(\mathring{t}))\,d\Theta_{ic}}{\int f(\Theta_{ic}|d(\mathring{t}))\,d\Theta_{ic}\int f(\Theta_{i\tilde{c}}|d(\mathring{t}))\,d\Theta_{i\tilde{c}}} = \\ &= \frac{{}^{\lfloor ic\tilde{c}}f(d(\mathring{t}))}{f(d(\mathring{t}))}\frac{\int f(\Theta_{ic})f(\Theta_{i\tilde{c}}=\Theta_{ic})\,d\Theta_{ic}}{\int f(\Theta_{ic})\,d\Theta_{ic}\int f(\Theta_{i\tilde{c}})\,d\Theta_{i\tilde{c}}}.\end{aligned}$$

It gives the rule how to find the v-likelihood ${}^{\lfloor ic\tilde{c}}f(d(\mathring{t}))$ after merging

$$ {}^{\lfloor ic\tilde{c}}f(d(\mathring{t})) = f\left(d(\mathring{t})\right)\frac{\int f(\Theta_{ic}|d(\mathring{t}))\,d\Theta_{ic}\int f(\Theta_{i\tilde{c}}|d(\mathring{t}))\,d\Theta_{i\tilde{c}}}{\int f(\Theta_{ic}|d(\mathring{t}))f(\Theta_{i\tilde{c}}=\Theta_{ic}|d(\mathring{t}))\,d\Theta_{ic}}\frac{\int f(\Theta_{ic})f(\Theta_{i\tilde{c}}=\Theta_{ic})\,d\Theta_{ic}}{\int f(\Theta_{ic})\,d\Theta_{ic}\int f(\Theta_{i\tilde{c}})\,d\Theta_{i\tilde{c}}}. \tag{6.113}$$

The constructed estimate gives the possibility to make a group merging while the v-likelihood is improving. Then, it remains to merge components. Obviously, the components $c, \tilde{c}$ with all factors common can be taken as a single one. It makes no sense to distinguish them and their weights. Thus, the statistics of their weights are simply summed. This leads to the following algorithm.

Algorithm 6.6.7 (Merging of a group of factors)
Initial mode

- *Perform estimation of a mixture with a sufficient amount of components.*
- *Initialize the list of factors with rows $\rho = (i, c, \tilde{c}) \equiv i$-th factor is common for components $c, \tilde{c}$. Often, ρ is selected as the empty one.*

Evaluation mode

For $i = 1, \ldots, \mathring{d}$
 For $c = 1, \ldots, \mathring{c} - 1$
  For $\tilde{c} = c + 1, \ldots, \mathring{c}$
   Go to the end of the cycle over $\tilde{c}$ if the structures of Θ_{ic} and $\Theta_{i\tilde{c}}$ differ.
   Create and store the trial merger (6.112).
   Create the trial merger (6.112) of prior pdfs.
   Evaluate the factor-related change (6.113) of the v-likelihood predicted after merging.
   Go to the end of cycle over $\tilde{c}$ if no improvement is predicted.
   Replace prior and posterior factor estimates with indices $i\underline{c}$, $i\underline{\tilde{c}}$ by the trial merger.
   Extend the list of common factors by $\rho = [\rho; (i, \underline{c}, \underline{\tilde{c}})]$.
  end *of the cycle over $\tilde{c}$*
 end *of the cycle over c*
end *of the cycle over i*
 For $c = 1, \ldots, \mathring{c} - 1$
  For $\tilde{c} = c + 1, \ldots, \mathring{c}$
   Set $\kappa_{\tilde{c};\mathring{t}} = \kappa_{\tilde{c};\mathring{t}} + \kappa_{c;\mathring{t}}$ $\kappa_{\tilde{c};0} = \kappa_{\tilde{c};0} + \kappa_{c;0}$ and

cancel the component c if the components consist of common factors only.

end of the cycle over $\tilde{c}$

end of the cycle over c

Problem 6.6.2 (Influence of component weights) *Dirichlet factor influences final v-likelihood whenever components are merged. This fact is not taken into account in Algorithm 6.6.7 as it is not clear how to do that effectively. The modification is worth of inspecting.*

The above algorithm is now written for factors in exponential family. It is enriched so that common factors in several components can be recognized.

Algorithm 6.6.8 (Systematic merging of factors in exponential family)
Initial mode

- *Perform estimation of a mixture with factors in exponential family. The mixture estimate is described by the collection of statistics*

$$\{V_{ic;t}, \nu_{ic;t}\}_{c\in c^*, i=1,\ldots,\mathring{d}, t\in\{0,\mathring{t}\}}.$$

  The factors with the common i are supposed to describe the same entry of $d_{i;t}$ irrespective of the component number.

- *Initialize the list with rows $\rho = (i, c, \tilde{c}) \equiv i$-th factor is common for components $c, \tilde{c}$. Usually, ρ is initialized as the empty one.*

- *Evaluate the individual normalization factors* $\ln(\mathcal{I}(V_{ic;t}, \nu_{ic:t}))$, $\forall c \in c^*, i = 1, \ldots, \mathring{d}, t \in \{0, \mathring{t}\}$.

Evaluation mode

For $i = 1, \ldots, \mathring{d}$

Set pointers $c = 1, \tilde{c} = 2$ to trial components to be merged.

Test of the common structure

Set the indicator of the common structure $cs = 0$.

Set $cs = -1$ if the structures of Θ_{ic} and $\Theta_{i\tilde{c}}$ differ.

Do if $cs = 0$

Create the trial merger

$\tilde{V}_{i;\mathring{t}} = V_{ic;\mathring{t}} + V_{i\tilde{c};\mathring{t}},\ \tilde{V}_{i;0} = V_{ic;0} + V_{i\tilde{c};0}$

$\tilde{\nu}_{i;\mathring{t}} = \nu_{ic;\mathring{t}} + \nu_{i\tilde{c};\mathring{t}},\ \tilde{\nu}_{i;0} = \nu_{ic;0} + \nu_{i\tilde{c};0}.$

Evaluate increment $\tilde{l}$ of the log-v-likelihood

$$\tilde{l} = +\left\{\ln(\mathcal{I}(\tilde{V}_{i;\mathring{t}}, \tilde{\nu}_{i;\mathring{t}})) - \ln(\mathcal{I}(V_{ic;\mathring{t}}, \nu_{ic;\mathring{t}})) - \ln(\mathcal{I}(V_{i\tilde{c};\mathring{t}}, \nu_{i\tilde{c};\mathring{t}}))\right\}$$
$$-\left\{\ln(\mathcal{I}(\tilde{V}_{i;0}, \tilde{\nu}_{i;0})) - \ln(\mathcal{I}(V_{ic;0}, \nu_{ic;0})) - \ln(\mathcal{I}(V_{i\tilde{c};0}, \nu_{i\tilde{c};0}))\right\}.$$

end of the test on $cs = 0$

Do if $\tilde{l} \leq 0$ or $cs < 0$

Set $\tilde{c} = \tilde{c} + 1$.

Go to the Test of the common structure *if $\tilde{c} \leq \mathring{c}$. Otherwise continue.*

Set $c = c + 1$ and $\tilde{c} = c + 1$.

Go to the beginning of Test of the common structure *if $c < \mathring{c}$.*

Otherwise go to the end of cycle over i.

else replace prior and posterior estimates of factors with indices ic and $i\tilde{c}$

by the trial merger.

Extend the list of common factors by $\rho = [\rho; (i, c, \tilde{c})]$.

end of the test on improvement of v*-likelihood and of* $cs < 0$

end *of the cycle over* i

Merging of components

For $c = 1, \ldots, \mathring{c} - 1$

For $\tilde{c} = c + 1, \ldots, \mathring{c}$

Set $\kappa_{\tilde{c};\mathring{t}} = \kappa_{\tilde{c};\mathring{t}} + \kappa_{c;\mathring{t}}, \quad \kappa_{\tilde{c};0} = \kappa_{\tilde{c};0} + \kappa_{c;0}$ *and*

cancel the component c *if the components consist of common factors only.*

end *of the cycle over* $\tilde{c}$

end *of the cycle over* c

Component cancelling

The technique used in connection with merging of components allows us predict the influence of cancelling on the v-likelihood without re-estimation. Consequently all cancelling possibilities can be checked with a relatively small computational load. The explicit solution is presented here. It needs a bit care as for $\alpha_{\mathring{c}} = 0$, the prior and posterior pdfs are zero. The chain rule $f(a|b) = f(a,b)/f(b)$ and properties of the Dirichlet pdf, the formula for marginal pdf (10.4) in Proposition 10.1.1, however, imply that for $f({}^{\lfloor\mathring{c}}\alpha) = Di_{{}^{\lfloor\mathring{c}}\alpha}\left({}^{\lfloor\mathring{c}}\kappa\right)$ the conditional pdf

$$ {}^{\lfloor\mathring{c}}f(\alpha|\alpha_{\mathring{c}} = 0) = Di_{[\alpha_1,\ldots,\alpha_{\mathring{c}-1}]}\left(\left[{}^{\lfloor\mathring{c}}\kappa_1, \ldots, {}^{\lfloor\mathring{c}}\kappa_{\mathring{c}-1}\right]\right) \equiv {}^{\lfloor\mathring{c}-1}f(\alpha) \equiv Di_{\alpha}\left({}^{\lfloor\mathring{c}-1}\kappa\right). $$

The posterior pdf on parameters before and after cancelling the $\mathring{c}$-th component are related to their prior pdf by the Bayes rule

$$ \begin{aligned} {}^{\lfloor\mathring{c}}f(\Theta|d(\mathring{t})) &= Di_{\alpha}({}^{\lfloor\mathring{c}}\kappa_{\mathring{t}}) \prod_{c=1}^{\mathring{c}} \frac{{}^{\lfloor\mathring{c}}f(\Theta_c|d(\mathring{t}))}{{}^{\lfloor\mathring{c}}\mathcal{I}_c(d(\mathring{t}))} = \frac{{}^{\lfloor\mathring{c}}f(d(\mathring{t})|\Theta)}{{}^{\lfloor\mathring{c}}f(d(\mathring{t}))} Di_{\alpha}({}^{\lfloor\mathring{c}}\kappa_0) \prod_{c=1}^{\mathring{c}} \frac{{}^{\lfloor\mathring{c}}f(\Theta_c|d(0))}{{}^{\lfloor\mathring{c}}\mathcal{I}_c(d(0))} \\ {}^{\lfloor\mathring{c}-1}f(\Theta|d(\mathring{t})) &= Di_{\alpha}({}^{\lfloor\mathring{c}-1}\kappa_{\mathring{t}}) \left({}^{\lfloor\mathring{c}-1}\kappa_{\mathring{t}}\right) \prod_{c=1}^{\mathring{c}-1} \frac{{}^{\lfloor\mathring{c}-1}f(\Theta_c|d(\mathring{t}))}{{}^{\lfloor\mathring{c}-1}\mathcal{I}_c(d(\mathring{t}))} = \\ &= \frac{{}^{\lfloor\mathring{c}-1}f(d(\mathring{t})|\Theta)}{{}^{\lfloor\mathring{c}-1}f(d(\mathring{t}))} Di_{\alpha}\left({}^{\lfloor\mathring{c}-1}\kappa_0\right) \prod_{c=1}^{\mathring{c}-1} \frac{{}^{\lfloor\mathring{c}}f(\Theta_c|d(0))}{{}^{\lfloor\mathring{c}}\mathcal{I}_c(d(0))}. \end{aligned} $$

As usual, ${}^{\lfloor\mathring{c}}\mathcal{I}_c(d(t)),\ t \in \{0, \mathring{t}\}$ denote normalizing integrals of the pdfs describing parameters Θ_c conditioned on $d(t)$ for the mixture with $\mathring{c}$ components.

Comparing to the component merging, the embedding of the model with smaller amount of components is more peculiar as the most obvious option $\alpha_{\mathring{c}} = 0$ gives zero values of both prior and posterior pdfs. Instead of this choice, we assume that there are parameter values $\Theta_c,\ c = 1, \ldots, \mathring{c}$ such that

$$ f\left(d_t|d(t-1), {}^{\lfloor\mathring{c}}\Theta_c\right) = g(d(t)) \text{ for } c = 1, \ldots, \mathring{c}, \text{ and} \tag{6.114} $$

some function $g(d(t))$. For parameter values meeting (6.114), the likelihood values ${}^{\lfloor\mathring{c}}f\left(d(\mathring{t})|\,{}^{\lfloor\mathring{c}}\Theta\right)$,, ${}^{\lfloor\mathring{c}-1}f\left(d(\mathring{t})|\,{}^{\lfloor\mathring{c}}\Theta\right)$ assigned to both parameterized models coincide. Taking the ratios of the posterior pdfs of both mixture models at such parameters, we get

$$ \begin{aligned} {}^{\lfloor\mathring{c}-1}f(d(\mathring{t}))\alpha_{\mathring{c}}^{{}^{\lfloor\mathring{c}}\kappa_{\mathring{c};\mathring{t}} - {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};0}} (1-\alpha_{\mathring{c}})^{\sum_{c=1}^{\mathring{c}-1} {}^{\lfloor\mathring{c}}\kappa_{c;\mathring{t}} - {}^{\lfloor\mathring{c}}\kappa_{c;0}} &= {}^{\lfloor\mathring{c}}f(d(\mathring{t})) \times \\ \times \quad \frac{{}^{\lfloor\mathring{c}}f\left(\Theta_{\mathring{c}}|d(\mathring{t})\right)}{{}^{\lfloor\mathring{c}}f\left(\Theta_{\mathring{c}}|d(0)\right)} \frac{\mathcal{B}\left(\sum_{c=1}^{\mathring{c}-1} {}^{\lfloor\mathring{c}}\kappa_{c;0},\ {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};0}\right)}{\mathcal{B}\left(\sum_{c=1}^{\mathring{c}-1} {}^{\lfloor\mathring{c}}\kappa_{c;\mathring{t}},\ {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};\mathring{t}}\right)} \frac{\mathcal{I}_{\mathring{c}}(d(0))}{\mathcal{I}_{\mathring{c}}(d(\mathring{t}))} \alpha_{\mathring{c}}^{{}^{\lfloor\mathring{c}}\kappa_{\mathring{c};0} - {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};\mathring{t}}}. \end{aligned} \tag{6.115} $$

The value $\Theta_{\mathring{c}}$ is fixed by the requirement (6.114), however, the value $\alpha_{\mathring{c}}$ is free. It indicates that the derived formula is only approximate one and holds exactly only if ${}^{\lfloor\mathring{c}}\kappa_{\mathring{c};\mathring{t}} = {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};0}$, if no data are assigned to the component ${}^{\lfloor\mathring{c}}$. The component should be taken as superfluous whenever ${}^{\lfloor\mathring{c}}\kappa_{\mathring{c};\mathring{t}} \leq {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};0}$. In the remaining case, we shall take averaged version of the above identity by simply integrating out $\alpha_{\mathring{c}}$ over its domain $[0,1]$. Alternatively, the expected value $\hat{\alpha}_{\mathring{c};\mathring{t}} \propto {}^{\lfloor\mathring{c}}\kappa_{\mathring{c};\mathring{t}}$ can be inserted instead of unknown α. Preliminary numerical experience provides a slight preference for the first option. It gives directly the algorithm searched for. It is written for the exponential family.

Algorithm 6.6.9 (Systematic cancelling of components in the exponential family)
Initial mode

- *Perform estimation of a mixture with factors in exponential family. The mixture estimate is described by the collection of statistics* $\{V_{ic;t}, \nu_{ic;t}\}_{c\in c^*, i=1,\ldots,\mathring{d}, t\in\{0,\mathring{t}\}}$. *The factors with the common* i *are supposed to describe the same entry of* $d_{i;t}$ *irrespective of the component number.*

- *Evaluate the individual normalization factors* $\ln(\mathcal{I}(V_{ic;t}, \nu_{ic;t}))$, $\forall c \in c^*, i = 1, \ldots, \mathring{d}, t \in \{0, \mathring{t}\}$.

- *Evaluate* $s = \sum_{c=1}^{\mathring{c}} \kappa_{c;\mathring{t}}$, $s_0 = \sum_{c=1}^{\mathring{c}} \kappa_{c;0}$ *and set* $c = 1$.

Evaluation mode

Do while $c \leq \mathring{c}$ *and* $\mathring{c} > 1$

Set $s = s - \kappa_{c;\mathring{t}}$, $s_0 = s_0 - \kappa_{c;0}$.

If $\kappa_{c;\mathring{t}} > \kappa_{c;0}$

$$l = \ln\left(\mathcal{B}(s_0, \kappa_{c;0})\right) - \ln\left(\mathcal{B}(s, \kappa_{c;\mathring{t}})\right) + \ln\left(f(\Theta_c|d(\mathring{t}))\right) - \ln\left(f(\Theta_c|d(0))\right) - \ln\left(\mathcal{B}(s - s_0, \kappa_{c;\mathring{t}} - \kappa_{c;0})\right)$$

For $i = 1, \ldots, \mathring{d}$

$$l = l + \ln\left(\mathcal{I}(V_{ic;0}, \nu_{ic;0})\right) - \ln\left(\mathcal{I}(V_{ic;\mathring{t}}, \nu_{ic;\mathring{t}})\right)$$

end of the cycle over i

else set $l = 1$

end of the test on increase of κ_c

If $l > 0$

Swap c *with* $\mathring{c}$ *and set* $\mathring{c} = \mathring{c} - 1$, *i.e. cancel the component*

Stop if $\mathring{c} = 1$

else set $s = s + \kappa_{c;\mathring{t}}$, $s_0 = s_0 + \kappa_{c;0}$, $c = c + 1$

end of the test on v-log-likelihood increase

end of the while cycle over c

Remark(s) 6.6.3

1. *Originally, components were cancelled using hypothesis testing with re-estimation after each trial cancelling. It worked but it was computationally expensive.*

2. *Intuitively, the components with very small estimates* $\hat{\alpha}_{c;\mathring{t}}$ *are candidates for cancelling. This check is made indirectly in the proposed algorithm as the increase of the statistics* κ_c *determining them is the same as the increase of statistics* ν_{ic}. *The presented solution respects moreover the values of pdfs defining the individual components. Consequently, the hard selection of the level defining the small values is avoided.*

3. *Alternative solution of the cancelling problem was proposed and tested in [77]. It estimates a background level by including a fixed flat pdf into the estimated mixture. Then, all components that have weights well below it are cancelled.*

Problem 6.6.3 (Predicted v-likelihood in model validation) *Both merging and cancelling exploits prediction of the* v*-likelihood. Quality of the prediction is strongly correlated with quality of the model. It leads to the obvious idea: to exploit the discrepancy between predicted and evaluated* v*-likelihood for model validation, see Section 6.7. The discrepancy can be potentially used in controlling the learning process, to be used as the main input in sequential stopping rules.*

Problem 6.6.4 (Removal of superfluous data items) *The final mixture found may model variables that have, even indirectly, no influence on data determining markers of the problem, cf. step 9 in Algorithm 5.6.1. Then, they can and should be removed from considerations totally. The adequate analysis of the graph describing non-trivial relationships among variables is to be complemented to the tool set being developed.*

6.7 Model validation

Complexity of the estimation of the mixture model makes *model validation* a necessary part of the overall design of the p-system. A full scale use of the p-system is a decisive validation test. However, a number of off line tests is needed in order to check the quality of its design while developing it. This aspect makes the topic of Section.

Generally, we test the hypothesis $H \equiv$ estimated model is good. All possible tests differ in specification of alternatives to this hypothesis. The validation art consists of a proper selection of a real competitor that can be evaluated with the limited computer power available.

6.7.1 Learning with labelled data

Construction of the advisory system is simplified if learning data are qualified by experts. Typically, data subsets can be categorized as reflecting excellent, good, bad (possibly with a type of errors) behaviors. It provides an important additional data item, say $e \in e^* \equiv \{1, \ldots, \mathring{e}\}$, $\mathring{e} < \infty$, that is worth of being exploited.

Obviously, the excellent or bad results may be caused by excellent/bad conditions that are out of control of the operator. For instance, excellent output quality of the rolling might be attributed to the operator or to the excellent input quality of the rolled material. It motivates the important warning.

<u>Do not take automatically the quality labels as a direct measure of the managing quality</u>.

Some of the discussed labels might be directly measured. Then, they are simply discrete data that can be modelled by Markov type factors and predicted. Thus, they do not differ from other data and require no special discussion.

Different situation arises when the data are classified <u>ex post</u>. In this case, a finer modelling is desirable as a reasonably managed system provides a few data sets with the label "bad" (errors are exceptional). They could get too small weight or could be completely overlooked when they are treated without taking into account their specific position. This danger is enhanced by the fact that such labels are constant over whole data blocks.

For the finer modelling of this case, we formulate adequate hypotheses. In their formulation, we assume that the standard and labelled data form blocks with their specific time counters.

$H_s \equiv$ The observed difference in quality is caused by inseparable influence of external conditions and the way of operating. In technical terms, a single mixture describes the standard $d(\mathring{t}_s)$ as well as the labelled $d(\mathring{t}_e)$ data. Thus, for $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$,

$$f(d(\mathring{t})|H_s) = \int f(d(\mathring{t})|\Theta, H_s) f(\Theta|H_s)\, d\Theta. \tag{6.116}$$

The symbol H_s in the conditions stresses that both the structure of mixture and the prior pdf are chosen under this hypothesis.

H_e The observed difference in quality is caused by difference in operating. In technical terms, different mixtures should be used for the standard $d(\mathring{t}_s)$ and labelled $d(\mathring{t}_e)$ data, respectively. Thus, for $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$,

$$f(d(\mathring{t})|H_e) = \int f(d(\mathring{t}_s)|\Theta_s) f(\Theta_s)\, d\Theta_s \int f(d(\mathring{t}_e)|\Theta_e, H) f(\Theta_e)\, d\Theta_e. \tag{6.117}$$

The structure of the mixture and the prior parameter estimate for processing of $d(\mathring{t}_e)$ may differ from those used on $d(\mathring{t}_s)$. This fact is stressed by the indices $_s$, $_e$.

With these elements and no prejudice, $f(H_s) = f(H_e)$, the Bayes rule provides the posterior pdf $f(H_s|d(\mathring{t}))$. The common model – hypothesis H_s – can be accepted if this probability is high enough.

Even if H_s is accepted then the available labelling might help us in labelling the factors active on $d(\mathring{t}_e)$ as potentially dangerous.

If H_e is accepted both models should be used separately. The factors of the predictor obtained from $f(d(\mathring{t}_s)|\Theta_s)$ close to the predictive factors obtained from $f(d(\mathring{t}_e)|\Theta_e)$ should be treated as potentially dangerous. The conceptual testing algorithm looks as follows.

Algorithm 6.7.1 (Test of data homogeneity)

1. *Run complete mixture estimations on the standard* $d(\mathring{t}_s)$*, labelled* $d(\mathring{t}_e)$ *and concatenated data* $d(\mathring{t}) \equiv \big(d(\mathring{t}_s), d(\mathring{t}_e)\big)$.
2. *Evaluate the corresponding v-likelihood*

$$\begin{aligned} f(d(\mathring{t}_s)) &= \int f(d(\mathring{t}_s)|\Theta_s) f(\Theta_s)\, d\Theta_s, \quad f(d(\mathring{t}_e)) = \int f(d(\mathring{t}_e)|\Theta_e) f(\Theta_e)\, d\Theta \\ f(d(\mathring{t})|H_s) &= \int f(d(\mathring{t}_s), d(\mathring{t}_e)|\Theta_s, H_s) f(\Theta_s|H_s)\, d\Theta_s. \end{aligned}$$

3. *Determine probability of* H_s *that a single standard model should be used*

$$f(standard|d(\mathring{t})) \equiv f(H_s|d(\mathring{t})) = \frac{f(d(\mathring{t})|H_s)}{f(d(\mathring{t})|H_s) + f(d(\mathring{t}_s)) f(d(\mathring{t}_e))}. \tag{6.118}$$

4. *Use the single model further on if* $f(H_s|d(\mathring{t}))$ *is close to one. Inspect the factors that were active on* $d(\mathring{t}_e)$ *as potentially dangerous.*
5. *Use both mixtures independently if* $f(H_s|d(\mathring{t}))$ *is close to zero. A danger connected with the situations labelled by e should be signaled whenever the model fitted to* $d(\mathring{t}_e)$ *makes better predictions than the model fitted to the standard data* $d(\mathring{t}_s)$.

6.7.2 Learning results

During the learning, basic assumptions used in the design of the p-system should be tested. The following questions have to be answered during the model validation, see Chapter 5.

- Can the o-system be described by a time invariant mixture model in all practically possible working conditions?
- Do learning data cover sufficiently all these working conditions?

Essentially, we are asking how good is the obtained model in extrapolation of the past to the future. In the off line mode, it can be tested by dividing data available $d(\mathring{t})$ into *learning data* $d(\mathring{t}_l)$ and *validation data* $d(\mathring{t}_v)$. To check it, we assume that tests on labelled data, Section 6.7.1, were performed and we are dealing with homogeneous data describable by a single mixture. For validation of learning results, we formulate hypotheses similar to those in Section 6.7.1 but inspecting the following validation aspect.

$H_l \equiv$ All recorded data – both learning and validation ones – $d(\mathring{t}) \equiv (d(\mathring{t}_l), d(\mathring{t}_v))$ are described by a single mixture model.

The v-likelihood of this hypothesis is obtained by running of the quasi-Bayes or batch quasi-Bayes algorithms on all data

$$f(d(\mathring{t})|H_l) \equiv \int f(d(\mathring{t})|\Theta_l) f(\Theta_l)\, d\Theta_l. \tag{6.119}$$

In this case, both the structure of the model $f(d(\mathring{t})|\Theta_l, H_l)$ and the prior pdf used in learning phase $f(\Theta_l)$ are used also for the validation data $d(\mathring{t}_v)$.

$H_v \equiv$ Learning data and validation data should be described by individual models.

The v-likelihood of this hypothesis is obtained by independent runs of the quasi-Bayes or batch quasi-Bayes algorithms on both data collections giving

$$f(d(\mathring{t})|H_v) \equiv \int f(d(\mathring{t}_l)|\Theta_l) f(\Theta_l)\, d\Theta_l \int f(d(\mathring{t}_v)|\Theta_v) f(\Theta_v)\, d\Theta_v. \tag{6.120}$$

The structure of the used model $f(d(\mathring{t}_v)|\Theta_v)$ and the prior pdf $f(\Theta_v)$ for the validation data $d(\mathring{t}_v)$ may differ from those for learning data.

With these elements and no prejudice, $f(H_l) = f(H_v)$, the Bayes rule provides the posterior pdf $f(H_l|d(\mathring{t}))$. The learnt model can be accepted as good one if the posterior pf $f(H_l|d(\mathring{t}))$ is high enough. Otherwise, we have to search for reasons why the chosen model is not reliable enough. It gives the algorithmic solution.

Algorithm 6.7.2 (Model validation on a homogenous validation data)

1. *Run complete mixture estimations on learning $d(\mathring{t}_l)$, validation $d(\mathring{t}_v)$ and full $d(\mathring{t}) \equiv (d(\mathring{t}_l), d(\mathring{t}_v))$ data collections.*

2. *Evaluate the corresponding v-likelihood values $f\left(d(\mathring{t}_l)\right)$, $f\left(d(\mathring{t}_v)\right)$, $f\left(d(\mathring{t})|H_l\right)$.*

3. *Determine the probability of successful learning*

$$f\left(success|d(\mathring{t})\right) \equiv f\left(H_l|d(\mathring{t})\right) = \frac{f\left(d(\mathring{t})|H_l\right)}{f\left(d(\mathring{t})|H_l\right) + f\left(d(\mathring{t}_l)\right) f\left(d(\mathring{t}_v)\right)}. \tag{6.121}$$

4. *The test is successfully passed if $f(H_l|d(\mathring{t}))$ is close to 1. Otherwise, measures for a better learning have to be taken.*

Results of the test depend, often strongly, on the adopted division of the available data to learning and validation parts. For the generic dynamic case, the division should be obviously restricted to sufficiently long sub-sequences of consecutive records. Thus, it makes sense to perform the test for various cutting moments $t_u \in t_u^* \subset t^*$. The *cutting moment* identifies $d_{l;t} \equiv d_t$ for $t \leq t_u$ and $d_{v;t} = d_t$ for $t > t_u$. The inspected test is the stronger the higher is the v-likelihood

$$f\left(d(\mathring{t})|H_v, t_u\right) \equiv f\left(d(t_u)\right) f(d\left((t_u+1)\_\mathring{t}\right)) \tag{6.122}$$

assigned to the alternative H_v. Thus, we should search for maximum of (6.122) over $t_u \in t_u^*$.

This demanding task that can be performed, by a version of golden-cut rule, by a random choice or by systematic inspection over a predefined grid. The search is significantly simplified by a proper definition of the admissible set t_u^* of cutting moments. For the considered long data sequence $\mathring{t} \approx 10^3 - 10^6$, it makes no sense to vary t_u just slightly. It can be expected that grids defined by the steps 0.1-0.2 $\times\, \mathring{t}$ are sufficient.

The following algorithm that uses fixed grid summarizes the test.

Algorithm 6.7.3 (Model validation with varying cutting time)
Initial mode

- *Perform a complete estimation on* $d(\mathring{t})$.
- *Evaluate the corresponding v-likelihood* $l_l \equiv f(d(\mathring{t})|H_l)$, *where* H_l *labels hypothesis that learning is successful.*
- *Specify the step* h *on the grid* t_u^* *of* t_u, *say* $h = 0.1\mathring{t}$.
- *Set* $t_u =$ *a lower bound on the length of learning data, say 0.5*$\mathring{t}$.

Iterative mode

1. *Perform complete unrelated estimations for the learning* $d_l(t_u) \equiv d(t_u)$ *and validation data* $d_v((t_u+1)\_\mathring{t}) \equiv d((t_u+1)\_\mathring{t})$.
2. *Evaluate the respective v-likelihood values* $l_v(t_u) \equiv f(d_l(t_u))f(d_v((t_u+1)\_\mathring{t}))$.
3. *Evaluate probability*
$$f(learning\ success|d(\mathring{t}), t_u) \equiv f(H_l|d(\mathring{t}), t_u) = \frac{l_l}{l_l + l_v(t_u)}.$$
4. *Increase* $t_u = t_u + h$.
5. *Go to the beginning of* Iterative mode *if* $\mathring{t} - t_u \geq h$.
6. *Find* $p = \min_{t_u^*} f(learning\ success|d(\mathring{t}), t_u)$.
7. *Take the learning as successful one if the probability* p *is close to one. Otherwise analyze reasons for the failure.*

Remark(s) 6.7.1

Learning and validation data have to overlap in dynamic case in order to fill the richest regression vector occurring in the mixture. This fine point is neglected in Algorithm 6.7.3 as it has negligible influence on its performance.

6.7.3 Forgetting-based validation

Check whether the estimated mixture is a good model for the whole data set is the main aim of tests oriented on the learning part of the addressed design. Forgetting techniques were proposed to cope with slow variation of models caused either by real changes or by under-modelling. Thus, it offers a "natural" alternative to the tested hypothesis on sufficiency of the fixed model. It gives immediately the following validation algorithm.

Algorithm 6.7.4 (Forgetting-based validation)
Initial mode

- *Estimate completely the mixture model, i.e. both parameters and structure.*
- *Apply flattening to* $f(\Theta|d(\mathring{t}))$ *in the branching version, Section 6.4.3, so that a good prior pdf* $f(\Theta)$ *is obtained.*
- *Select several forgetting factors* $0 \approx \lambda_1 < \lambda_2 < \ldots \lambda_{\mathring{i}-1} < \lambda_{\mathring{i}} = 1$, $1 < \mathring{i} < \infty$.
- *Set* $f_i(\Theta) = f_{\lambda_i}(\Theta) = f(\Theta)$.

Validation mode

1. *Perform estimation with the stabilized forgetting, Section 3.1 for all* λ_i, *using the tested prior pdf* $f(\Theta)$ *as the alternative.*
2. *Evaluate values of the v-likelihood* $l_{i;\mathring{t}} = f(d(\mathring{t})|\lambda_i)$ *and compute MAP estimate* $\hat{\lambda}$ *of* λ.

3. *Take the model as successful one if* $\lambda_1 = \hat{\lambda}$. *Otherwise search for its improvements.*

Problem 6.7.1 (Analysis of forgetting based validation) *Algorithm 6.7.1 is intuitively appealing and practical experience with it is good one. Appropriate insight is however missing and a deeper analysis is highly desirable.*

Remark(s) 6.7.2

Often, $\mathring{i} = 3$ *forgetting factors are sufficient for a good testing.*

6.7.4 Inspection by human designer

Complete visual inspections are mostly excluded due to the excessive dimensionality of data records. Partial visual inspection is possible and useful especially when design tools are being constructed. Human beings are able to grasp relationships that are hard to be found in an algorithmic way. At the same time, this ability is restricted to 2-3 dimensional spaces. Proposition 7.1.2, proved in Chapter 7, shows how to get them in a computationally cheap way.

6.7.5 Operating modes

A good and stable model does not guarantee that we get a good and efficient advisory system. As discussed in Section 5.1.1, the p-system makes sense if there are good and bad modes of operating that are reflected in the observation space of the p-system. These conditions have to be tested too.

Distance of the observable behavior of the o-system to the user ideal pdf ${}^{\lfloor U}f$ is measured by the KL distance. Recall that we extended the user ideal pdf on the full data space of the p-system (5.6). The distance takes the form

$$\mathcal{D}\left(f||\,{}^{\lfloor U}f\right) = \int f(d(\mathring{t}))\ln\left(\frac{f(d(\mathring{t}))}{{}^{\lfloor U}f(d(\mathring{t}))}\right)dd(\mathring{t}) = \sum_{t\in t^*}\mathcal{E}\left[\int f(d_t|d(t-1))\ln\left(\frac{f(d_t|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))}\right)dd_t\right]. \tag{6.123}$$

Let us assume, that $f(d_t|d(t-1)) = \sum_{c\in c^*}\alpha_c f(d_t|d(t-1),c)$ is a known, i.e. well estimated, mixture. Then, the Jensen inequality (2.14) and the inequality between weighted arithmetic and geometric means imply

$$\begin{aligned}
\mathcal{D}\left(f||\,{}^{\lfloor U}f\right) &\le \sum_{c\in c^*}\alpha_c\sum_{t\in t^*}\mathcal{E}\left[\int f(d_t|d(t-1),c)\ln\left(\frac{f(d_t|d(t-1),c)}{{}^{\lfloor U}f(d_t|d(t-1))}\right)dd_t\right] \text{ and} \qquad (6.124)\\
\mathcal{D}\left(f||\,{}^{\lfloor U}f\right) &\ge \sum_{c\in c^*}\sum_{\tilde{c}\in c^*}\alpha_c\alpha_{\tilde{c}}\sum_{t\in t^*}\mathcal{E}\left[\int f(d_t|d(t-1),\tilde{c})\ln\left(\frac{f(d_t|d(t-1),c)}{{}^{\lfloor U}f(d_t|d(t-1))}\right)dd_t\right].
\end{aligned}$$

The inequality (6.124) reduces to equality if the operator manages to stay at a fixed component, say $\bar{c}$, i.e. $\alpha_c = \delta_{c,\bar{c}}$. At qualitative level, the advisory system aims at such a stabilization and tries to select the component $\bar{c}$ so that the inspected distance is the smallest one. Advising have a chance for success if there is a component whose KL distance to the user ideal pdf is (significantly) smaller than the KL distance of other components. If such a component with a sufficiently high weight α exists then there is chance for designing a good and efficient advisory system. These considerations lead to the following validation algorithm.

Algorithm 6.7.5 (Test of operation modes)

1. *Estimate the best possible mixture model of the o-system.*
2. *Specify the user ideal* ${}^{\lfloor U}f(d(\mathring{t}))$, *extending possibly* ${}^{\lfloor U}f(d_o(\mathring{t}))$ *to* ${}^{\lfloor U}f(d(\mathring{t}))$.
3. *Evaluate the KL distances of individual components* $c \in c^*$

$$f(d(\mathring{t})|c) \approx \prod_{t\in t^*}\prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t},\hat{\Theta}_{ic})$$

to the user ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$. *Here,* $\hat{\Theta}_{ic}$ *should be sufficiently precise estimates of parameters* Θ_{ic}.

4. *Accept that the p-system has a chance for success if there is a components with a relatively high $\hat{\alpha}_{c;\mathring{t}} \propto \kappa_{c;\mathring{t}}$ that has a lower distance to the user ideal pdf than other components.*

   Additional information has to be collected if no such component exists.

Problem 6.7.2 (Systematic model validation) *Above, various ideas on the model validation are outlined. Experimental results as well as diversity of techniques proposed indicate that we are dealing more with a bag of tricks than with a systematic approach. This status should be improved.*

We have at disposal a wide supply of indicators when addressing the problem.

For instance, on the learning side, structure estimates should be stable over various data sets and no branching, merging or cancelling of components improves the value of the v-likelihood, etc.

With respect to design part, prediction of quality markers as well as prediction of recognizable operator actions, if known, are good partial tests.

Also, having in mind warning that the o-system has to be judged as mapping of external conditions on quality of results, see Section 6.7.1, we can proceed as follows.

- *Split available data on those with good and bad management results.*
- *Built independent mixture models on them.*
- *Judge whether the operator has tendency to improve or spoil quality of the overall behavior.*
- *Take operating modes leading to improvement (deterioration) as good (bad) ones.*
- *Search for similarity of good (bad) modes obtained on both data files.*

Problem 6.7.3 (Indicators of model quality) *Problem 6.7.2 is quite hard. In meanwhile, even collection of and stabilization of intuitively acceptable indicators like*

- *comparison of data moments predicted by the estimated model with their sample counterparts*
- *analysis of components as dynamic mappings (eigenstructure)*
- *comparison of theoretical and measured properties (whiteness of prediction errors)*
- *...*

would be invaluable.

Chapter 7

Solution and principles of its approximation: design part

A formal solution of the design of the advisory system helps us to clarify the structure of computations that should be made. Their practical applicability is restricted by the "curse of dimensionality". Thus, the formal solution has to be complemented by approximate but feasible evaluations. The discussion of their principles for learning part of the advisory system is presented in Chapter 6. The discussion of the design part is presented here.

The design is based on the model of the o-system. The observed behavior, tailored to the rate of operator actions, is described by the mixture (5.8)

$$f(d_t|\phi_{t-1},\Theta) = \sum_{c\in c^*} \alpha_c f(d_t|\phi_{c;t-1},\Theta_c,c).$$

Each component $f(d_t|\phi_{c;t-1},\Theta_c,c)$ is decomposed into the product of factors, see Agreement 5.2.1,

$$f(d_t|\phi_{c;t-1},\Theta_c,c) = \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t},\Theta_{ic},c),\ i^* \equiv \{1,\ldots,\mathring{d}\}.$$

The factors are parameterized by individual parameters Θ_{ic}. Collection of these parameters, together with probabilistic weights $\alpha \in \alpha^*$ (5.8), form the multi-variate parameter Θ of the mixture. Structure of the mixture as well as its parameters are assumed to be known throughout this chapter. For the fixed advisory system, their reliable point estimates are supposed to be obtained during the off line learning phase. For the adaptive advisory system, the initial off line estimates are permanently corrected. In both cases, the certainty-equivalence strategy, Section 4.2.2, is adopted. Formally, parameters are taken as known quantities and thus can be and are omitted in conditions.

Chapter is organized as follows. Section 7.1 discusses common design tools like evaluation of the stationary behavior of the o-system and projections of the mixture model to low-dimensional spaces as well as approximate predictors. The model relating advices to responses of the o-system is described in Section 7.1.3. This model is restricted to the case of the fully co-operating operator.

Then, the design of advising strategies is addressed in Section 7.2. Academic, industrial and simultaneous cases are distinguished, Agreements 5.4.3, 5.4.4. The presentation of advices and evaluation of the overall state are addressed Section 7.3.

The adopted fully probabilistic design, Section 2.4.2 is often infeasible. For this reason, upper bounds of a Jensen type (2.14) are found that contain no logarithms of mixtures or their ratios. The effort spent on the approximation instead of a direct invention of feasible loss functions pays back: significant relationships between various elements are "discovered". It was the most visible on the design of interactions with operator in the academic case. Attempts to invent directly feasible loss functions have failed to judge the presentation quality whenever components of the estimated and ideal model are identical.

The basic evaluations performed here are simple but sometimes cumbersome due to the use of integral expressions for the KL distance. We prefer this way that allows us to distinguish clearly quantities in conditions, quantities integrated out and those optimized.

7.1 Common tools

7.1.1 Model projections in design

Let ϕ be the richest state vector of the considered mixture. For its given initial value, the time invariant mixture model defines a complete distribution of data $d(\mathring{t})$. Its derivatives, called *projections*, are used in analysis, design and exploitation of the p-system. They are discussed here.

Steady state pdfs

Analysis of the steady-state behavior of individual, permanently active, components serves both for the model validation, Section 6.7, and approximate design, Section 7.2. In both cases, the steady-state moments of the marker, Section 5.2, are primarily judged. The choice of the marker depends on the specific application. The evaluation of the full steady state pdf described bellow preserves the freedom in the marker choice by avoiding a specific one.

Proposition 7.1.1 (The steady state pdf for a component with the phase-form state) *Let a component be described by the pdf*

$$f(d_t|d(t-1)) = f(d_t|d_{t-1},\ldots,d_{t-\partial}),\ 1 \le \partial < \infty, \tag{7.1}$$

i.e. it has the state ϕ_{t-1} *in the phase form* $\phi'_{t-1} = [d'_{t-1},\ldots,d'_{t-\partial},1]$*, Agreement 5.2.1. Let the* steady state pdf $f_\infty \equiv \lim_{t\to\infty} f_t$ *of the pdf* $f_t \equiv f(\phi_t)$ *exist on* $\phi^* \equiv \{[\phi'_1,\ldots,\phi'_\partial]'\}, \phi_k \in d^*,\ k \in \{1,\ldots,\partial\}$*. Then, it solves the equation*

$$f_\infty(\phi_1,\ldots,\phi_\partial) = \int_{d^*} f\left(d_t = \phi_1|\phi_{t-1} = \left[\phi_2,\ldots,\phi_\partial,\tilde{d}\right]\right) f_\infty\left(\phi_2,\ldots,\phi_\partial,\tilde{d}\right)\, d\tilde{d}. \tag{7.2}$$

Proof: It is directly implied by the marginalization and chain rules and by the phase form of the state. □

Remark(s) 7.1.1

1. *Proposition 7.1.1 provides the necessary but not sufficient condition for existence of the steady state pdf. Thus, the solution of (7.2) has to be checked whether it represents the steady state pdf* f_∞*. Uniqueness is not guaranteed.*
2. *Note that mixture containing components that do not possess the steady state pdf may have steady state pdf, see Remark 9.1.1 and the item 8 of Remarks 9.2.4.*
3. *For static components,* $\partial = 0$*, the steady state pdf* f_∞ *coincides with the pdf describing the component. This makes the evaluation of individual static components trivial. It indicates their descriptive weakness.*

Marginal and conditional pdfs

The important question how to present the obtained results to the operator in a comprehensible manner arises. The probabilistic description of data by conditional pdfs provides an efficient presentation tool.

In the industrial case, Agreement 5.4.3, the advisory system directly provides the ideal randomized control strategy for the recognizable actions $u_{o;t}$, see Section 5.4.5. If the recognizable inputs have a high dimension, then it is necessary to specify preference among them: either automatically, Section 7.3.1, or manually. Marginal pdfs of the most important recognizable actions conditioned on the known past data are presented to the operator. Similarly, selected entries of the o-innovations $\Delta_{o;t}$ are predicted using the ideal pdf according to which the operator should behave.

The condition of the prediction contains measured values, the values proposed by the p-system and values contemplated by the operator. The latter case supports question and answer mode of advising. The operator can inspect what happen if he chooses a specific value of the selected variable. He can select a suitable value of a suitable variable in order to move other quantities to a desired range.

In all these cases, predictions by the mixture model of the o-system or by the ideal mixture model have to be done and relevant marginal pdfs of these predictions presented.

Conditioning and marginalization are thus the universal operations needed for exploiting the mixture-based predictors. We summarize them for reference purposes.

Proposition 7.1.2 (Marginal and conditional pdfs of mixture model) *Let the joint pdf of data $d(\mathring{t})$ be generated by the known Markov mixture model in the factorized form*

$$f(d_t|d(t-1)) = \sum_{c\in c^*} \alpha_c \prod_{i\in i^*} f(d_{ic;t}|d_{(i+1)\_\mathring{i}c;t}, \phi_{c;t-1}, c) \equiv \sum_{c\in c^*} \alpha_c \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t}, c), \text{ where}$$

$d_{ic;t}$ are permuted entries of d_t and $\phi_{c;t-1}$ are observable states of the individual components and $\psi_{ic;t}$ are regression vectors of respective factors.

Let $d_{jc;t}$, $j \in j^* \equiv \{j_1, \ldots, j_{\mathring{j}}\} \subset \{1, \ldots, \mathring{i}\} \equiv j^* \cup \overline{j^*}$, $j^* \cap \overline{j^*} = \emptyset$ be selected entries of d_t. Then, their predictor is the mixture

$$f(d_{j_1\_\mathring{j}\,c;t}|d(t-1)) = \sum_{c\in c^*} \alpha_c \int \prod_{i\in i^*} f(d_{ic;t}|d_{(i+1)\_\mathring{i}c;t}, \phi_{c;t-1}, c)\, dd_{k\in\overline{j^*}c;t}, \text{ where} \tag{7.3}$$

the integration is performed over entries whose prediction is not required.

All factors predicting $d_{ic;t}$ with $i \in \overline{j^*}$ and $i < \min\{j \in j^*\}$ integrate to unity in (7.3). Except of them, the non predicted variables have to be integrated out explicitly.

Let $j^* = \beta^* \cup \gamma^*$, $\beta^* \cap \gamma^* = \emptyset$. If $\beta^* \neq \emptyset$, $\gamma^* \neq \emptyset$, then, the predictor of the data entries $d_{\beta\in\beta^*;t}$, conditioned on the past data $d(t-1)$ and the data entries $d_{\gamma\in\gamma^*;t}$, is the ratio of mixtures

$$f(d_{\beta_1\_\mathring{\beta}\,c;t}|d_{\gamma_1\_\mathring{\gamma}\,c;t}, d(t-1)) = \frac{\sum_{c\in c^*} \alpha_c \int \prod_{i\in i^*} f(d_{ic;t}|d_{(i+1)\_\mathring{i}c;t}, \phi_{c;t-1}, c)\, dd_{k\in\overline{j^*}c;t}}{\sum_{c\in c^*} \alpha_c \int \prod_{i\in i^*} f(d_{ic;t}|d_{(i+1)\_\mathring{i}c;t}, \phi_{c;t-1}, c)\, dd_{k\in\overline{j^*}c;t} d_{\beta\in\beta^*c;t}}. \tag{7.4}$$

Proof: It is directly implied by the marginalization and chain rules. □

Remark(s) 7.1.2

1. *Sometimes, the integrations can be approximated by inserting expectations of the appropriate variables instead of the values to be integrated out.*
2. *The rational form (7.4) shows clearly the need for predictors of the variables with indices in γ^* that are used in the condition only.*

7.1.2 Dynamic predictors in advising

Long-horizon predictors

Generalized Bayesian estimation, Proposition 2.5.1, evaluates the posterior pdf $f(\Theta|d(\mathring{t}))$ on unknown parameters Θ. For the fixed advisory system with the pdf $f(\Theta|d(0))$ obtained on learning data, the estimate serves for constructing fixed predictors $f(d_t|d(t-1)) = \int f(d_t|d(t-1), \Theta) f(\Theta|d(0))\, d\Theta$. They are used in the design of advices and applied to new data d_t measured in on line mode. We assume that under-modelling is negligible and data are informative enough. Then, the predictors are approximated $f(d_t|d(t-1)) \approx f(d_t|d(t-1), \hat{\Theta})$, i.e. a point estimate $\hat{\Theta}$ of Θ is inserted into the parameterized model $f(d_t|d(t-1), \Theta)$.

We process normal operation records. Thus, information content of data may be restricted. It does not influence quality of short-horizon predictors but may produce rather bad long-horizon predictors and good predictors are important for the design. The situation when the predictor behaves as an unstable dynamic mapping is the worst and the most visible case. Recall, that the predictor is unstable if it has no steady-state pdf.

We have to face this problem that may occur even if all measures increasing the information content of data and decreasing inevitable under-modelling are exhausted. A Bayesian solution is outlined here.

Let $\Theta_s^* \subset \Theta^*$ be set of reasonable parameters. Typically, Θ_s^* describes stable mappings. This prior information on adequate predictors implies that support of the prior pdf $f(\Theta)$ has to be reduced from Θ^* to Θ_s^*. It does not influence evaluation of the likelihood and the "restricted" posterior pdf becomes simply proportional to $f(\Theta|d(\mathring{t}))\chi_{\Theta_s^*}(\Theta)$, where $\chi_{\Theta_s^*}(\cdot)$ is indicator of the set Θ_s^*. The corresponding expected value $\hat{\Theta}_s \equiv \mathcal{E}\left[\Theta|d(\mathring{t})\right]$ is then inserted into the parameterized model $f(d_t|d(t-1),\Theta)$. Thus, the predictor gets the form $f(d_t|d(t-1),\hat{\Theta}_s)$. Evaluation of $\hat{\Theta}_s$ is the only trouble faced. Typically, it must be done numerically. A straightforward use of Monte Carlo technique seems to be sufficient.

The above discussion motivates the following algorithm projecting predictors on those given by parameters in a "reasonable" set Θ_s^*.

Algorithm 7.1.1 (Choice of good dynamic predictors)

Initial (off line) mode

- *Estimate the mixture model of the o-system, Chapter 6.*
- *Specify set $\Theta_s^* \subset \Theta^*$ determining meaningful dynamic predictors.*
- *Test whether the unrestricted point estimate $\hat{\Theta}$ of Θ belongs to Θ_s^*.*
- *Set $\hat{\Theta}_s = \hat{\Theta}$ if $\hat{\Theta} \in \Theta_s^*$ and stop. Otherwise, continue.*
- *Select the upper bound $\mathring{n}$ on the number of iterations n and set $n = 0$.*
- *Select the required number $\mathring{m}$ of samples to be observed in Θ_s^* and set $m = 0$.*
- *Set $\hat{\Theta}_s = 0$, where this zero matrix has the same structure as the inspected Θ.*

Iterative mode

1. *Do while $n < \mathring{n}$ & $m < \mathring{m}$.*
2. *Set $n = n + 1$.*
3. *Take an independent sample $\tilde{\Theta} \sim f\left(\Theta|d(\mathring{t})\right)$.*
4. *Go to the step 1 if $\tilde{\Theta} \notin \Theta_s^*$.*
5. *Set $\tilde{m} = m + 1$ and $\tilde{\Theta}_s = \hat{\Theta}_s + \frac{1}{\tilde{m}}(\hat{\Theta}_s - \tilde{\Theta})$.*
6. *Set $\hat{\Theta}_s = \tilde{\Theta}_s$ and $m = \tilde{m}$ if $\tilde{\Theta}_s \in \Theta_s^*$.*
7. *Go to the step 1.*

Accept the value $\hat{\Theta}_s$ as the final estimate if $m = \mathring{m}$ otherwise take the construction as unsuccessful.

Remark(s) 7.1.3

1. *The stopping when $\hat{\Theta} \in \Theta_s^*$ is justified by the small support of the posterior pdf that is expected after processing large amount of data. It should not be applied whenever there are doubts in this respect.*
2. *The algorithm rests on our ability to determine efficiently whether a given sample $\tilde{\Theta}$ belongs to Θ_s^*. It restricts possible selections of Θ_s^*.*
3. *Θ_s^* is typically set of stable predictors but it can be specified by the range of static gains or knowledge of some signs of parameter entries.*
4. *The exact Bayesian estimation is not influenced by the set Θ_s^*. The approximate estimation, see Section 6.5, is generally influenced by it as the set Θ_s^* influences one-step ahead predictors used. We conjecture that this influence can be neglected if one-step-ahead predictions are good. It seems to be confirmed by experiments.*
5. *The test whether the updated mean value $\hat{\Theta}_s$ belongs to Θ_s^* is generally necessary as Θ_s^* need not be convex.*

Approximation of dynamic predictors

Discussion in Section 5.3 has shown that dynamic mixtures with data-dependent weights cannot be efficiently used. In the same section, a universal approach has been proposed that overcomes drawbacks of mixtures with constant weights. This solution requires, however, at least n-times (grouping rate $n > 1$) more evaluations than the standard estimation. Here, a generally applicable approximation is proposed that covers cases when a component is active for a long time period.

In Section 5.3, the data dependent component weights $\tilde{\alpha}_c(d(t-1), \Theta)$ are interpreted through a projection and approximation of a more complex parameterized model. Alternatively, $\tilde{\alpha}_c(d(t-1), \Theta)$ can be seen as the probability that the data vector Ψ_t, Agreement 5.2.1, belongs to a set Ψ_c^* on which the c-th component is the best approximation of the "objective" system description, cf. Chapter 2. It is also the probability that the pointer $c_t = c$. This probability is evaluated during the approximate estimation and found to be proportional to $\alpha_c f(d_t|d(t-1), c)$, cf. Section 6.5. Thus, we know it after measuring Ψ_t.

For the discussed case, that models properly a rich class of applications, we can and will assume that

$$\Psi_t \in \Psi_c^* \text{ with a high probability if } \Psi_{t-1} \in \Psi_c^*. \tag{7.5}$$

This assumption leads to the following approximate dynamic predictor

$$\begin{aligned} f(d_t|d(t-1)) &= \sum_{c \in c^*} w_c(d(t-1)) f(d_t|d(t-1), c) \quad \text{with} \\ w_c(d(t-1)) &\equiv \frac{\alpha_c f(d_{t-1}|d(t-2), c)}{\sum_{\tilde{c} \in c^*} \alpha_{\tilde{c}} f(d_{t-1}|d(t-2), \tilde{c})}. \end{aligned} \tag{7.6}$$

It has to be stressed that the predictor (7.6) can be successful only when the condition (7.5) is met.

7.1.3 Advices and their influence

We repeat the classification of actions of the p-system with an extended presentation of the adopted models of their influence, see Chapter 5. The *superscript* $^{\lfloor I}$ distinguishes the optimized elements.

Agreement 7.1.1 (Advices and their influence) *Advices, i.e. the actions of the p-system,*

$$a_{p;t} \equiv (c_t, u_{o;t}, s_t, p_t) \text{ are classified as follows.} \tag{7.7}$$

Recommended pointers $c_t \in c^*$ *to active components represent actions in the academic design. With them, the optimized ideal pdf is*

$$^{\lfloor I}f(d_t, c_t|d(t-1)) \equiv f(d_t|d(t-1), c_t)\, ^{\lfloor I}f(c_t|d(t-1)), \text{ where} \tag{7.8}$$

the pdf $f(d_t|d(t-1), c_t)$ *describes the* c_t*-th component of the estimated o-model. The collection of pdfs* $\left\{ ^{\lfloor I}f(c_t|d(t-1)) \right\}_{t \in t^*}$ *describes the strategy generating the recommended pointers* $\{c_t\}_{t \in t^*}$.

The ideal pdf presented to the operator has the form

$$^{\lfloor I}f(d_{o;t}|d(t-1)) = \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1)) f(d_{o;t}|d(t-1), c_t). \tag{7.9}$$

Recommended recognizable actions $u_{o;t} \in u_o^*$ *guide the user in selecting recognizable actions.*

Influence of the advising strategy $\left\{ ^{\lfloor I}f(u_{o;t}|d(t-1)) \right\}_{t \in t^*}$ *on the constructed ideal depends on the design context. It differs for industrial and simultaneous designs.*

Industrial design *assumes either learnt component weights* $^{\lfloor I}f(c_t|d(t-1)) = \alpha_{c_t}$ *or weights* $^{\lfloor I}f(c_t|d(t-1))$ *obtained in the preceding academic design. With them, the optimized ideal pdf* $^{\lfloor I}f(d_t|d(t-1))$ *takes the form*

$$\begin{aligned} &^{\lfloor I}f(d_t|d(t-1)) \equiv {}^{\lfloor I}f(u_{o;t}|d(t-1))\, ^{\lfloor I}f(\Delta_t|u_{o;t}, d(t-1)) = \\ &= {}^{\lfloor I}f(u_{o;t}|d(t-1)) \frac{\sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1)) f(\Delta_t|u_{o;t}, d(t-1), c_t) f(u_{o;t}|d(t-1), c_t)}{\sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1)) f(u_{o;t}|d(t-1), c_t)}, \text{ where} \end{aligned} \tag{7.10}$$

the pdfs $f(\Delta_t|u_{o;t}, d(t-1), c)$, $f(u_{o;t}|d(t-1), c)$ *are marginal pdfs of the estimated c-th component. The ideal pdf presented to the user is then*

$$^{\lfloor I}f(d_{o;t}|d(t-1)) = {^{\lfloor I}f}(u_{o;t}|d(t-1)) \frac{\sum_{c_t \in c^*} {^{\lfloor I}f}(c_t|d(t-1)) f(\Delta_{o;t}|u_{o;t}, d(t-1), c_t) f(u_{o;t}|d(t-1), c_t)}{\sum_{c_t \in c^*} {^{\lfloor I}f}(c_t|d(t-1)) f(u_{o;t}|d(t-1), c_t)}. \quad (7.11)$$

Simultaneous design *selects the joint strategy* $\left\{ {^{\lfloor I}f}(c_t, u_{o;t}|d(t-1)) \right\}_{t \in t^*}$. *The joint optimization allows us to make the recommended recognizable action* $u_{o;t}$ *dependent on the recommended pointer* c_t. *It simplifies the optimized model to*

$$^{\lfloor I}f(d_t, c_t|d(t-1)) = f(\Delta_t|u_{o;t}, d(t-1), c_t)\, {^{\lfloor I}f}(c_t, u_{o;t}|d(t-1)), \text{ where} \quad (7.12)$$

$f(\Delta_t|u_{o;t}, d(t-1), c)$ *is the conditional pdf derived from the c-th estimated component. The corresponding ideal pdf presented to the user becomes*

$$^{\lfloor I}f(d_{o;t}|d(t-1)) = \sum_{c_t \in c^*} f(\Delta_{o;t}|u_{o;t}, d(t-1), c_t)\, {^{\lfloor I}f}(c_t, u_{o;t}|d(t-1)). \quad (7.13)$$

Priority actions $p_t \in p^*$ *consist of a* $\mathring{p}-$*vector* $(\mathring{p} \leq \mathring{d}_o)$ *of different indices* $p_{k;t} \in \{1, \ldots, \mathring{d}_o\}$, $k \in \{1, \ldots \mathring{p}\}$; p_t *selects entries of* $d_{o;t}$ *to be shown to the operator.*

Given the ideal pdf $^{\lfloor I}f(c_t, u_{o;t}|d(t-1))$ *describing the recommended pointers* c_t *and the recommended recognizable actions* $u_{o;t}$, *the vectors of priority actions* p_t *are generated by the strategy* $\left\{ {^{\lfloor I}f}(p_t|d(t-1)) \right\}_{t \in t^*}$. *The priority actions are supposed to define the following optimized ideal pdf*

$$\begin{aligned} ^{\lfloor I}f(d_t|p_t, d(t-1)) &= f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1))\, {^{\lfloor I}f}(d_{p_t;t}|d(t-1)) = \\ &= f(d_t|d(t-1)) \frac{^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))}. \end{aligned} \quad (7.14)$$

In (7.14), $d_{\bar{p}_t;t}$ *denotes entries of* d_t *not presented to the operator. The pdf* $f(d_t|d(t-1))$ *is the estimated mixture, the pdf* $f(d_{p_t;t}|d(t-1))$ *is its marginal pdf (also mixture!) and* $^{\lfloor I}f(d_{p_t;t}|d(t-1))$ *is the marginal pdf of the ideal pdf designed previously.*

The ideal presented to the user is $^{\lfloor I}f(d_{\hat{p}_t;t}|d(t-1))$, *where* $\hat{p}_t$ *is a sample from* $^{\lfloor I}f(p_t|d(t-1))$.

Signaling actions $s_t \in s^* \equiv \{0, 1\}$ *stimulate the operator to take appropriate measures when behavior of the o-system significantly differs from the desired one. These actions, when respected by the operator, modify the behavior of the non-guided o-system* $f(d_t|s_t = 0, d(t-1)) \equiv f(d_t|d(t-1))$ *to the behavior described by the ideal pdf* $f(d_t|s_t = 1, d(t-1)) \equiv {^{\lfloor I}f}(d_t|d(t-1))$. *The ideal pdf* $^{\lfloor I}f(d_t|d(t-1))$ *arises from the academic, industrial or simultaneous design of the p-system. The optimized ideal pdf has the form*

$$^{\lfloor I}f(d_t, s_t|d(t-1)) = {^{\lfloor I}f}(d_t|s_t, d(t-1))\, {^{\lfloor I}f}(s_t|d(t-1)), \text{ where} \quad (7.15)$$

the probabilities $\left\{ {^{\lfloor I}f}(s_t|d(t-1)) \right\}_{t \in t^*}$ *describe the signaling strategy. The optimized probabilities are presented to the operator. Typically, the probability* $^{\lfloor I}f(s_t = 1|d(t-1))$ *is converted into a color that reflects the degree of the need for operator actions.*

The above items determine the overall model of the connection of the o-system and p-system. It is used for designing advices and it assumes the fully co-operating operator.

Problem 7.1.1 (Real operator) *It would be desirable to model an imperfect co-operation of a real operator. Yet, we found no simple, sufficiently universal, way to do that. This problem will be addressed in the future. At this moment, we can just passively check whether the operator co-operates: the proposed model is expected to provide a higher v-likelihood than the original model of the o-system. This test should be implemented in the on line use of the advisory system.*

7.1.4 Fully probabilistic design in advising

The fully probabilistic design, Proposition 2.4.4, makes the conceptual framework we follow. It requires an additional inspection of the KL distance that serves us as the loss function.

Let us split the p-data into the p-innovations Δ_t and the p-actions a_t

$$d(\mathring{t}) \equiv d_p(\mathring{t}) \equiv (d_o(\mathring{t}), d_{p+}(\mathring{t})) = (\Delta_o(\mathring{t}), a(\mathring{t}), \Delta_{p+}(\mathring{t})).$$

Here, we need not refine the structure of the p-actions a. Innovation entries Δ_o belong to the o-data space and Δ_{p+} to the surplus data space of the p-system.

A recursive expression of the KL distance for a fixed advising strategy suits for initializations of the design and for checking its results. It is described by the following proposition.

Proposition 7.1.3 (Recursive evaluation of the KL distance) *The KL distance of the pdfs* $^{\lfloor I}f(d(\mathring{t}))$ *and* $^{\lfloor U}f(d(\mathring{t}))$ *can be expressed as follows*

$$\begin{aligned}
\mathcal{D}\left({}^{\lfloor I}f(d(\mathring{t})) || {}^{\lfloor U}f(d(\mathring{t})) \right) &= \mathcal{E}\left[\sum_{t \in t^*} \omega(d(t-1)) \right], \text{ where the conditional KL distance} \qquad (7.16)\\
\omega(d(t-1)) &\equiv \int {}^{\lfloor I}f(d_t|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_t|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))} \right) d_t.
\end{aligned}$$

The value of the KL distance

$$\mathcal{D}\left({}^{\lfloor I}f(d(\mathring{t})) || {}^{\lfloor U}f(d(\mathring{t})) \right) \equiv -\ln(\gamma(d(0))) \text{ is generated recursively for } t = \mathring{t}, \mathring{t}-1, \ldots, 1$$

$$-\ln(\gamma(d(t-1))) \equiv \mathcal{E}\left[\omega_\gamma(d(t))|d(t-1)\right] \equiv \int {}^{\lfloor I}f(d_t|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_t|d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(d_t|d(t-1))} \right) d_t,$$

where the weighted conditional KL distance *is defined by*

$$\begin{aligned}
\omega_\gamma(d(t-1)) &\equiv \int {}^{\lfloor I}f(d_t|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_t|d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(d_t|d(t-1))} \right) d_t. \text{ The terminal } \gamma \text{ is}\\
\gamma(d(\mathring{t})) &\equiv 1.
\end{aligned}$$

Proof: It holds

$$\begin{aligned}
\mathcal{D}\left({}^{\lfloor I}f(d(\mathring{t})) || {}^{\lfloor U}f(d(\mathring{t})) \right) &\equiv \int {}^{\lfloor I}f(d(\mathring{t})) \ln\left(\frac{{}^{\lfloor I}f(d(\mathring{t}))}{{}^{\lfloor U}f(d(\mathring{t}))} \right) dd(\mathring{t}) \underbrace{=}_{chain\ rule}\\
&= \int {}^{\lfloor I}f(d(\mathring{t})) \sum_{t \in t^*} \ln\left(\frac{{}^{\lfloor I}f(d_t|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))} \right) dd(\mathring{t}) \underbrace{=}_{linearity,\, marginalization,\, chain\ rule,\, Fubini}\\
&= \underbrace{\sum_{t \in t^*} \int {}^{\lfloor I}f(d(t-1)) \underbrace{\left[\int {}^{\lfloor I}f(d_t|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_t|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))} \right) d_t \right]}_{\omega(d(t-1))} dd(t-1)}_{\mathcal{E}[\omega(d(t-1))]}.
\end{aligned}$$

Defining

$$\begin{aligned}
-\ln(\gamma(d(t-1))) &\equiv \mathcal{E}\left[\sum_{\tau=t}^{\mathring{t}} \omega(d(\tau))|d(t-1) \right] \underbrace{\Rightarrow}_{chain\ rule\ for\ \mathcal{E}}\\
-\ln(\gamma(d(t-1))) &\equiv \mathcal{E}\left[\omega(d(t)) - \ln(\gamma(d(t))|d(t-1)\right] \underbrace{\equiv}_{definition\ of\ \omega_\gamma} \mathcal{E}\left[\omega_\gamma(d(t))|d(t-1)\right] \underbrace{\Rightarrow}_{\mathcal{E}[\cdot] \equiv \mathcal{E}[\cdot|d(0)]}\\
\mathcal{D}\left({}^{\lfloor I}f || {}^{\lfloor U}f \right) &= -\ln(\gamma(0)).
\end{aligned}$$

□

The ideal pdf ${}^{\lfloor I}f(d(\mathring{t})) \equiv {}^{\lfloor I}f(\Delta(\mathring{t}), a(\mathring{t}))$ is constructed so that it reflects managing objectives expressed by the true user ideal pdf ${}^{\lfloor U}f(d_{o;t}|d_o(t-1))$, extended to d^* by the pdf ${}^{\lfloor I}f(d_{p+;t}|d(t-1))$ as discussed in Section 5.1.3,

$$ {}^{\lfloor U}f(d(\mathring{t})) \underbrace{\equiv}_{(5.6)} \prod_{t\in t^*} {}^{\lfloor U}f(\Delta_{o;t}|d_o(t-1))\, {}^{\lfloor I}f(\Delta_{p+;t}|a_t, d(t-1))\, {}^{\lfloor U}f(a_t|d(t-1)). \qquad (7.17) $$

This special form of the user ideal pdf leads to a special form of Proposition 2.4.4.

Proposition 7.1.4 (Fully probabilistic design with a special target) *Let the joint pdf*

$$ {}^{\lfloor I}f(d(\mathring{t})) \equiv {}^{\lfloor I}f(\Delta(\mathring{t}), a(\mathring{t})) \equiv \prod_{t\in t^*} f(\Delta_t|a_t, d(t-1))\, {}^{\lfloor I}f(a_t|d(t-1)) $$

be influenced by the optional strategy $\left\{ {}^{\lfloor I}f(a_t|d(t-1))\right\}_{t\in t^*}$.

Let the innovations $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t})$ *be split into the part* $\Delta_{o;t}$ *belonging to the data space of the operator and the part* $\Delta_{p+;t}$ *belonging to the surplus data space of the advisory system.*

Then, the optimal strategy, minimizing the KL distance

$$ \mathcal{D}\left({}^{\lfloor I}f|| {}^{\lfloor U}f\right) \equiv \int {}^{\lfloor I}f(\Delta(\mathring{t}), a(\mathring{t})) \ln\left(\frac{{}^{\lfloor I}f(\Delta(\mathring{t}), a(\mathring{t}))}{{}^{\lfloor U}f(\Delta(\mathring{t}), a(\mathring{t}))}\right) d(\Delta(\mathring{t}), a(\mathring{t})) $$

to the user-given ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$ *fulfilling (7.17), has the form*

$$ \begin{aligned} {}^{\lfloor I}f(a_t|d(t-1)) &= {}^{\lfloor U}f(a_t|d(t-1)) \frac{\exp[-\omega_\gamma(a_t, d(t-1))]}{\gamma(d(t-1))}, \quad \textit{where} \qquad (7.18)\\ \gamma(d(t-1)) &\equiv \int {}^{\lfloor U}f(a_t|d(t-1)) \exp[-\omega_\gamma(a_t, d(t-1))]\, da_t \\ \omega_\gamma(a_t, d(t-1)) &\equiv \int f(\Delta_t|a_t, d(t-1)) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, a_t, d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|a_t, d_o(t-1))}\right) d\Delta_t \\ \gamma(d(\mathring{t})) &= 1. \end{aligned} $$

The solution is performed against the time course, starting at $t = \mathring{t}$.

Proof: Comparing to Proposition 2.4.4, the main change concerns the definition of the function $\omega_\gamma(a_t, d(t-1))$. It is implied directly by the assumption (7.17). Other changes are just typographical. The target pdf is distinguished by the superscript ${}^{\lfloor U}$, the optimized one by ${}^{\lfloor I}$ and experience is explicitly expressed in terms of the observed p-data $d(t-1)$. □

7.1.5 Approximations of the KL distance

Some predictors forming the optimized pdf ${}^{\lfloor I}f(d(\mathring{t}))$ are mixtures. This makes evaluation of the KL distance $\mathcal{D}\left({}^{\lfloor I}f|| {}^{\lfloor U}f\right)$ and consequently the application of Proposition 7.1.4 hard. To avoid this trouble, we use the Jensen-type inequalities (2.14) for finding an upper bound on this distance. It gives us a chance to optimize at least such an upper bound.

Proposition 7.1.5 (The J distance of a mixture f to ${}^{\lfloor U}f$) *Let us consider the joint pdf on observed data* $d(\mathring{t}) \in d^*(\mathring{t})$

$$ f(d(\mathring{t})) \equiv \prod_{t\in t^*} \sum_{c\in c^*} \alpha_{c;t} f(d_t|d(t-1), c), \ \textit{where} $$

the components $f(d_t|d(t-1), c)$ *as well as their probabilistic weights* $\alpha_{c;t}$*, possibly dependent on* $d(t-1)$*, are known. Let* ${}^{\lfloor U}f(d(\mathring{t}))$ *be another pdf on the same space.*

Then, the following inequality holds

$$\begin{aligned} \mathcal{D}\left(f||\,{}^{\lfloor U}f\right) &\leq \mathcal{J}\left(f||\,{}^{\lfloor U}f\right) \equiv \mathcal{E}\left\{\sum_{t\in t^*}\omega(d(t-1))\right\} \\ \omega(d(t-1)) &\equiv \sum_{c\in c^*}\alpha_{c;t}\omega(c,d(t-1)) \qquad (7.19) \\ \omega(c,d(t-1)) &\equiv \int f(d_t|d(t-1),c)\ln\left(\frac{f(d_t|d(t-1),c)}{{}^{\lfloor U}f(d_t|d(t-1))}\right)dd_t \geq 0. \end{aligned}$$

The Jensen distance *(*J distance*)* $\mathcal{J}\left(f||\,{}^{\lfloor U}f\right)$ *is*

- *non-negative,*
- *equal to zero iff* $f(d_t|d(t-1),c) = {}^{\lfloor U}f(d_t|d(t-1))$ *a.s. for all those* $t \in t^*$ *and* $c \in c^*$ *for which* $\alpha_{c;t} \not\equiv 0$,
- *infinite iff for some* $t \in t^*$, $c \in c^*$ *the pdf* ${}^{\lfloor U}f(d_t|d(t-1)) = 0$ *and the pdf* $f(d_t|d(t-1),c) > 0$ *on a subset of* d^* *of a positive measure while* $\alpha_{c;t} \not\equiv 0$.

Proof: The chain rule and definition of the KL distance, Proposition 2.4.3, imply

$$\mathcal{D}\left(f||\,{}^{\lfloor U}f\right) = \sum_{t\in t^*}\int f(d(t-1))\left[\int f(d_t|d(t-1))\ln\left(\frac{f(d_t|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))}\right)dd_t\right]dd(t-1).$$

For a given $d(t-1)$, the inner integration over d_t^* is made of

$$\begin{aligned} & f(d_t|d(t-1))\ln\left(\frac{f(d_t|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))}\right) = \sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c)\ln\left(\sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c)\right) - \\ - & \sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c)\ln\left({}^{\lfloor U}f(d_t|d(t-1))\right) \leq \\ \underbrace{\leq}_{(2.14)} & \sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c)\ln(f(d_t|d(t-1),c)) - \sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c)\ln\left({}^{\lfloor U}f(d_t|d(t-1))\right) = \\ = & \sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c)\ln\left(\frac{f(d_t|d(t-1),c)}{{}^{\lfloor U}f(d_t|d(t-1))}\right). \end{aligned}$$

Integration of this inequality over d_t^* and the definitions of $\omega(d(t-1))$ and $\omega(c,d(t-1))$ imply (7.19). The non-negativity of $\omega(d(t-1))$ is obvious as, for any fixed $d(t-1)$, it is a convex combination of the non-negative conditional KL distances $\omega(c,d(t-1))$. Remaining properties of the J distance follow from its form and properties of the KL distance, Proposition 2.4.3. □

The inequality (7.19) between the KL and J distances is proved for the mixture model. We have to deal with the ratio of mixture models, Proposition 7.1.4, whenever the surplus data space $d_{p+;t}^*$ of the p-system is non-empty and the special form of the user ideal pdf (7.17) is used. Luckily enough the inequality (7.19) can be extended to this case, too.

Proposition 7.1.6 (The J distance of a pdf f to ${}^{\lfloor U}f$ of the form (7.17)) *Let us consider the joint pdf on observed data* $d(\mathring{t}) \in d^*(\mathring{t})$, $d_t = (d_{o;t}, d_{p+;t})$ *with a non-trivial part* $d_{p+;t}$ *belonging to the surplus data space of the advisory system. Let us assume that*

$$f(d(\mathring{t})) \equiv \prod_{t\in t^*}\sum_{c\in c^*}\alpha_{c;t}f(d_t|d(t-1),c), \text{ where}$$

the components $f(d_t|d(t-1),c)$ *as well as their probabilistic weights* $\alpha_{c;t}$, *possibly dependent on* $d(t-1)$, *are known. Let*

$${}^{\lfloor U}f(d(\mathring{t})) = \prod_{t\in t^*}{}^{\lfloor U}f(d_{o;t}|d_o(t-1))\sum_{c\in c^*}\alpha_{c;t}f(d_{p+;t}|d(t-1),c)$$

be another pdf defining (7.17). Let for some $K \geq 1$ and some pdf $g(d_{p+;t}|d(t-1))$

$$f(d_{p+;t}|d(t-1),c) \leq K g(d_{p+;t}|d(t-1)), \forall c \in c^* \text{ and almost all data in arguments.} \tag{7.20}$$

Then, the following inequality holds

$$\begin{aligned}
\mathcal{D}\left(f|| {}^{\lfloor U}f\right) &\leq \mathcal{J}\left(f|| {}^{\lfloor U}f\right) + \ln(K) \quad \text{with} \\
\mathcal{J}\left(f|| {}^{\lfloor U}f\right) &\equiv \mathcal{E}\left\{\sum_{t\in t^*} \omega(d(t-1))\right\} \\
\omega(d(t-1)) &\equiv \sum_{c\in c^*} \alpha_{c;t}\omega(c,d(t-1)) \\
\omega(c,d(t-1)) &\equiv \int f(d_t|d(t-1),c) \ln\left(\frac{f(d_{o;t}|d_{p+;t},d(t-1),c)}{{}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \geq 0.
\end{aligned} \tag{7.21}$$

Proof: Use of the chain rule for expressing the KL distance implies that we have to inspect the difference

$$\begin{aligned}
& \sum_{t\in t^*} \mathcal{E}\left\{\sum_{c\in c^*} \alpha_{c;t} \int f(d_t|d(t-1),c) \left[\ln\left(\frac{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c};t} f(d_t|d(t-1),\tilde{c})}{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c};t} f(d_{p+;t}|d(t-1),\tilde{c})\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) - \right.\right. \\
- & \left.\left. \ln\left(\frac{K f(d_t|d(t-1),c)}{f(d_{p+;t}|d(t-1),c)\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right)\right] dd_t\right\} \underbrace{\leq}_{(2.14)} \\
\leq & \sum_{t\in t^*} \mathcal{E}\left\{\sum_{c\in c^*} \alpha_{c;t} \int f(d_t|d(t-1),c) \ln\left(\frac{f(d_{p+;t}|d(t-1),c)}{K\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c};t} f(d_{p+;t}|d(t-1),\tilde{c})}\right) dd_t\right\} \underbrace{\leq}_{(7.20)} \\
\leq & \sum_{t\in t^*} \mathcal{E}\left\{\sum_{c\in c^*} \alpha_{c;t} \int f(d_t|d(t-1),c) \ln\left(\frac{g(d_{p+;t}|d(t-1))}{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c};t} f(d_{p+;t}|d(t-1),\tilde{c})}\right) dd_t\right\} = \\
= & -\sum_{t\in t^*} \mathcal{E}\left\{\sum_{c\in c^*} \alpha_{c;t} \int f(d_t|d(t-1),c) \ln\left(\frac{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c};t} f(d_{p+;t}|d(t-1),\tilde{c})}{g(d_{p+;t}|d(t-1))}\right) dd_t\right\} \underbrace{=}_{marginalisation} \\
= & -\sum_{t\in t^*} \mathcal{E}\left\{\mathcal{D}\left(\sum_{c\in c^*} \alpha_{c;t} f(d_{p+;t}|d(t-1),c) || g(d_{p+;t}|d(t-1))\right)\right\} \underbrace{\leq}_{KL\ distance \geq 0} 0.
\end{aligned}$$

□

In the industrial design and when designing presentation priorities, the KL distances of more general ratios of finite mixtures to a given pdf are inspected. The following proposition shows how to approximate such distances.

Proposition 7.1.7 (The J-like distance of a mixture ratio ${}^{\lfloor I}f$ to ${}^{\lfloor U}f$) *Let us consider mixture*

$$f(x,y) = \sum_{c\in c^*} \alpha_c f(x,y|c)$$

on variables x,y given by known components $f(x,y|c)$ and their probabilistic weights α_c, $c \in c^*$. Its marginal pdf is $f(y) = \sum_{c\in c^*} \alpha_c f(y|c)$. Let ${}^{\lfloor U}f(x,y)$, ${}^{\lfloor I}f(y)$ be given pdfs. Then, for

$$\begin{aligned}
{}^{\lfloor I}f(x,y) &\equiv \frac{f(x,y)\, {}^{\lfloor I}f(y)}{f(y)} \\
\mathcal{D}\left({}^{\lfloor I}f|| {}^{\lfloor U}f\right) &\equiv \int \frac{f(x,y)\, {}^{\lfloor I}f(y)}{f(y)} \ln\left(\frac{f(x,y)\, {}^{\lfloor I}f(y)}{f(y)\, {}^{\lfloor U}f(x,y)}\right) dxdy \leq \int {}^{\lfloor I}f(y)\left[\ln\left(\frac{{}^{\lfloor I}f(y)}{{}^{\lfloor U}f(y)}\right) + \omega(y)\right] dy \\
\omega(y) &\equiv \sum_{c\in c^*} f(c|y)\left[\omega(c,y) + \ln\left(\frac{f(c|y)}{\alpha_c}\right)\right], \quad \text{where} \\
\omega(c,y) &\equiv \int f(x|y,c) \ln\left(\frac{f(x|y,c)}{{}^{\lfloor U}f(x|y)}\right) dx \text{ and } f(c|y) \equiv \frac{\alpha_c f(y|c)}{\sum_{c\in c^*} \alpha_c f(y|c)}.
\end{aligned} \tag{7.22}$$

Proof: It holds

$$\begin{aligned}
\mathcal{D} &\equiv \int \frac{f(x,y)\,{}^{\lfloor I}f(y)}{f(y)} \ln\left(\frac{f(x,y)\,{}^{\lfloor I}f(y)}{f(y)\,{}^{\lfloor U}f(x,y)}\right) dxdy \underbrace{\leq}_{(2.14)} \\
&\leq \sum_{c\in c^*} \alpha_c \int \frac{f(x,y|c)}{f(y)}\,{}^{\lfloor I}f(y) \ln\left(\frac{f(x,y|c)\,{}^{\lfloor I}f(y)}{f(y)\,{}^{\lfloor U}f(x,y)}\right) dxdy \qquad \underbrace{=}_{\substack{\text{chain rule}\\ \text{marginalization}\\ f(y)=\sum_{c\in c^*}\alpha_c f(y|c)}} \\
&= \int {}^{\lfloor I}f(y) \ln\left(\frac{{}^{\lfloor I}f(y)}{{}^{\lfloor U}f(y)}\right) dy + \sum_{c\in c^*} \int {}^{\lfloor I}f(y) \underbrace{\frac{\alpha_c f(y|c)}{f(y)}}_{f(c|y)} \ln\left(\frac{\alpha_c f(y|c)}{f(y)\alpha_c}\right) dy + \\
&+ \sum_{c\in c^*} \int {}^{\lfloor I}f(y) f(c|y) \underbrace{\int f(x|y,c) \ln\left(\frac{f(x|y,c)}{{}^{\lfloor U}f(x|y)}\right) dx}_{\omega(c,y)\geq 0}\, dy.
\end{aligned}$$

□

Problem 7.1.2 (Upper bounds on the KL distances with mixture ratios) *There are surely variants of the derived upper bound. The problem should be inspected further on in order to get tighter upper bounds while preserving possibility to evaluate them at least for normal and Markov mixtures.*

The approximations discussed up to now counteract evaluation problems caused by the form of the adopted models: the KL distance is evaluated for mixtures and/or their ratios. The found bounds overcome them. In the adopted fully probabilistic design, Proposition 7.1.4, we have to evaluate two functions $\gamma(d(t))$ and $\omega_\gamma(d(t))$, see (7.18). This brings additional problems and calls for additional approximations. They are prepared here.

We search for upper bounds in order to guarantee that the strategy found by their minimization bounds the KL distance of interest from above.

We bound first the function $\gamma(d(t))$. The Bellman function of the solved design is $-\ln(\gamma(t))$, thus if we find lower bound on $\gamma(d(t))$, we get the desired upper bound on it. The result we call γ-*bound.*

Proposition 7.1.8 (γ-bound) *Let ${}^{\lfloor U}f(a_{t+1}|d(t))$ be a given pdf on a^*_{t+1} and $\omega_\gamma(a_{t+1},d(t)) \geq 0$ be such a function that*

$$\omega_\gamma(d(t)) \equiv \int \omega_\gamma(a_{t+1},d(t))\,{}^{\lfloor U}f(a_{t+1}|d(t))\,da_t < \infty \text{ a.s. on } d^*(t).$$

$$\text{Then,}\quad \gamma(d(t)) \equiv \int {}^{\lfloor U}f(a_{t+1}|d(t)) \exp[-\omega_\gamma(a_{t+1},d(t))]\,da_{t+1} \geq \exp[-\omega_\gamma(d(t))] > 0 \text{ a.s.} \qquad (7.23)$$

Proof: For fixed $d(t)$, $\gamma(d(t))) \equiv \mathcal{E}[\exp(-\omega(\cdot,d(t)))|d(t)]$. The function $\exp(-\omega)$ is a convex function of the argument ω. Thus, Jensen inequality (2.14) is directly applicable. It gives the lower bound searched for. Its positivity is implied by the assumed finiteness of $\omega_\gamma(d(t))$. □

The estimate (7.23) can be refined if the considered actions have a finite amount of possible values, if $\mathring{a} < \infty$. It is the case of advices restricted to $a^* \subset (c^*,p^*,s^*)$, cf. Agreement 7.1.1. Essentially, bounds on individual functions $\omega_\gamma(a_{t+1},d(t))$, $a_{t+1} \in a^*$ are constructed. Their approximation leads to the upper bound on $-\ln(\gamma(d(t-1)))$. The corresponding bound is called ω-*bound.*

Proposition 7.1.9 (ω-bound) *For $\mathring{a} < \infty$ and $t = \mathring{t}, \mathring{t}-1,\ldots,1$, let us consider sequence functions, cf. (7.18),*

$$\gamma(d(t)) \equiv \sum_{a_{t+1}\in a^*} {}^{\lfloor U}f(a_{t+1}|d(t)) \exp[-\omega_\gamma(a_{t+1},d(t))]$$

$$\omega_\gamma(a_t, d(t-1)) \equiv \int f(\Delta_t|a_t, d(t-1)) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, a_t, d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|a_t, d_o(t-1))}\right) d\Delta_t \equiv$$
$$\equiv \underbrace{\int f(\Delta_t|a_t, d(t-1)) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, a_t, d(t-1))}{{}^{\lfloor U}f(\Delta_{o;t}|a_t, d_o(t-1))}\right) d\Delta_t}_{\omega(a_t, d(t-1))} - \underbrace{\int f(\Delta_t|a_t, d(t-1)) \ln(\gamma(d(t)))\, d\Delta_t}_{q(a_t, d(t-1))}.$$

The last decomposition separates the influence of $\gamma(d(t))$ on the value $\omega_\gamma(a_t, d(t-1))$. The first term is $\omega(a_t, d(t-1)) \equiv \omega_{\gamma\equiv 1}(a_t, d(t-1))$.

For fixed t, $d(t-1)$, a_t, let us select arbitrary $a_{t+1} \equiv a_{a_t} \in a^*$ and let us define

$$\tilde{\omega}_\gamma(a_t, d(t-1)) \equiv \omega(a_t, d(t-1)) + \int f(\Delta_t|a_t, d(t-1)) \ln\left({}^{\lfloor U}f(a_{a_t}|d(t)) \exp[-\tilde{\omega}_\gamma(a_{a_t}, d(t))]\right) d\Delta_t$$

with zero second term for $t = \mathring{t}$. Then, $\forall t \in t^*$

$$-\ln[\gamma(d(t))] \equiv -\ln\left[\sum_{a_{t+1}\in a^*} {}^{\lfloor U}f(a_{t+1}|d(t)) \exp[-\omega_\gamma(a_{t+1}, d(t))]\right] \leq -\ln[\tilde{\gamma}(d(t))] \equiv$$
$$\equiv -\ln\left[\sum_{a_{t+1}\in a^*} {}^{\lfloor U}f(a_{t+1}|d(t)) \exp[-\tilde{\omega}_\gamma(a_{t+1}, d(t))]\right]. \quad (7.24)$$

Proof: Let us consider, for fixed $d(t-1)$, a_t, the second term $q(a_t, d(t-1))$ in the expression of $\omega_\gamma(a_t, d(t-1))$ that depends on $\gamma(d(t))$.

By definition, $\gamma(d(t))$ is superposition of non-negative terms. Omitting all of them except one, we get lower bound on $q(a_t, d(t-1))$. Thus, for arbitrary fixed $a_{t+1} \equiv a_{a_t} \in a^*$,

$$\omega_\gamma(a_t, d(t-1)) \equiv \omega(a_t, d(t-1)) - q(a_t, d(t-1)) \leq$$
$$\leq \omega(a_t, d(t-1)) - \int f(\Delta_t|a_t, d(t-1)) \ln\left({}^{\lfloor U}f(a_{a_t}|d(t)) \exp[-\omega_\gamma(a_{a_t}, d(t))]\right) d\Delta_t \equiv \tilde{\omega}(a_{a_t}, d(t-1)).$$

Using definition of $\gamma(d(t-1))$, we get

$$-\ln(\gamma(d(t-1))) \equiv -\ln\left[\sum_{a_t} {}^{\lfloor U}f(a_t|d(t-1)) \exp[-\omega_\gamma(a_t, d(t-1)]\right] \leq$$
$$\leq -\ln\left\{\sum_{a_t} {}^{\lfloor U}f(a_t|d(t-1) \exp\left[-\tilde{\omega}(a_t, d(t-1))\right]\right\}.$$

Thus, the use of the derived upper bound on individual $\omega_\gamma(a_{t+1}, d(t))$ for definition of $\omega_\gamma(a_t, d(t-1))$ guarantees that we get the upper bound on the Bellman function at time $t-1$. For $t = \mathring{t}$, inequality is guaranteed by the common starting condition $\gamma(d(\mathring{t})) = 1$. □

It remains to decide on a proper choice of a_{a_t} to get the tight version of the derived ω-bound. We postpone it as it is the design-case and model specific.

7.2 Design of advising strategies

7.2.1 Academic design

The recommended pointers $c_t \in c^*$ to components form the actions of the academic p-system. They are generated by a causal strategy $d^*(t-1) \to c^*$ described by pfs $\left\{ {}^{\lfloor I}f(c_t|d(t-1)) \right\}_{t\in t^*}$. The strategy $\left\{ {}^{\lfloor I}f(c_t|d(t-1)) \right\}_{t\in t^*}$ determines the potential ideal pdfs

$${}^{\lfloor I}f(d_t, c_t|d(t-1)) = f(d_t|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1)).$$

The extended user ideal pdf, Section 5.1.3, has the form

$$ {}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) = \prod_{t \in t^*} {}^{\lfloor U}f(d_{o;t}|d_o(t-1))\, {}^{\lfloor I}f(d_{p+;t}|d(t-1), c_t)\, {}^{\lfloor U}f(c_t|d(t-1)). \tag{7.25} $$

The user target ${}^{\lfloor U}f(c_t|d(t-1))$ is assumed to have support that excludes dangerous components, see Agreement 5.4.6 and Section 7.1.1. The optimal design of the academic p-system is thus solved by this exclusion if a single non-dangerous component remains. The optimal advising strategy has to be searched for if several non-dangerous components are found.

Proposition 7.2.1 (Academic advisory system minimizing the KL distance) *Let us consider the design of the academic advisory system with actions $c_t \in c^*$ restricted to those indices in c^* that point to the non-dangerous components only. Let us search for the optimal causal advising strategy*

$$ d^*(t-1) \rightarrow c_t \in c^* $$

that minimizes the KL distance of ${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ to the user ideal ${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t}))$ as given in (7.25).

Then, this strategy is given by the following formulas, solved for $t = \mathring{t}, \mathring{t}-1, \ldots, 1$,

$$ \begin{aligned} {}^{\lfloor I}f(c_t|d(t-1)) &= {}^{\lfloor U}f(c_t|d(t-1)) \frac{\exp[-\omega_\gamma(c_t, d(t-1))]}{\gamma(d(t-1))}, \quad \textit{where} \\ \gamma(d(t-1)) &\equiv \sum_{c_t \in c^*} {}^{\lfloor U}f(c_t|d(t-1)) \exp[-\omega_\gamma(c_t, d(t-1))] \\ \omega_\gamma(c_t, d(t-1)) &\equiv \int f(d_t|d(t-1), c_t) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))} \right) dd_t \\ \gamma(d(\mathring{t})) &= 1. \end{aligned} \tag{7.26} $$

The needed pdf $f(d_{o;t}|d_{p+;t}, d(t-1), c)$ is the conditional pdf of the c-th component $f(d_t|d(t-1), c)$ obtained in learning of the o-system.

Proof: Proposition coincides with Proposition 7.1.4, see (7.18), for $d_t \equiv \Delta_t$ and the specific choice of actions of the p-system and their assumed influence on the resulting ideal pdf. □

The above proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the practically used state description with the state ϕ.

Algorithm 7.2.1 (Academic design optimizing the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state ϕ_t, Chapter 6.*
- *Evaluate the steady state behavior of individual components, Section 7.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if $\mathring{c} = 0$.*
- *Take the non-dangerous component as the ideal pdf offered to the user and stop if $\mathring{c} = 1$.*
- *Specify the user ideal pdf ${}^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1})$ on the response of the o-system.*
- *Specify the user ideal pdf ${}^{\lfloor U}f(c_t|d(t-1)) \equiv {}^{\lfloor U}f(c_t|\phi_{t-1})$ on the recommended pointers c_t to the non-dangerous components.*
- *Select the length of the receding horizon $T \geq 1$.*

On line (sequential) mode, *running for $t = 1, 2, \ldots$*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*

2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t + T$ and $\gamma(\phi_\tau) = 1$.*

   Iterative mode

   (a) *Evaluate the* *functions*

$$\begin{aligned}\omega_\gamma(c, \phi_{\tau-1}) &\equiv \int f(d_\tau|\phi_{\tau-1}, c) \ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau}, \phi_{\tau-1}, c)}{\gamma(\phi_\tau)\ {}^{\lfloor U}f(d_{o;\tau}|\phi_{\tau-1})}\right) dd_\tau,\ c \in c^*\\ \gamma(\phi_{\tau-1}) &\equiv \sum_{c_\tau \in c^*} {}^{\lfloor U}f(c_\tau|\phi_{\tau-1}) \exp[-\omega_\gamma(c_\tau, \phi_{\tau-1})].\end{aligned}$$

   (b) *Continue if $\tau = t + 1$. Otherwise decrease $\tau = \tau - 1$ and go to the step 3a.*

4. *Evaluate the advising strategy*

$$ {}^{\lfloor I}f(c_{t+1}|\phi_t) = {}^{\lfloor U}f(c_{t+1}|\phi_t)\frac{\exp[-\omega_\gamma(c_{t+1}, \phi_t)]}{\gamma(\phi_t)}, \quad cf.\ (7.26).$$

5. *Present to the user projections of the ideal pdf*

$$ {}^{\lfloor I}f(d_{o;t+1}|\phi_t) = \sum_{c_{t+1} \in c^*} f(d_{o;t+1}|\phi_t, c_{t+1})\ {}^{\lfloor I}f(c_{t+1}|\phi_t),\ \text{where}$$

   the pdf $f(d_{o;t+1}|\phi_t, c)$ is derived from the c-th estimated component $f(d_{t+1}|\phi_t, c)$.

6. *Go to the beginning of* Sequential mode.

Remark(s) 7.2.1

1. *The quasi-Bayes or quasi-EM algorithms, Section 6.5, possibly with the stabilized forgetting, Section 3.1, suit for the on line updating of the mixture model.*
2. *Other approximate strategies than the certainty-equivalence one are possible but their use makes probably a little sense.*
3. *Presentation of the design results is discussed in Section 7.3.1.*

Feasibility of Algorithm 7.2.1 depends on our ability evaluate functions $\omega_\gamma(\cdot)$ and $\gamma(\cdot)$. It is hard even for normal and Markov mixtures due to the additive form determining $\gamma(\cdot)$. In order to simplify the situation, we use the freedom in the choice of the user ideal pdf for advising actions ${}^{\lfloor U}f(c_{t+1}|\phi_t)$.

Proposition 7.2.2 (Academic design with the KL distance & the most probable advices) *Let us consider the design of the academic advisory system with actions $c_t \in c^*$ restricted to those indices in c^* that point to the non-dangerous components only. Let us search for the optimal causal advising strategy $d^*(t-1) \to c_t \in c^*$ that minimizes the KL distance of ${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ to the user ideal ${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t}))$ (7.25). The obtained minimum depends on the used ideal probabilities $\left\{{}^{\lfloor U}f(c_t|d(t-1))\right\}_{t\in t^*}$. Let us select these probabilities so that the reached minimum is the smallest one.*

Then, the optimal advising strategy is the deterministic feedback recommending the component with the index

$$\begin{aligned} {}^{\lfloor I}c_t &\in \operatorname{Arg}\min_{c_t \in c^*} \omega_\gamma(c_t, d(t-1)) \qquad (7.27)\\ \omega_\gamma(c_t, d(t-1)) &\equiv \int f(d_t|d(t-1), c_t) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1), c_t)}{\exp(-\omega_\gamma({}^{\lfloor I}c_{t+1}, d(t)))\ {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t\\ \omega_\gamma\left({}^{\lfloor I}c_{\mathring{t}+1}, d(\mathring{t})\right) &= 0.\end{aligned}$$

The pdf $f(d_{o;t}|d_{p+;t}, d(t-1), c)$ is the conditional pdf of the c-th component $f(d_t|d(t-1), c)$ obtained by learning the mixture model of the o-system.

Proof: Complexity of the academic design described by Proposition 7.2.1 stems from complexity of the Bellman function that equals to $-\ln(\gamma(d(t)))$ with

$$\begin{aligned}
\gamma(d(t)) &\equiv \sum_{c_{t+1}\in c^*} {}^{\lfloor U}f(c_{t+1}|d(t))\exp[-\omega_\gamma(c_{t+1},d(t))]\\
\omega_\gamma(c_t,d(t-1)) &\equiv \int f(d_t|d(t-1),c_t)\ln\left(\frac{f(d_{o;t}|d_{p+;t},d(t-1),c_t)}{\gamma(d(t))\,{}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right)dd_t.
\end{aligned}$$

The pf ${}^{\lfloor U}f(c_{t+1}|d(t))$ is the optional design knob. We select it so that the Bellman function is minimized with respect to this pf. For that, we have to concentrate ${}^{\lfloor U}f(c_{t+1}|d(t))$ on ${}^{\lfloor I}c_{t+1} \in \operatorname{Arg}\min_{c_{t+1}\in c^*}\omega_\gamma(c_{t+1},d(t))$. In this way, we get $\gamma(d(t)) = \exp\left[-\omega_\gamma\left({}^{\lfloor I}c_{t+1},d(t)\right)\right]$ and the optimal strategy is deterministic

$$ {}^{\lfloor I}f(c_{t+1}|d(t)) = \delta_{c_{t+1},\,{}^{\lfloor I}c_{t+1}} \equiv \begin{cases} 1 & \text{if} \quad c_{t+1} = {}^{\lfloor I}c_{t+1} \\ 0 & \text{if} \quad c_{t+1} \neq {}^{\lfloor I}c_{t+1} \end{cases}. $$

Moreover, the made choice ${}^{\lfloor U}f(c_{t+1}|d(t))$ maximizes the probability ${}^{\lfloor U}f(c_{t+1}|d(t))\exp[-\omega_\gamma(c_{t+1},d(t))]$, i.e. it selects the most probable academic advice among those given by Proposition 7.2.1.

The terminal value $\omega_\gamma\left({}^{\lfloor I}c_{\mathring{t}+1},d(\mathring{t})\right)=0$ is implied by the general terminal value $\gamma(d(\mathring{t}))=1$ valid for the fully probabilistic design.

The additional optimization replaces the random choice (7.26) of the recommended pointers c_t. It takes the single component $f\left(d_t|d(t-1),\,{}^{\lfloor I}c_t\right)$ of the learnt model as the constructed ideal pdf. □

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the practically used state description with the state ϕ.

Algorithm 7.2.2 (The most probable academic design optimizing the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state ϕ_t, Chapter 6.*
- *Evaluate steady state behavior of individual components, Section 7.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if $\mathring{c}=0$.*
- *Take the non-dangerous component as the ideal pdf offered to the user and stop if $\mathring{c}=1$.*
- *Specify the user ideal pdf on the response of the o-system ${}^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1})$.*
- *Select the length of the receding horizon $T \geq 1$.*

Sequential (on line) mode, *running for $t = 1,2,\ldots,$*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t+T$ and set $\omega_\gamma\left({}^{\lfloor I}c_{\tau+1},\phi_\tau\right)=0$.*

   Iterative mode

   (a) *Evaluate the <u>functions</u>*

   $$\begin{aligned}
   \omega_\gamma(c_\tau,\phi_{\tau-1}) &\equiv \int f(d_\tau|\phi_{\tau-1},c_\tau)\ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau},\phi_{\tau-1},c_\tau)}{\exp\left[-\omega_\gamma\left({}^{\lfloor I}c_{\tau+1},\phi_\tau\right)\right]\,{}^{\lfloor U}f(d_{o;\tau}|\phi_{\tau-1})}\right)dd_\tau,\ c_\tau\in c^*\\
   {}^{\lfloor I}c_\tau(\phi_{\tau-1}) &\in \operatorname{Arg}\min_{c_\tau\in c^*}\omega_\gamma(c_\tau,\phi_{\tau-1}).
   \end{aligned}$$

   (b) *Continue if $\tau = t+1$ otherwise decrease $\tau = \tau - 1$ and go to the step 3a.*

4. *Present to the user the ideal pdf* ${}^{\lfloor I}f(d_{o;t+1}|\phi_t) = f\left(d_{o;t+1}|\phi_t, {}^{\lfloor I}c_{t+1}\right)$*, which is marginal pdf of the estimated components* $\{f(d_{t+1}|d(t), c_{t+1})\}$.
5. *Go to the beginning of* Sequential mode.

Feasibility of the design can be achieved also by minimizing an upper bound on the KL distance. The following proposition provides such a solution that suits both to normal components, for which $\omega_\gamma(c_{t+1}, d(t))$ are quadratic forms in data, see Chapter 9, as well as for Markov chains for which $\omega_\gamma(c_{t+1}, d(t))$ are finite-dimensional arrays, see Chapter 11.

Proposition 7.2.3 (Academic advisory system minimizing γ-*bound* on the KL distance) *Let us consider the design of the academic advisory system with actions $c_t \in c^*$ restricted to those indices in c^* that point to the non-dangerous components only. Let us search for the optimal causal advising strategy*

$$d^*(t-1) \to c_t \in c^*$$

that approximately minimizes the KL distance of ${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ to the user ideal ${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t}))$, see (7.25).

Then, the strategy given by the following formulas, solved for $t = \mathring{t}, \mathring{t}-1, \ldots, 1$,

$$\begin{aligned}
{}^{\lfloor I}f(c_t|d(t-1)) &\propto {}^{\lfloor U}f(c_t|d(t-1)) \exp[-\omega_\gamma(c_t, d(t-1))], \quad \textit{where} \qquad (7.28)\\
\omega_\gamma(c_t, d(t-1)) &\equiv \int f(d_t|d(t-1), c_t) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t\\
\gamma(d(t)) &\equiv \exp\left[-\sum_{c_{t+1}\in c^*} {}^{\lfloor U}f(c_{t+1}|d(t))\omega_\gamma(c_{t+1}, d(t))\right]\\
\gamma(d(\mathring{t})) &= 1
\end{aligned}$$

minimizes the γ-bound, Proposition 7.1.8, on the KL distance.

The pdf $f(d_{o;t}|d_{p+;t}, d(t-1), c)$ is the conditional version of the c-th component $f(d_t|d(t-1), c)$ of the learnt mixture model of the o-system.

Proof: We apply Proposition 7.1.4, see (7.18), for the specific choice of actions of the p-system and their influence on the resulting ideal pdf.

$$\begin{aligned}
{}^{\lfloor I}f(c_t|d(t-1)) &\propto {}^{\lfloor U}f(c_t|d(t-1)) \exp[-\omega_\gamma(c_t, d(t-1))], \quad \text{where}\\
\omega_\gamma(c_t, d(t-1)) &\equiv \int f(d_t|d(t-1), c_t) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t\\
\gamma(d(\mathring{t})) &= 1 \text{ and}\\
\gamma(d(t)) &\equiv \sum_{c_{t+1}\in c^*} {}^{\lfloor U}f(c_{t+1}|d(t)) \exp[-\omega_\gamma(c_{t+1}, d(t))].
\end{aligned}$$

The Bellman function of this optimization equals to $-\ln(\gamma(d(t)))$. Thus, any replacement of $\gamma(d(t))$ by a lower value provides an upper bound on the optimized functional. We get it, by using inequality between the weighted arithmetic and geometric means. Thus,

$$\gamma(d(t)) \geq \exp\left[-\sum_{c_{t+1}\in c^*} {}^{\lfloor U}f(c_{t+1}|d(t))\omega_\gamma(c_{t+1}, d(t))\right].$$

The use of the right-hand side of this inequality in the role $\gamma(d(t))$ gives the claimed result. This estimate coincides with the γ-bound, see Proposition 7.1.8. □

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the practically used state description with the state ϕ.

Algorithm 7.2.3 (Academic advisory system minimizing γ-bound on the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state* ϕ_t*, Chapter 6.*
- *Evaluate the steady state behavior of individual components, Section 7.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if* $\mathring{c} = 0$.
- *Take the non-dangerous component as the ideal pdf offered to the user and stop if* $\mathring{c} = 1$.
- *Specify the user ideal pdf* ${}^{\lfloor U}f(d_{o;t}|d(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1})$ *on the response of the o-system.*
- *Specify the user ideal pf* ${}^{\lfloor U}f(c_t|d(t-1)) \equiv {}^{\lfloor U}f(c_t|\phi_{t-1})$ *on the actions* $c_t \in c^*$ *of the academic p-system.*
- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $\gamma(\phi_\tau) = 1$.

   Iterative mode

   (a) *Evaluate the* <u>*functions*</u>

   $$\omega_\gamma(c_\tau, \phi_{\tau-1}) \equiv \int f(d_\tau|\phi_{\tau-1}, c_\tau) \ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau}, \phi_{\tau-1}, c_\tau)}{\gamma(\phi_\tau)\, {}^{\lfloor U}f(d_{o;\tau}|\phi_{\tau-1})}\right) dd_\tau$$
   $$\gamma(\phi_{\tau-1}) \equiv \exp\left[-\sum_{c_\tau \in c^*} {}^{\lfloor U}f(c_\tau|\phi_{\tau-1})\omega_\gamma(c_\tau, \phi_{\tau-1})\right].$$

   (b) *Continue if* $\tau = t + 1$ *otherwise decrease* $\tau = \tau - 1$ *and go to the step 3a.*
4. *Evaluate the advising strategy*

   $${}^{\lfloor I}f(c_{t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}|\phi_t) \exp[-\omega_\gamma(c_{t+1}, \phi_t)], \quad \text{cf. (7.26)}.$$
5. *Present to the user projections of the ideal pdf*

   $${}^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1} \in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) f(d_{t+1}|\phi_t, c_{t+1}), \text{ where}$$

   the pdf $f(d_{t+1}|\phi_t, c)$ *is the* c*-th estimated component.*
6. *Go to the beginning of* Sequential mode.

Remark(s) 7.2.2

The general γ*-bound can be improved whenever* $\omega(c, d(t))$, $c \in c^*$, *have non-zero constant lifts, say* k_c, *i.e.* $\omega(c, d(t)) = k_c + \tilde{\omega}(c, d(t))$ *Then, the pf* $\propto {}^{\lfloor U}f(c|d(t-1))\exp(-k_c)$ *can be and should be used in approximation of the function* $\gamma(d(t))$ *instead of* ${}^{\lfloor U}f(c|d(t-1))$*: the expressions*

$$\sum_{c \in c^*} {}^{\lfloor U}f(c|d(t-1))\omega(c, d(t)) \quad \text{are replaced by} \quad \sum_{c \in c^*} {}^{\lfloor U}f(c|d(t-1)) \exp\left(-k_c\right) \tilde{\omega}(c, d(t)).$$

This is done systematically in the normal case, see Chapter 8.

For the desirable receding horizon $T > 1$, all proposed variants depend heavily on our ability to evaluate the Bellman function $-\ln(\gamma(\phi))$. This calls for variants based on the strategy of iterations spread in time (IST), Section 4.2.1. For it, the horizon $T = 1$ is expected to be sufficient in majority cases.

The common patch introducing IST strategy in all algorithms looks as follows.

Algorithm 7.2.4 (The common IST patch)
Initial (off line) mode

- $\ldots, \ldots$
- *Parameterize properly* $\gamma(\phi) = \gamma(\vartheta, \phi)$ *by a finite-dimensional parameter* ϑ.
- *Select the initial value* ϑ_0 *so that the general initial condition* $\gamma(\vartheta_0, \phi) = 1$ *is fulfilled.*

Sequential (on line) mode, *running for* $t = 1, 2, \ldots$

$\vdots$

Omit resetting of $\gamma(\cdot)$ *before* Iterative mode.
Iterative mode

$\vdots$

Approximate by $\gamma(\vartheta_t, \phi)$ *the minimum reached after the* t*-th step of* Iterative mode.

Problem 7.2.1 (Extend the set of approximate designs) *The list of approximate designs presented in this chapter should be gradually completed. For instance, the following restriction on the target pdfs in advising*

$$
\begin{aligned}
& {}^{\lfloor U}f(a_t|d(t-1)) \propto {}^{\lfloor I}f(a_t|d(t-1)) \exp\left(-0.5\left|\left|a_t - {}^{\lfloor U}a_t\right|\right|^2\right) \Leftrightarrow \qquad (7.29)\\
\Leftrightarrow \quad & {}^{\lfloor I}f(a_t|d(t-1)) \propto {}^{\lfloor U}f(a_t|d(t-1)) \exp\left(0.5\left|\left|a_t - {}^{\lfloor U}a_t\right|\right|^2\right) \forall t \in t^*,\ a_t \in a_t^*,\ d(t-1) \in d^*(t-1)
\end{aligned}
$$

was considered. It reduces the choice of ${}^{\lfloor U}f(a_t|d(t-1))$ *to the choice of the appropriate weighted norm* $||\cdot||$ *and specification of the target value* ${}^{\lfloor U}a_t$ *of the action* a_t. *The assumption (7.29) assigns low desirable probabilities to the actions far from the given* ${}^{\lfloor U}a_t$.

It can be shown that with this restriction, deterministic strategy arises and a sort of quadratic programming has to be solved for normal mixtures.

Problem 7.2.2 (Use of ω-bound in approximate designs) *The* ω*-bound, Proposition 7.1.9, opens a way for construction of a wide set of approximate designs that are expected to improve the elaborated use of the* γ*-bound.*

7.2.2 Choice of ${}^{\lfloor U}f(c_{t+1}|d(t))$ and its consequences

The pfs $\left\{ {}^{\lfloor U}f(c_t|d(t-1))\right\}_{t\in t^*}$ have an important role in academic and simultaneous designs. They are out of the direct user interest. According to the common approach discussed in Section 5.1.2, they should be left to its fate. This leads to the choice ${}^{\lfloor U}f(c_t|d(t-1)) = \alpha_{c_t}$, where α are estimated component weights. This choice has the following interpretation in γ-bounding of the Bellman function $-\ln(\gamma(d(t))) \equiv$

$$
\begin{aligned}
\equiv \quad & -\ln\left\{\sum_{c_{t+1}\in c^*} \alpha_{c_{t+1}} \exp\left[-\int f(d_{t+1}|d(t), c_{t+1}) \ln\left(\frac{f(d_{t+1}|d(t), c_{t+1})}{\gamma(d(t+1))\, {}^{\lfloor U}f(d_{t+1}|d(t))}\right) d_{t+1}\right]\right\} \leq \\
\leq \quad & \sum_{c_{t+1}\in c^*} \int f(d_{t+1}, c_{t+1}|d(t)) \ln\left(\frac{f(d_{t+1}, c_{t+1}|d(t))}{\gamma(d(t+1))\, {}^{\lfloor U}f(d_{t+1}|d(t))\alpha_{c_{t+1}}}\right) d_{t+1}. \qquad (7.30)
\end{aligned}
$$

Thus, the bound describing the future loss is derived from the objectively estimated pdf $f(d_{t+1}, c_{t+1}|d(t))$ and it is expected to lead to a realistic but conservative approximation.

At the same time, recommended pointers are assumed to change the non-guided situation. Thus, it makes sense to deviate from the complete resignation on the desired ${}^{\lfloor U}f(c_t|d(t-1))$. For instance, a restriction of the support of ${}^{\lfloor U}f(c_t|d(t-1))$ serves us for the exclusion of dangerous components. An active choice of ${}^{\lfloor U}f(c_t|d(t-1))$ opens other useful possibilities that are discussed here.

The low rate of operators actions calls for a low rate of changes of recommended components. This requirement can be respected by the specification

$$\begin{aligned} {}^{\lfloor U}f(c_\tau|d(\tau-1)) &= \delta_{c_\tau,c_{nt+1}} \text{ for } \tau = nt+1,\ldots,n(t+1) \\ {}^{\lfloor U}f(c_{nt+1}|d(nt)) &\equiv \text{a given pdf, where} \end{aligned} \tag{7.31}$$

$n \geq 1$ determines *grouping rate* driven by the rate of advices offered to the operator and $t \in t^*$ counts the groups.

The choice (7.31) means that the recommended pointer is allowed to be changed at most each n-th time moment. This choice provides a special form of the academic design.

Proposition 7.2.4 (Academic design minimizing the KL distance for ${}^{\lfloor U}f(c_t|d(t-1))$ (7.31)) *Let us consider the design of the academic advisory system with actions $c_t \in c^*$ restricted to indices in c^* that point to the non-dangerous components only. Let us also assume that the user pf on recommended pointers is selected according to (7.31). Let us search for the optimal causal advising strategy $d^*(t-1) \to c_t \in c^*$ that minimizes the KL distance of ${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ to the user ideal ${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t}))$ as given in (7.25).*

Then, this strategy has the structure

$${}^{\lfloor I}f(c_\tau|d(\tau-1)) = \delta_{c_\tau,c_{nt+1}} \quad \textit{for } \tau = nt+1,\ldots,n(t+1), \textit{ where} \tag{7.32}$$

${}^{\lfloor I}f(c_{nt+1}|d(nt))$ is given by the following formulas, solved for $t = \mathring{t}, \mathring{t}-1, \ldots, 1$,

$$\begin{aligned} {}^{\lfloor I}f(c_{nt+1}|d(nt)) &= {}^{\lfloor U}f(c_{nt+1}|d(nt)) \frac{\exp[-\omega_\gamma(c_{nt+1}, d(nt))]}{\gamma(d(nt))}, \quad \textit{where} \\ \gamma(d(nt)) &\equiv \sum_{c_{nt+1}\in c^*} {}^{\lfloor U}f(c_{nt+1}|d(nt)) \exp[-\omega_\gamma(c_{tn+1}, d(nt))] \\ \omega_\gamma(c_{nt+1}, d(nt)) &\equiv \frac{1}{n} \sum_{\tau=nt+1}^{n(t+1)} \int f(d_\tau|d(nt), c_{nt+1}) \ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau}, d(nt), c_{nt+1})}{\gamma(d(n(t+1)))\, {}^{\lfloor U}f(d_{o;\tau}|d_o(\tau-1))} \right) dd_\tau \\ \gamma(d(n\mathring{t})) &= 1. \end{aligned} \tag{7.33}$$

The needed pdf $f(d_{o;\tau}|d_{p+;\tau}, d(\tau-1), c)$ is the conditional pdfs of the c-th component $f(d_\tau|d(\tau-1), c)$ obtained in learning of the o-system. The ideal pdf presented to the user has the form

$${}^{\lfloor I}f(d_{\tau+1}|d(\tau)) = \sum_{c_{tn+1}\in c^*} {}^{\lfloor I}f(c_{nt+1}|d(nt)) f(d_{\tau+1}|d(\tau), c_{nt+1}), \quad \tau = nt+1,\ldots,n(t+1), \quad \textit{where} \tag{7.34}$$

the learnt components $f(d_{\tau+1}|d(\tau), c)$ are used.

Proof: The optimized KL distance is infinite if the optimized pf ${}^{\lfloor I}f(c_\tau|d(\tau-1))$ does not have smaller support than the pf ${}^{\lfloor U}f(c_\tau|d(\tau-1))$. This determines the structure of ${}^{\lfloor I}f(c_\tau|d(\tau-1))$. For such a pf, the optimized KL distance can be given the form

$$\begin{aligned} \mathcal{D}\left({}^{\lfloor I}f || {}^{\lfloor U}f \right) \equiv \mathcal{E}\Bigg\{ &\sum_{t\in t^*} \sum_{c_{(tn+1)\_(t+1)n}} \int {}^{\lfloor I}f(d_{(tn+1)\_(t+1)n}, c_{(tn+1)\_(t+1)n}|d(tn)) \times \\ \times \quad &\ln\left(\frac{{}^{\lfloor I}f(d_{(tn+1)\_(t+1)n}, c_{(tn+1)\_(t+1)n}|d(tn))}{{}^{\lfloor U}f(d_{(tn+1)\_(t+1)n}, c_{(tn+1)\_(t+1)n}|d(tn))} \right) dd_{(tn+1)\_(t+1)n} \Bigg\} = \\ = \quad \mathcal{E}\Bigg\{ &\sum_{t\in t^*} \sum_{c_{tn+1}} \sum_{\tau=tn+1}^{(t+1)n} \int f(d_\tau|d(\tau-1), c_{tn+1})\, {}^{\lfloor I}f(c_{tn+1}|d(tn)) \times \end{aligned}$$

$$\times \quad \ln\left(\frac{f(d_\tau|d(\tau-1),c_{tn+1})\,{}^{\lfloor I}f(c_{tn+1}|d(tn))}{{}^{\lfloor U}f(d_\tau|d(\tau-1),c_{tn+1})\,{}^{\lfloor U}f(c_{tn+1}|d(tn))}\right) dd_\tau \Bigg\} =$$

$$= \quad n\mathcal{E}\left\{\sum_{t\in t^*}\sum_{c_{tn+1}} {}^{\lfloor I}f(c_{tn+1}|d(tn))\left[\ln\left(\frac{{}^{\lfloor I}f(c_{tn+1}|d(tn))}{{}^{\lfloor U}f(c_{tn+1}|d(tn))}\right)+\omega(c_{tn+1},d(tn))\right]\right\}$$

$$\omega(c_{tn+1},d(tn)) = \frac{1}{n}\left[\sum_{\tau=tn+1}^{(t+1)n}\int f(d_\tau|d(tn),c_{tn+1})\ln\left(\frac{f(d_\tau|d(\tau-1),c_{tn+1})}{{}^{\lfloor U}f(d_\tau|d(\tau-1),c_{tn+1})}\right) d_\tau\right].$$

With the introduced function $\omega(c_{tn+1}, d(tn))$, we have arrived to the standard expression for the KL distance and we can used the standard dynamic programming to get the claimed result. □

Let us present algorithm corresponding to the proved proposition combined with the receding-horizon certainty-equivalent strategy. As usual, it is written for the practically used state description with the state ϕ.

Algorithm 7.2.5 (Grouped academic design optimizing the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state ϕ_t, Chapter 6.*
- *Evaluate the steady state behavior of individual components, Section 7.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if $\mathring{c} = 0$.*
- *Take the non-dangerous component as the ideal pdf offered to the user and stop if $\mathring{c} = 1$.*
- *Specify the user ideal pdf ${}^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1})$ on the response of the o-system.*
- *Select the grouping rate $n > 1$ and specify the user ideal pdf ${}^{\lfloor U}f(c_{nt+1}|d(nt)) \equiv {}^{\lfloor U}f(c_{nt+1}|\phi_t)$ on the recommended pointers c_t to the non-dangerous components.*
- *Select the length of the receding horizon $T \geq 1$.*

Sequential (on line) mode, *running for $t = 1, 2, \ldots$,*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t + T$ and $\gamma(\phi_\tau) = 1$.*

   Iterative mode

   (a) *Evaluate the functions*

$$\omega_\gamma(c,\phi_{\tau n-1}) \;\equiv\; \frac{1}{n}\sum_{\tilde\tau=\tau n+1}^{(\tau+1)n}\int f(d_{\tilde\tau}|\phi_{\tau n},c)\ln\left(\frac{f(d_{o;\tilde\tau}|d_{p+;\tilde\tau},\phi_{\tau n-1},c)}{\gamma(\phi_{\tau n-1})\,{}^{\lfloor U}f(d_{o;\tilde\tau}|\phi_{\tau n-1})}\right) dd_{\tilde\tau},\; c\in c^*$$

$$\gamma(\phi_{\tau n-1}) \;\equiv\; \sum_{c_{\tau n}\in c^*} {}^{\lfloor U}f(c_{\tau n}|\phi_{\tau n-1})\exp[-\omega_\gamma(c_{\tau n},\phi_{\tau n-1})].$$

   (b) *Continue if $\tau = t + 1$ otherwise decrease $\tau = \tau - 1$ and go to the step 3a.*

4. *Evaluate the advising strategy*

$${}^{\lfloor I}f(c_{nt+1}|\phi_{nt}) = {}^{\lfloor U}f(c_{nt+1}|\phi_{tn})\frac{\exp[-\omega_\gamma(c_{nt+1},\phi_{tn})]}{\gamma(\phi_{nt})}.$$

5. *Present to the user projections of the ideal pdfs*

$$^{\lfloor I}f(d_{o;\tau}|\phi_\tau) = \sum_{c_{nt+1}\in c^*} f(d_{o;\tau}|\phi_{\tau-1}, c_{nt+1})\, ^{\lfloor I}f(c_{nt+1}|\phi_{nt}),\ \tau = nt+1, \ldots n(t+1),\ \textit{where}$$

the pdf $f(d_{o;t+1}|\phi_t, c)$ is derived from the c-th estimated components .

6. *Go to the beginning of* Sequential mode.

The introduced grouping can be used for an alternative, more problem tailored, analysis of dangerous components. The non-weighted version of the conditional KL distance that assumes a fixed selection of the recommended pointers for n steps says what quality can be expected when a single component is recommended for a long time. Thus, dangerous components are those for which

$$\lim_{n\to\infty} \omega(c_{tn+1} = c, d(tn)) = \frac{1}{n}\left[\sum_{\tau=tn+1}^{(t+1)n} \int f(d_\tau|d(tn), c_{tn+1}) \ln\left(\frac{f(d_\tau|d(\tau-1), c_{tn+1})}{^{\lfloor U}f(d_\tau|d(\tau-1), c_{tn+1})}\right) d_\tau\right] \equiv \omega_\infty(c) \tag{7.35}$$

is too large. Moreover, the values $\omega_\infty(\cdot)$ provide an indicator on preferences among components and give the rule how to select user pf $^{\lfloor U}f(c_t|d(t-1))$ even when no grouping is applied. We select

$$^{\lfloor U}f(c_t|d(t-1)) \in \operatorname{Arg}\min_f \left[q\mathcal{E}(\omega_\infty) + \mathcal{D}\left(f||\mathring{c}^{-1}\right)\right],$$

i.e. the pf close to the uniform $f(c) \equiv \mathring{c}^{-1}$ and minimizing expected value of ω_∞. The positive optional weight q defines compromise between these requirements. The pf in question has the form

$$^{\lfloor U}f(c_t|d(t-1)) \propto \exp[-q\omega_\infty(c_t)]. \tag{7.36}$$

Proposition 7.2.3 optimizes a computationally attractive upper bound on the KL distance. Preliminary experiments, however, indicate that the resulting strategy is not sensitive enough to differences in quality of individual components. This sensitivity can be increased by applying the same approximation on grouped advices discussed above. This motivates the following proposition.

Proposition 7.2.5 (Grouped academic advising minimizing γ-bound on the KL distance) *Let us consider the design of the academic advisory system with actions $c_t \in c^*$ restricted to indices in c^* that point to the non-dangerous components only. Let us also assume that the user pf on the recommended pointers is selected according to (7.31). Let us search for the optimal causal advising strategy*

$$d^*(t-1) \to c_t \in c^*$$

that minimizes the γ-bound, Proposition 7.1.8, on the KL distance of $^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ to the user ideal $^{\lfloor U}f(d(\mathring{t}), c(\mathring{t}))$ as given in (7.25).

Then, such a strategy has the structure

$$^{\lfloor I}f(c_\tau|d(\tau-1)) = \delta_{c_\tau, c_{nt+1}}\, ^{\lfloor I}f(c_{nt+1}|d(nt))\ \textit{for}\ \tau = nt+1, \ldots, n(t+1),\ \textit{where}$$

$^{\lfloor I}f(c_{nt+1}|d(nt))$ is given by the following formulas, solved for $t = \mathring{t}, \mathring{t}-1, \ldots, 1$,

$$\begin{aligned}
^{\lfloor I}f(c_{nt+1}|d(nt)) &\propto\ ^{\lfloor U}f(c_{nt+1}|d(nt)) \exp[-\omega_\gamma(c_{nt+1}, d(nt))], \quad \textit{where} \\
\omega_\gamma(c_{nt+1}, d(nt)) &\equiv \frac{1}{n}\sum_{\tau=nt+1}^{n(t+1)} \int f(d_\tau|d(nt), c_{nt+1}) \ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau}, d(nt), c_{nt+1})}{\gamma(d(n(t+1)))\, ^{\lfloor U}f(d_{o;\tau}|d_o(\tau-1))}\right) dd_\tau \\
\gamma(d(nt)) &\equiv \exp\left[-\sum_{c_{nt+1}\in c^*} {^{\lfloor U}f(c_{nt+1}|d(nt))}\omega_\gamma(c_{tn+1}, d(nt))\right] \\
\gamma(d(n\mathring{t})) &= 1.
\end{aligned} \tag{7.37}$$

The needed pdf $f(d_{o;\tau}|d_{p+;\tau}, d(\tau-1), c)$ *is the conditional pdfs of the* c*-th component* $f(d_\tau|d(\tau-1), c)$ *obtained in learning of the o-system. The ideal pdf presented to the user has the form*

$$^{\lfloor I}f(d_{\tau+1}|d(\tau)) = \sum_{c_{tn}\in c^*} {}^{\lfloor I}f(c_{nt+1}|d(nt))f(d_{\tau+1}|d(\tau), c_{nt+1}), \quad \tau = nt+1, \ldots, n(t+1), \quad \text{where}$$

the c*-th learnt component* $f(d_{\tau+1}|d(\tau), c)$ *are used.*

Proof: The result is a straightforward extension of Proposition 7.2.4 when approximating the Bellman function from above through inequality between geometric and arithmetic means as it is done in the proof of Proposition 7.2.3 or in Proposition 7.1.8. □

This proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the practically used state description with the state ϕ.

Algorithm 7.2.6 (Grouped academic advising minimizing γ-bound on the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state* ϕ_t*, Chapter 6.*
- *Evaluate the steady state behavior of individual components, Section 7.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if* $\mathring{c} = 0$.
- *Take the non-dangerous component as the ideal pdf offered to the user and stop if* $\mathring{c} = 1$.
- *Specify the user ideal pdf* $^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1})$ *on the response of the o-system.*
- *Select the grouping rate* $n > 1$ *and specify the user ideal pdf* $^{\lfloor U}f(c_{nt+1}|d(nt)) \equiv {}^{\lfloor U}f(c_{nt+1}|\phi_t)$ *on the recommended pointers* c_t *to the non-dangerous components.*
- *Select the length of the receding horizon equal to* nT *with* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_{nt} *and determine the state vector* ϕ_{nt}.
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting* $\tau = n(t+T)$ *and* $\gamma(\phi_{n\tau}) = 1$.

   Iterative mode

   (a) *Evaluate the* <u>*functions*</u>

   $$\begin{aligned} \omega_\gamma(c, \phi_{\tau n-1}) &\equiv \frac{1}{n}\sum_{\tilde{\tau}=\tau n+1}^{(\tau+1)n} \int f(d_{\tilde{\tau}}|\phi_{\tau n}, c) \ln\left(\frac{f(d_{o;\tilde{\tau}}|d_{p+;\tilde{\tau}}, \phi_{\tau n-1}, c)}{\gamma(\phi_{\tau n-1})\, {}^{\lfloor U}f(d_{o;\tilde{\tau}}|\phi_{\tau n-1})}\right) dd_{\tilde{\tau}}, \ c \in c^* \\ \gamma(\phi_{\tau n-1}) &\equiv \exp\left[-\sum_{c_{\tau n}\in c^*} {}^{\lfloor U}f(c_{\tau n}|\phi_{\tau n-1})\omega_\gamma(c_{\tau n}, \phi_{\tau n-1})\right]. \end{aligned}$$

   (b) *Continue if* $\tau = t+1$ *otherwise decrease* $\tau = \tau - 1$ *and go to the step 3a.*
4. *Evaluate the advising strategy*

$$^{\lfloor I}f(c_{nt+1}|\phi_{nt}) \propto {}^{\lfloor U}f(c_{nt+1}|\phi_{tn})\exp[-\omega_\gamma(c_{nt+1}, \phi_{tn})].$$

5. *Present to the user projections of the ideal pdfs*

$$^{\lfloor I}f(d_{o;\tau}|\phi_\tau) = \sum_{c_{nt+1}\in c^*} f(d_{o;\tau}|\phi_{\tau-1}, c_{nt+1})\, ^{\lfloor I}f(c_{nt+1}|\phi_{nt}),\ \tau = nt+1,\ldots n(t+1),\ \textit{where}$$

the pdf $f(d_{o;t+1}|\phi_t, c)$ *is derived from the* c*-th estimated components .*

6. *Go to the beginning of* Sequential mode.

Problem 7.2.3 (Design and application of grouped advices) *Estimation in the adaptive version of the grouped design should run at the highest data collection rate. The grouped design can be repeated either after* n *real-time steps or repeated in every real time moment, cf. [78].*

Both possibilities have their pros and cons that need a closer look.

7.2.3 Industrial design

The industrial design has seemed to have a limited use. Moreover, it deals with a rather complex and hardly manageable model. For these reasons, we have considered possibility to omit this design completely and to rely on the simultaneous design solved in the next section. There is, however, an important class of problems in which component weights are objectively given. Thus, the industrial design inspected here is a necessary part of the constructed design toolkit.

Proposition 7.2.6 (Industrial design minimizing a bound (7.22) on the KL distance) *Let the joint pdf (7.10)*

$$^{\lfloor I}f(\Delta(\mathring{t}), u_o(\mathring{t})) \equiv \prod_{t\in t^*} \frac{\sum_{c\in c^*} \alpha_c f(\Delta_t|u_{o;t}, d(t-1), c) f(u_{o;t}|d(t-1), c)}{\sum_{c\in c^*} \alpha_c f(u_{o;t}|d(t-1), c)}\, ^{\lfloor I}f(u_{o;t}|d(t-1))$$

be determined by the optional industrial advising strategy described by pdfs $\left\{ ^{\lfloor I}f(u_{o;t}|d(t-1))\right\}_{t\in t^*}$. *The involved mixture with components* $\{f(d_t|u_{o;t}, d(t-1), c) = f(\Delta_t|u_{o;t}, d(t-1), c) f(u_{o;t}|d(t-1), c)\}_{t\in t^*}$ *and their weights* α_c *are assumed to be known (well estimated).*

Let us search for the strategy minimizing the upper bound of the type (7.22) on the KL distance $\mathcal{D}\left(^{\lfloor I}f || ^{\lfloor U}f\right)$ *to the user target pdf*

$$^{\lfloor U}f(d(\mathring{t})) \equiv \prod_{t\in t^*} {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\, ^{\lfloor U}f(u_{o;t}|d_o(t-1))\, ^{\lfloor I}f(\Delta_{p+;t}|d(t-1)).$$

Let us denote

$$f(c|u_{o;t}, d(t-1)) \equiv \frac{\alpha_c f(u_{o;t}|d(t-1), c)}{\sum_{c\in c^*} \alpha_c f(u_{o;t}|d(t-1), c)}. \tag{7.38}$$

Then, the strategy searched for is described by the following formulas, solved for $t = \mathring{t}, \mathring{t}-1, \ldots, 1$,

$$\begin{aligned}
^{\lfloor I}f(u_{o;t}|d(t-1)) &= {}^{\lfloor U}f(u_{o;t}|d(t-1)) \frac{\exp[-\omega_\gamma(u_{o;t}, d(t-1))]}{\gamma(d(t-1))}, \quad \textit{where} \\
\gamma(d(t-1)) &\equiv \int {}^{\lfloor U}f(u_{o;t}|d(t-1)) \exp[-\omega_\gamma(u_{o;t}, d(t-1))]\, du_{o;t} \\
\omega_\gamma(u_{o;t}, d(t-1)) &\equiv \sum_{c\in c^*} f(c|u_{o;t}, d(t-1)) \omega_\gamma(c, u_{o;t}, d(t-1)) \qquad (7.39) \\
\omega_\gamma(c, u_{o;t}, d(t-1)) &\equiv \ln\left(\frac{f(c|u_{o;t}, d(t-1))}{\alpha_c}\right) + \\
&+ \int f(\Delta_t|u_{o;t}, d(t-1), c) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c)}{\gamma(d(t))\, ^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))}\right) d\Delta_t \\
\gamma(d(\mathring{t})) &\equiv 1.
\end{aligned}$$

Proof: We set the correspondence with quantities used in Proposition 7.1.7: $d(t-1)$ is a fixed condition, $x \Leftrightarrow \Delta_t$, $y \Leftrightarrow u_{o;t}$. Then, we can write the optimized upper bound as the expected value of the loss function

$$\sum_{t \in t^*} \int {}^{\lfloor I}f(u_{o;t}|d(t-1)) \left[\ln\left(\frac{{}^{\lfloor I}f(u_{o;t}|d(t-1))}{{}^{\lfloor U}f(u_{o;t}|d(t-1))} \right) + \omega(u_{o;t}, d(t-1)) \right] du_{o;t}, \quad \text{with}$$

$$\begin{aligned}
\omega_\gamma(u_{o;t}, d(t-1)) &\equiv \sum_{c \in c^*} f(c|u_{o;t}, d(t-1)) \omega_\gamma(c, u_{o;t}, d(t-1)) \\
\omega_\gamma(c, u_{o;t}, d(t-1)) &\equiv \ln\left(\frac{f(c|u_{o;t}, d(t-1))}{\alpha_c} \right) + \\
&+ \int f(\Delta_t|u_{o;t}, d(t-1), c) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c)}{{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))} \right) d\Delta_t.
\end{aligned}$$

The standard backward induction for $t = \mathring{t}$ gives

$$^{\lfloor I}f(u_{o;\mathring{t}}|d(\mathring{t}-1)) \propto {}^{\lfloor U}f(u_{o;\mathring{t}}|d(\mathring{t}-1)) \exp[-\omega(u_{o;\mathring{t}}, d(\mathring{t}-1))]$$

and the minimum value $-\ln(\gamma(d(\mathring{t}-1)))$ transferred to the further step with

$$\gamma(d(\mathring{t}-1)) = \int {}^{\lfloor U}f(u_{o;\mathring{t}}|d(\mathring{t}-1)) \exp[-\omega(u_{o;\mathring{t}}, d(\mathring{t}-1))]\, du_{o;\mathring{t}}.$$

For a generic t, we re-define

$$\omega_\gamma(c, u_{o;t}, d(t-1)) \equiv \int f(\Delta_t|u_{o;t}, d(t-1), c) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c)}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))} \right) d\Delta_t,$$

in the description of $\omega(u_{o;t}, d(t-1))$. It shows that the computation has the same structure as for $t = \mathring{t}$. The full conformity is reached when defining $\gamma(d(\mathring{t})) = 1$. $\square$

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the state description with the state ϕ.

Algorithm 7.2.7 (Industrial design minimizing the bound (7.22) on the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state ϕ_t, Chapter 6.*
- *Specify the user ideal pdf on the response of the o-system*

$$^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1)).$$

- *Select the length of the receding horizon $T \geq 1$.*

Sequential (on line) mode, *running for $t = 1, 2, \ldots,$*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t + T$ and $\gamma(\phi_\tau) = 1$.*

   Iterative mode

   (a) *Evaluate the functions*

$$\begin{aligned}
f(c|u_{o;\tau}, \phi_{\tau-1}) &\equiv \frac{\alpha_c f(u_{o;\tau}|\phi_{\tau-1}, c)}{\sum_{c \in c^*} \alpha_c f(u_{o;\tau}|\phi_{\tau-1}, c)} \\
\omega_\gamma(c, u_{o;\tau}, \phi_{\tau-1}) &\equiv \ln\left(\frac{f(c|u_{o;\tau}, \phi_{\tau-1})}{\alpha_c} \right) +
\end{aligned}$$

$$
\begin{aligned}
& + \int f(\Delta_\tau | u_{o;\tau}, \phi_{\tau-1}, c) \ln\left(\frac{f(\Delta_{o;\tau}|\Delta_{p+;\tau}, u_{o;\tau}, \phi_{\tau-1}, c)}{\gamma(\phi_\tau)\ {}^{\lfloor U}f(\Delta_{o;\tau}|\phi_{\tau-1})}\right) d\Delta_\tau \quad \textit{and} \\
\omega_\gamma(u_{o;\tau}, \phi_{\tau-1}) \ & \equiv \ \sum_{c \in c^*} f(c|u_{o;\tau}, \phi_{\tau-1}) \omega_\gamma(c, u_{o;\tau}, \phi_{\tau-1}) \\
\gamma(\phi_{\tau-1}) \ & \equiv \ \int {}^{\lfloor U}f(u_{o;\tau}|\phi_{\tau-1}) \exp[-\omega_\gamma(u_{o;\tau}, \phi_{\tau-1})]\, du_{o;\tau}
\end{aligned}
$$

(b) *Continue if $\tau = t+1$, otherwise decrease $\tau = \tau - 1$ and go to the step 3a.*

4. *Evaluate the industrial strategy*

$$
{}^{\lfloor I}f(u_{o;t+1}|\phi_t) \propto {}^{\lfloor U}f(u_{o;t+1}|\phi_t) \exp[-\omega_\gamma(u_{o;t+1}, \phi_t)].
$$

5. *Present to the user projections of the ideal pdf*

$$
{}^{\lfloor I}f(d_{o;t+1}|\phi_t) \ = \ \frac{\sum_{c \in c^*} \alpha_c f(\Delta_{o;t}|u_{o;t+1}, \phi_t, c) f(u_{o;t+1}|\phi_t, c)}{\sum_{c \in c^*} \alpha_c f(u_{o;t+1}|\phi_t, c)} \ {}^{\lfloor I}f(u_{o;t+1}|\phi_t), \quad \textit{where}
$$

the pdf $f(\Delta_{o;t}|u_{o;t+1}, \phi_t, c)$, $f(u_{o;t+1}|\phi_t, c)$ are derived from c-th learn mixture component.

6. *Go to the beginning of* Sequential mode.

Remark(s) 7.2.3

1. *Various bounds have been and can be derived. The bound (7.22) seems to be the best one of those tried. Alternatives have to be considered in future anyway.*

2. *Use of γ- or ω-bounds, Propositions 7.1.8 7.1.9 is probably unnecessary as the used bound (7.22) includes them anyway.*

3. *The maximum probable choice of the recognizable actions is possible.*

7.2.4 Simultaneous academic and industrial design

The separation of academic and industrial designs is often motivated pedagogically only. Whenever possible, the optimization of c_t and $u_{o;t}$ should be performed simultaneously. It may give much better results. The recognizable actions may change completely character of the analyzed components, for instance, to change dangerous components into non-dangerous ones. Even less drastic changes reflected in the improved quality are worth of considering. Moreover, the simultaneous design addressed here is simpler than the industrial one.

Proposition 7.2.7 (Simultaneous design minimizing the KL distance) *Let the joint pdf*

$$
{}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t})) \equiv \prod_{t \in t^*} f(\Delta_t|u_{o;t}, d(t-1), c_t)\ {}^{\lfloor I}f(u_{o;t}|d(t-1), c_t)\ {}^{\lfloor I}f(c_t|d(t-1))
$$

be determined by the optional simultaneous advising strategy described by pdfs

$$
\left\{ {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) = {}^{\lfloor I}f(u_{o;t}|d(t-1), c_t)\ {}^{\lfloor I}f(c_t|d(t-1)) \right\}_{t \in t^*}.
$$

The pdfs $\{f(\Delta_t|u_{o;t}, d(t-1), c_t)\}_{t \in t^*}$ are determined by the components of the known (well estimated) mixture. Let us search for the strategy minimizing the KL distance $\mathcal{D}\left({}^{\lfloor I}f || {}^{\lfloor U}f\right)$ to the user target pdf

$$
{}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) \equiv \prod_{t \in t^*} {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\ {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\ {}^{\lfloor I}f(\Delta_{p+;t}|d(t-1))\ {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)).
$$

Then, the optimal strategy is described by the following formulas, solved for $t = \mathring{t}, \mathring{t}-1, \ldots, 1,$

$$\begin{aligned}
{}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) &= {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)) \frac{\exp[-\omega_\gamma(c_t, u_{o;t}, d(t-1))]}{\gamma(d(t-1))}, \quad \textit{where} \\
\gamma(d(t-1)) &\equiv \sum_{c_t \in c^*} \int {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)) \exp[-\omega_\gamma(c_t, u_{o;t}, d(t-1))]\, du_{o;t} \\
\omega_\gamma(c_t, u_{o;t}, d(t-1)) &\equiv \int f(\Delta_t|u_{o;t}, d(t-1), c_t) \ln\left[\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d(t-1))}\right] d\Delta_t \qquad (7.40) \\
\gamma(d(\mathring{t})) &\equiv 1.
\end{aligned}$$

Proof: Let us assume that the Bellman function in the addressed problem equals to $-\ln(\gamma(d(t)))$. For $t = \mathring{t}$, it holds with $\gamma(d(\mathring{t})) = 1$. A generic optimization step should verify

$$\begin{aligned}
&-\ln(\gamma(d(t-1))) \overbrace{\equiv}^{?} \min_{\{{}^{\lfloor I}f(c_t, u_{o;t}|d(t-1))\}} \sum_{c_t \in c^*} \int f(\Delta_t|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) \\
&\quad \ln\left[\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))}\right] dd_t \underbrace{=}_{(7.40)} \\
&= \sum_{c_t \in c^*} \int {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) \\
&\quad \left[\omega_\gamma(c_t, u_{o;t}, d(t-1)) + \ln\left(\frac{{}^{\lfloor I}f(c_t, u_{o;t}|d(t-1))}{{}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))}\right)\right] du_{o;t} \quad \text{with} \\
&\omega_\gamma(c_t, u_{o;t}, d(t-1)) \equiv \int f(\Delta_t|u_{o;t}, d(t-1), c_t) \ln\left[\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d(t-1))}\right] d\Delta_t.
\end{aligned}$$

The minimizing ideal pdf of pointers and recognizable actions is

$$\begin{aligned}
{}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) &= \frac{{}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|d(t-1)) \exp[-\omega_\gamma(c_t, u_{o;t}, d(t-1))]}{\gamma(d(t-1))} \\
\gamma(d(t-1)) &\equiv \sum_{c_t \in c^*} \int {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d(t-1)) \exp[-\omega_\gamma(c_t, u_{o;t}, d(t-1))]\, du_{o;t}.
\end{aligned}$$

The achieved minimum reproduces the assumed form with

$$\gamma(d(t-1)) = \sum_{c_t \in c^*} \int {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1)) \exp[-\omega_\gamma(c_t, u_{o;t}, d(t-1))]\, du_{o;t}.$$

□

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the state description with the state ϕ.

Algorithm 7.2.8 (Simultaneous design optimizing the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state* ϕ_t, *Chapter 6.*
- *Specify the user ideal pdf on the response of the o-system*

$${}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1), c_t)\, {}^{\lfloor U}f(u_{o;t}, c_t|d_o(t-1)) \equiv {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, \phi_{t-1})\, {}^{\lfloor U}f(u_{o;t}, c_t|\phi_{t-1}).$$

- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t + T$ and $\gamma(\phi_\tau) = 1$.*
   Iterative mode
   (a) *Evaluate the functions*

$$\omega_\gamma(c_\tau, u_{o;\tau}, \phi_{\tau-1}) \equiv \int f(\Delta_\tau | u_{o;\tau}, \phi_{\tau-1}, c_\tau) \ln\left(\frac{f(\Delta_{o;\tau}|u_{o;\tau}, \Delta_{p+;\tau}, \phi_{\tau-1}, c_\tau)}{\gamma(\phi_\tau)\, {}^{\lfloor U}f(\Delta_{o;\tau}|u_{o;\tau}, \phi_{\tau-1})} \right) d\Delta_\tau$$
$$\gamma(\phi_{\tau-1}) \equiv \sum_{c_\tau \in c^*} \int {}^{\lfloor U}f(u_{o;\tau}|\phi_{\tau-1})\, {}^{\lfloor U}f(c_\tau|u_{o;\tau}, \phi_{\tau-1}) \exp[-\omega_\gamma(c_\tau, u_{o;\tau}, \phi_{\tau-1})]\, du_{o;\tau}.$$

   (b) *Continue if $\tau = t + 1$, otherwise decrease $\tau = \tau - 1$ and go to the step 3a.*
4. *Evaluate the optimal simultaneous strategy*

$${}^{\lfloor I}f(c_{t+1}, u_{o;t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}|u_{o;t+1}, \phi_t)\, {}^{\lfloor U}f(u_{o;t+1}|\phi_t) \exp[-\omega_\gamma(c_{t+1}, u_{o;t+1}, \phi_t)].$$

5. *Present to the user projections of the ideal pdf*

$${}^{\lfloor I}f(d_{o;t+1}|\phi_t) = \sum_{c_{t+1} \in c^*} f(\Delta_{o;t}|u_{o;t+1}, \phi_t, c_{t+1})\, {}^{\lfloor I}f(c_{t+1}, u_{o;t+1}|\phi_t), \text{ where}$$

   the pdf $f(\Delta_{o;t}|u_{o;t+1}, \phi_t, c_{t+1})$ is derived from the c_{t+1}-th learn mixture component.
6. *Go to the beginning of* Sequential mode.

The additional minimization of the achieved minimum of the KL distance with respect to the optional ${}^{\lfloor U}f(c_t|d(t-1))$ does not simplify the evaluation. For this reason, no counterpart of Proposition 7.2.2 is presented. However, the replacement of $\gamma(d(t))$ in the definition of ω_γ by a lower bound on $\gamma(d(t))$ simplifies the design substantially.

Proposition 7.2.8 (Simultaneous advising minimizing γ-bound the KL distance) *Let the joint pdf*

$${}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t})) \equiv \prod_{t \in t^*} f(\Delta_t|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(u_{o;t}|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1))$$

be determined by the optional simultaneous advising strategy described by pdfs

$$\left\{ {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) = {}^{\lfloor I}f(u_{o;t}|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1)) \right\}_{t \in t^*}.$$

The pdfs $\{f(\Delta_t|u_{o;t}, d(t-1), c_t)\}_{t \in t^*}$ are derived from the components of the known (well estimated) mixture. Let us search for the strategy approximately minimizing the KL distance $\mathcal{D}\left({}^{\lfloor I}f||\,{}^{\lfloor U}f\right)$ to the user ideal pdf

$${}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) \equiv \prod_{t \in t^*} {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor I}f(\Delta_{p+;t}|d(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)).$$

Then, the following strategy minimizes the γ-bound, Proposition 7.1.8, on the KL distance

$${}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) \propto {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)) \exp[-\omega_\gamma(c_t, u_{o;t}, d(t-1))], \text{ where}$$
$$\omega_\gamma(u_{o;t}, c_t, d(t-1)) \equiv \int f(\Delta_t|u_{o;t}, d(t-1), c_t) \ln\left[\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d(t-1))} \right] d\Delta_t \quad (7.41)$$
$$\gamma(d(t-1)) \equiv \int {}^{\lfloor U}f(u_{o;t}|d_o(t-1)) \exp\left[-\sum_{c_t \in c^*} {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))\omega_\gamma(c_t, u_{o;t}, d(t-1)) \right] du_{o;t}$$
$$\gamma(d(\mathring{t})) \equiv 1,\ t = \mathring{t}, \mathring{t}-1, \ldots, 1.$$

Proof: The optimal strategy is determined by Proposition 7.2.7 and its Bellman function equals to $-\ln(\gamma(d(t)))$. Thus, it is sufficient to find a lower bound on $\gamma(d(t))$ in order to care about minimization of an upper bound on the KL distance. The inequality between weighted arithmetic and geometric means is used here, similarly as in Proposition 7.2.3.

Notice that the γ-bound, Proposition 7.1.8, is applied to the marginal pf on pointers $c \in c^*$ only. □

The above proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for systems with the state ϕ.

Algorithm 7.2.9 (Simultaneous advising optimizing γ-bound on the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state ϕ_t, Chapter 6.*
- *Specify the user ideal pdf on the response of the o-system*

$$^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1), c_t)\, ^{\lfloor U}f(u_{o;t}, c_t|d_o(t-1)) \equiv \, ^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, \phi_{t-1})\, ^{\lfloor U}f(u_{o;t}, c_t|\phi_{t-1}).$$

- *Specify the user ideal pdf $^{\lfloor U}f(c_t|d(t-1)) \equiv \, ^{\lfloor U}f(c_t|\phi_{t-1})$ on the actions c_t of the academic part of the p-system.*
- *Select the length of the receding horizon $T \geq 1$.*

Sequential (on line) mode, *running for $t = 1, 2, \ldots$,*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t + T$ and $\gamma(\phi_\tau) = 1$.*

   Iterative mode

   (a) *Evaluate the functions*

$$\omega_\gamma(c_\tau, u_{o;\tau}, \phi_{\tau-1}) \equiv \int f(\Delta_\tau|u_{o;\tau}, \phi_{\tau-1}, c_\tau) \ln\left(\frac{f(\Delta_{o;\tau}|u_{o;\tau}, \Delta_{p+;\tau}, \phi_{\tau-1}, c_\tau)}{\gamma(\phi_\tau)\, ^{\lfloor U}f(\Delta_{o;\tau}|u_{o;\tau}, \phi_{\tau-1})}\right) d\Delta_\tau$$

$$\gamma(\phi_{\tau-1}) \equiv \int \, ^{\lfloor U}f(u_{o;\tau}|\phi_{\tau-1}) \exp\left[-\sum_{c_\tau \in c^*} \, ^{\lfloor U}f(c_\tau|u_{o;\tau}, \phi_{\tau-1})\omega_\gamma(c_\tau, u_{o;\tau}, \phi_{\tau-1})\right] du_{o;\tau}.$$

   (b) *Continue if $\tau = t + 1$, otherwise decrease $\tau = \tau - 1$ and go to the step 3a.*

4. *Evaluate the advising strategy*

$$^{\lfloor I}f(c_{t+1}, u_{o;t+1}|\phi_t) \propto \, ^{\lfloor U}f(c_{t+1}|u_{o;t+1}, \phi_t)\, ^{\lfloor U}f(u_{o;t+1}|\phi_t) \exp[-\omega_\gamma(c_{t+1}, u_{o;t+1}, \phi_t)].$$

5. *Present to the user projections of the ideal pdf*

$$^{\lfloor I}f(d_{o;t+1}|\phi_t) = \sum_{c_{t+1} \in c^*} f(\Delta_{o;t}|u_{o;t+1}, \phi_t, c_{t+1})\, ^{\lfloor I}f(c_{t+1}, u_{o;t+1}|\phi_t).$$

6. *Go to the beginning of* Sequential mode.

Remark(s) 7.2.4

1. *Grouped variant, mimic to Algorithm 7.2.6, can be simply derived. Its use seems to be desirable.*
2. *Other design variants, like maximum probable one, are possible and worth of inspecting.*

7.3 Interaction with operator

Priority p_t and signaling s_t actions are main tools for communication with the operator. Their approximate fully probabilistic design is addressed here.

7.3.1 Assigning priorities

When assigning priorities, we assume that the operator has been given a full go for making recommended actions. It means that the signaling action $s_t = 1$, see Section 7.3.2, and the model relating the presentation actions p_t to the response of the o-system reduces to

$$\begin{aligned} {}^{\lfloor I}f(d_t|p_t, d(t-1)) &\equiv f(d_t|d(t-1)) \frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))}, \text{ where} \qquad (7.42) \\ f(d_t|d(t-1)) &= \sum_{c \in c^*} \alpha_c f(d_t|d(t-1), c) \text{ is the learnt mixture and} \\ {}^{\lfloor I}f(d_t|d(t-1)) &= \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1))\, {}^{\lfloor I}f(d_t|d(t-1), c_t) \end{aligned}$$

is the ideal pdf resulting from the academic, industrial or simultaneous design.

Data $d_{p+;t}$ are not influenced directly by priorities as we cannot show the surplus p-data to the operator. Thus, even the extension of the true user ideal pdf

$$\begin{aligned} {}^{\lfloor U}f(d_t|p_t, d(t-1)) &\equiv {}^{\lfloor I}f(d_{p+;t}|p_t, d(t-1))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1)) = \\ &= {}^{\lfloor I}f(d_{p+;t}|d(t-1))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_t|d(t-1)) \end{aligned}$$

does not depend on the presentation actions p_t.

The conditional KL distance, the key quantity in the fully probabilistic design, has the form

$$\begin{aligned} \omega_\gamma(p_t, d(t-1)) &\equiv \int {}^{\lfloor I}f(d_t|p_t, d(t-1)) \ln \left[\frac{{}^{\lfloor I}f(d_t|p_t, d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))} \right] dd_t \underbrace{=}_{(7.42)} \\ &= \int f(d_t|d(t-1)) \frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))} \ln \left[\frac{f(d_t|d(t-1))\, {}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))\, {}^{\lfloor U}f(d_t|d_o(t-1))} \right] dd_t. \end{aligned}$$

The rational form (7.42) of the model ${}^{\lfloor I}f(d_t|p_t, d(t-1))$ implies that an upper bound on the conditional KL distance has to be constructed in order to get a loss function that can be effectively minimized. Essentially, Proposition 7.1.7 is tailored and complemented by a single use of the Jensen inequality, Proposition 7.1.6.

Proposition 7.3.1 (J-like upper bound on the conditional KL distance in presentation) *Let us consider that the academic, industrial or simultaneous design of the advisory system has provided the strategy $d^*(t-1) \to (c_t^*, u_{o;t}^*)$ determining the ideal pdf*

$$ {}^{\lfloor I}f(d_t|d(t-1)) = \sum_{c_t \in c^*} {}^{\lfloor I}f(d_t|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1)). $$

The presentation strategy $\{f(p_t|d(t-1))\}_{t \in t^*}$ *is inspected, assuming the influence of the presentation advices p_t through the model (7.42). Let us denote*

$$ f(c|d_{p_t;t}, d(t-1)) = \frac{\alpha_c f(d_{p_t;t}|d(t-1), c)}{\sum_{c \in c^*} \alpha_c f(d_{p_t;t}|d(t-1))}, \forall c \in c^*. \qquad (7.43) $$

Let us assume that the basic assumption (7.20) of Proposition 7.1.6 is met with a constant $K \geq 1$. Let $\bar{p}_t$ consist of indices of the data record d_t in $\{1, 2, \ldots, \mathring{d}\}$ not included in p_t and

$$ \omega(c, d_{p_t;t}, d(t-1)) \equiv \ln \left(\frac{f(c|d_{p_t;t}, d(t-1))}{\alpha_c} \right) + \qquad (7.44) $$

$$
\begin{aligned}
& + \int f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c) \ln\left(\frac{f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c)}{{}^{\lfloor U}f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1))}\right) dd_{\bar{p}_t;t} \\
\omega(p_t, d_{p_t;t}, d(t-1)) \equiv & \sum_{c\in c^*} f(c|d_{p_t;t}, d(t-1))\omega(c, d_{p_t;t}, d(t-1)).
\end{aligned}
$$

Then,

$$
\begin{aligned}
\omega(p_t, d(t-1)) \equiv & \int f(d_t|d(t-1)) \frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))} \\
& \ln\left[\frac{f(d_t|d(t-1))\, {}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))f(d_{p_t;t}|d(t-1))}\right] dd_t \leq \bar{\omega}(p_t, d(t-1)) \\
\bar{\omega}(p_t, d(t-1)) \equiv & \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|d(t-1)) \int {}^{\lfloor I}f(d_{p_t;t}|d(t-1), c_t) \\
& \left[\ln\left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d_{p+;t}, d(t-1), c_t)}{{}^{\lfloor U}f(d_{p_t;t}|d_o(t-1))}\right) + \sum_{c\in c^*} f(c|d_{p_t;t}, d(t-1))\omega(c, d_{p_t;t}, d(t-1))\right] dd_{p_t;t}
\end{aligned} \tag{7.45}
$$

Proof: For a fixed p_t, we use the following correspondence with quantities of Proposition 7.1.7

$$
\begin{aligned}
d(t-1) & \quad \text{is a fixed condition} \\
x & \Leftrightarrow d_{\bar{p}_t;t} \\
y & \Leftrightarrow d_{p_t;t} \\
f(x,y) & \Leftrightarrow f(d_t|d(t-1)) \\
f(y) & \Leftrightarrow f(d_{p_t;t}|d(t-1)) \\
{}^{\lfloor I}f(y) & \Leftrightarrow {}^{\lfloor I}f(d_{p_t;t}|d(t-1)) \\
{}^{\lfloor U}f(x,y) & \Leftrightarrow {}^{\lfloor U}f(d_t|d(t-1)) \underbrace{\equiv}_{(5.6)} {}^{\lfloor U}f(d_{o;t}|d_o(t-1))\, {}^{\lfloor I}f(d_{p+;t}|d(t-1)).
\end{aligned}
$$

It gives, cf. (7.22),

$$
\begin{aligned}
\omega(p_t, d(t-1)) \equiv & \int f(d_t|d(t-1)) \frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{f(d_{p_t;t}|d(t-1))} \ln\left(\frac{f(d_t|d(t-1))\, {}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{{}^{\lfloor U}f(d_t|d(t-1))f(d_{p_t;t}|d(t-1))}\right) dd_t \leq \\
\leq & \int {}^{\lfloor I}f(d_{p_t;t}|d(t-1)) \left[\ln\left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{{}^{\lfloor U}f(d_{p_t;t}|d(t-1))}\right) + \omega(d_{p_t;t}, d(t-1))\right] dd_{p_t;t} \quad \text{with} \\
\omega(d_{p_t;t}, d(t-1)) \equiv & \sum_{c\in c^*} f(c|d_{p_t;t}, d(t-1))\omega(c, d_{p_t;t}, d(t-1)) \\
\omega(c, d_{p_t;t}, d(t-1)) \equiv & \ln\left(\frac{f(c|d_{p_t;t}, d(t-1))}{\alpha_c}\right) + \\
& + \int f(d_{\bar{p}_t;t}|d_{p_t}, d(t-1), c) \ln\left(\frac{f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c)}{{}^{\lfloor U}f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1))}\right) dd_{\bar{p}_t;t}.
\end{aligned}
$$

The obtained upper bound still contains logarithms of the mixture ${}^{\lfloor I}f(d_{p_t;t}|d(t-1))$ in the third term. The use of the Jensen inequality extended according to Proposition 7.1.6 implies

$$
\begin{aligned}
& \int {}^{\lfloor I}f(d_{p_t;t}|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d(t-1))}{{}^{\lfloor U}f(d_{p_t;t}|d(t-1))}\right) dd_{p_t,t} \leq \\
\leq & \sum_{c_t\in c^*} \int {}^{\lfloor I}f(c_t|d(t-1))\, {}^{\lfloor I}f(d_{p_t;t}|d(t-1), c_t) \ln\left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d_{p+;t}, d(t-1), c_t)}{{}^{\lfloor U}f(d_{p_t;t}|d(t-1))}\right) dd_{p_t,t}.
\end{aligned}
$$

Inserting this estimate into the upper bound on $\omega(p_t, d(t-1))$, using the fact that $\sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|d(t-1)) = 1$ and the observation that the surplus p-data $d_{p+;t}$ are not presented, we get

$$
\omega(p_t, d(t-1)) \leq \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|d(t-1)) \int {}^{\lfloor I}f(d_{p_t;t}|d(t-1), c_t) \times
$$

$$\times \left[\ln \left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d_{p+;t}, d(t-1), c_t)}{{}^{\lfloor U}f(d_{p_t;t}|d_o(t-1))} \right) + \sum_{c \in c^*} f(c|d_{p_t;t}, d(t-1))\omega(c, d_{p_t;t}, d(t-1)) \right] dd_{p_t;t},$$

which is the claimed result. □

Using the found upper bound as the loss function in the fully probabilistic design, Proposition 7.1.4, we get directly presentation strategy searched for.

Proposition 7.3.2 (Presentation minimizing the upper bound (7.45)) *Let us consider that the academic, industrial or simultaneous design of the advisory system has provided the strategy* $d^*(t-1) \to (c_t^*, u_{o;t}^*)$ *determining the ideal pdf*

$$^{\lfloor I}f(d_t|d(t-1)) = \sum_{c_t \in c^*} {}^{\lfloor I}f(d_t|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1)).$$

Let the signaling strategy make the operator fully alert, i.e. signaling actions $s(\mathring{t}) \equiv 1$.

Let us assume that there is constant $K \geq 1$ *and pdf* $g(d_{p+;t}|d(t-1))$ *such that*

$$^{\lfloor I}f(d_{p+;t}|d(t-1), c) \leq K g(d_{p+;t}|d(t-1)), \forall c \in c^* \text{ and almost all data in arguments, cf. Proposition 7.1.6.} \tag{7.46}$$

Let us define for each $p_t \in p^*$

$$f(c|d_{p_t;t}, d(t-1)) = \frac{\alpha_c f(d_{p_t;t}|d(t-1), c)}{\sum_{c \in c^*} \alpha_c f(d_{p_t;t}|d(t-1))} \leq 1, \forall c \in c^*. \tag{7.47}$$

Let us specify user pf ${}^{\lfloor U}f(p_t|d(t-1))$ *on the set of possible presentation actions* $p^* \equiv \{[p_1, \ldots, p_{\mathring{p}}]\}$ *with* p_j *being a unique index in the set of indices of the data records* d_o.

The presentation strategy that minimizes the upper bound on the KL distance, implied by the inequality given in Proposition 7.1.7, is described by the following algorithm, in which $\bar{p}_t$ *mark indices of* d_t *in* $\{1, \ldots, \mathring{d}\}$ *excluded from* p_t,

$$\begin{aligned}
{}^{\lfloor I}f(p_t|d(t-1)) &\equiv \frac{{}^{\lfloor U}f(p_t|d(t-1)) \exp[-\omega_\gamma(p_t, d(t-1))]}{\gamma(d(t-1))} \qquad (7.48)\\
\gamma(d(t-1)) &\equiv \sum_{p_t \in p^*} {}^{\lfloor U}f(p_t|d(t-1)) \exp[-\omega_\gamma(p_t, d(t-1))]\\
\omega_\gamma(p_t, d(t-1)) &\equiv \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1)) \int {}^{\lfloor I}f(d_{p_t;t}|d(t-1), c_t) \times\\
&\times \left[\ln \left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d_{p+;t}, d(t-1), c_t)}{{}^{\lfloor U}f(d_{p_t;t}|d_o(t-1))} \right) + \sum_{c \in c^*} f(c|d_{p_t;t}, d(t-1))\omega(c, d_{p_t;t}, d(t-1)) \right] dd_{p_t;t}\\
\omega(c, d_{p_t;t}, d(t-1)) &\equiv \ln \left(\frac{f(c|d_{p_t;t}, d(t-1))}{\alpha_c} \right) +\\
&+ \int f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c) \ln \left(\frac{f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c)}{\gamma(d(t))\, {}^{\lfloor U}f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1))} \right) dd_{\bar{p}_t;t}\\
\gamma(d(\mathring{t})) &\equiv 1.
\end{aligned}$$

The solution is evaluated against the time course, starting at $t = \mathring{t}$.

Proof: The conditional KL distance is just replaced by weighted KL distance arising in dynamic programming. It preserves non-negativity of the original definition that was used in deriving the used upper bound. □

The above proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the practically used state description with the state ϕ.

Algorithm 7.3.1 (Design of priorities optimizing the bound (7.45) on the KL distance)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state* ϕ_t*, Chapter 6.*
- *Specify the user ideal pdf on* $d_{o;t}, u_{o;t}, c_t, p_t$.
- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots$,

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Design academic, industrial or simultaneous strategy generating the optimal* $c_t, u_{o;t}$ *and thus generating on* d_t^* *the ideal pdf*
$$^{\lfloor I}f(d_t|\phi_{t-1}) = \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1}) \, {}^{\lfloor I}f(d_t|\phi_{t-1}, c_t).$$
4. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $\gamma(\phi_\tau) = 1$.

   Iterative mode

   (a) *Evaluate for each* $p_\tau \in p^*$ *the functions*
$$\begin{aligned}
f(c|d_{p_\tau;\tau}, \phi_{\tau-1}) &\equiv \frac{\alpha_c f(d_{p_\tau;\tau}|\phi_{\tau-1}, c)}{\sum_{c\in c^*} \alpha_c f(d_{p_\tau;\tau}|\phi_{\tau-1}, c)} \\
\gamma(\phi_{\tau-1}) &\equiv \sum_{p_\tau \in p^*} {}^{\lfloor U}f(p_\tau|\phi_{\tau-1}) \exp[-\omega_\gamma(p_\tau, \phi_{\tau-1})] \\
\omega_\gamma(p_\tau, \phi_{\tau-1}) &\equiv \sum_{c_\tau \in c^*} {}^{\lfloor I}f(c_\tau|\phi_{\tau-1}) \int {}^{\lfloor I}f(d_{p_\tau;\tau}|\phi_{\tau-1}, c_\tau) \\
&\left[\ln\left(\frac{^{\lfloor I}f(d_{p_\tau;\tau}|d_{p+;\tau}, \phi_{\tau-1}, c_\tau)}{^{\lfloor U}f(d_{p_\tau;\tau}|\phi_{\tau-1})}\right) + \sum_{c\in c^*} f(c|d_{p_\tau;\tau}, \phi_{\tau-1}) \omega(c, d_{p_\tau;\tau}, \phi_{t-1})\right] dd_{p_\tau;\tau} \\
\omega(c, d_{p_\tau;\tau}, \phi_{\tau-1}) &\equiv \ln(\left(\frac{f(c|d_{p_\tau;\tau}, \phi_{\tau-1})}{\alpha_c}\right) + \\
&+ \int {}^{\lfloor I}f(d_{\bar{p}_\tau;\tau}|d_{p_\tau;\tau}, \phi_{\tau-1}, c) \ln\left(\frac{^{\lfloor I}f(d_{\bar{p}_\tau;\tau}|d_{p_\tau;\tau}\phi_{\tau-1}, c)}{\gamma(\phi_\tau)\, {}^{\lfloor U}f(d_{\bar{p}_\tau;\tau}|d_{p_\tau;\tau}, \phi_{\tau-1})}\right) dd_{\bar{p}_\tau;\tau}
\end{aligned}$$

   (b) *Continue if* $\tau = t + 1$*, otherwise decrease* $\tau = \tau - 1$ *and go to the step 4a.*
5. *Generate a sample* $\hat{p}_{t+1} \in p^*$ *from the pf (for instance, maximizing argument)*
$$^{\lfloor I}f(p_{t+1}|\phi_t) \propto {}^{\lfloor U}f(p_{t+1}|\phi_t) \exp[-\omega_\gamma(p_{t+1}, \phi_t)].$$
6. *Present the projections* $^{\lfloor I}f(d_{\hat{p}_{t+1};t+1}|\phi_t)$ *of the ideal pdf to* $^{\lfloor I}f(d_{t+1}|\phi_t)$ *the user.*
7. *Go to the beginning of* Sequential mode.

Even this algorithm may be infeasible. This makes us to design the presentation strategy that minimizes γ-bound, Proposition 7.1.8. Moreover, the number of compared presentation values is quickly growing with $\mathring{p}$. It equals to $\begin{pmatrix} \mathring{d}_o \\ \mathring{p} \end{pmatrix}$. Thus, it is almost necessary to select $\mathring{p} = 1$ or 2 and to show the user data entries (pairs) with the highest values of $^{\lfloor I}f(p_t|d(t-1))$. This variant is presented in the algorithm below that otherwise assumes the receding-horizon certainty-equivalent strategy and the state description with the state ϕ.

Algorithm 7.3.2 (Design of priorities based on γ-bound on (7.45) & alternative ordering)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state* ϕ_t*, Chapter 6.*
- *Specify the user ideal pdf on* $d_{o;t}, u_{o;t}, c_t, p_t$*.*
- *Select the length of the receding horizon* $T \geq 1$*.*

Sequential (on line) mode, *running for* $t = 1, 2, \ldots$,

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t*.*
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*
3. *Design academic, industrial or simultaneous strategy generating the optimal* $c_t, u_{o;t}$ *and thus determining the ideal pdf* ${}^{\lfloor I}f(d_t|\phi_{t-1}) = \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1})\, {}^{\lfloor I}f(d_t|\phi_{t-1}, c_t)$ *on* d_t*.*
4. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $\gamma(\phi_\tau) = 1$*.*

   Iterative mode

   (a) Evaluate, for each $p_\tau \in p^*$*, the* *<u>functions</u>*

$$
\begin{aligned}
f(c|d_{p_\tau;\tau}, \phi_{\tau-1}) &\equiv \frac{\alpha_c f(d_{p_\tau;\tau}|\phi_{\tau-1}, c)}{\sum_{c \in c^*} \alpha_c f(d_{p_\tau;\tau}|\phi_{\tau-1}, c)} \\
\gamma(\phi_{\tau-1}) &\equiv \exp\left[- \sum_{p_\tau \in p^*} {}^{\lfloor U}f(p_\tau|\phi_{\tau-1}) \omega_\gamma(p_\tau, \phi_{\tau-1}) \right] \\
\omega_\gamma(p_\tau, \phi_{\tau-1}) &\equiv \sum_{c_\tau \in c^*} {}^{\lfloor I}f(c_\tau|\phi_{\tau-1}) \int {}^{\lfloor I}f(d_{p_\tau;\tau}|\phi_{\tau-1}, c_\tau) \times \\
&\times \left[\ln\left(\frac{{}^{\lfloor I}f(d_{p_\tau;\tau}|d_{p+;\tau}, \phi_{\tau-1}, c_\tau)}{{}^{\lfloor U}f(d_{p_\tau;\tau}|\phi_{\tau-1})} \right) + \sum_{c \in c^*} f(c|d_{p_\tau;\tau}, \phi_{\tau-1}) \omega(c, d_{p_\tau;\tau}, \phi_{t-1}) \right] dd_{p_\tau;\tau} \\
\omega(c, d_{p_\tau;\tau}, \phi_{\tau-1}) &\equiv \ln\left(\frac{f(c|d_{p_\tau;\tau}, \phi_{\tau-1})}{\alpha_c} \right) + \\
&+ \int {}^{\lfloor I}f(d_{\bar{p}_\tau;\tau}|d_{p_\tau;\tau}, \phi_{\tau-1}, c) \ln\left(\frac{{}^{\lfloor I}f(d_{\bar{p}_\tau;\tau}|d_{p_\tau;\tau}\phi_{\tau-1}, c)}{\gamma(\phi_\tau)\, {}^{\lfloor U}f(d_{\bar{p}_\tau;\tau}|d_{p_\tau;\tau}, \phi_{\tau-1})} \right) dd_{\bar{p}_\tau;\tau}.
\end{aligned}
$$

   (b) Continue if $\tau = t + 1$*, otherwise decrease* $\tau = \tau - 1$ *and go to the beginning of* Iterative mode.

5. *Order the values* ${}^{\lfloor I}f(p_{t+1}|\phi_t) \propto {}^{\lfloor U}f(p_{t+1}|\phi_t) \exp[-\omega_\gamma(p_{t+1}, \phi_t)]$*.*
6. *Present projections of the ideal pdf* ${}^{\lfloor I}f(d_{\hat{p}_{t+1};t+1}|\phi_t)$ *to the user. The indices* $\hat{p}_{t+1}$ *are chosen so that they have the highest values of* ${}^{\lfloor I}f(p_{t+1}|\phi_t)$*.*

   The dimension of $\hat{p}_{t+1}$ *may be much higher than that of* p_{t+1} *when joining the "top"* p_{t+1}*s into* $\hat{p}_{t+1}$

7. *Go to the beginning of* Sequential mode.

Problem 7.3.1 (Completion of the set of presentation strategies) *A whole bunch of algorithms should be completed for presentation combining IST strategy, Section 4.2.1, grouping mimic (7.31), the maximum probable design, etc.*

Problem 7.3.2 (An alternative way of design of presentation strategies) *Alternatively, it seems possible to reduce data* d_o *to a sub-selection given by presentation action* p *and perform "ordinary", say simultaneous, design. By doing it for several* p*'s, we can compare the achieved minimum and present those variables corresponding to the smallest one. It avoids the cumbersome upper bounds but needs several simultaneous designs performed in parallel. It can be expected that a careful strategy of selecting the compared variants could lead to a feasible algorithm.*

7.3.2 Stimulating the operator

When designing signaling strategy, we assume that the academic, industrial or simultaneous design of the ideal pdf ${}^{\lfloor I}f(d(\mathring{t}))$ has been performed. Then, the model relating the signaling action s_t to the response of the o-system reduces to

$$
\begin{aligned}
{}^{\lfloor I}f(d_t,s_t|d(t-1)) &\equiv {}^{\lfloor I}f(d_t|s_t,d(t-1))\,{}^{\lfloor I}f(s_t|d(t-1)),\ s_t \in s^* \equiv \{0,1\} \text{ with}\\
{}^{\lfloor I}f(d_t|s_t=0,d(t-1)) &\equiv f(d_t|d(t-1)) \equiv \text{ learnt mixture describing the non-guided o-system}\\
{}^{\lfloor I}f(d_t|s_t=1,d(t-1)) &\equiv {}^{\lfloor I}f(d_t|d(t-1)) = \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|d(t-1))\,{}^{\lfloor I}f(d_t|d(t-1),c_t), \quad \text{where}\\
{}^{\lfloor I}f(d_t|d(t-1),c_t) &\equiv f(d_t=\Delta_t|d(t-1),c_t) \text{ for the academic design}\\
{}^{\lfloor I}f(d_t|d(t-1),c_t) &\equiv f(\Delta_t|u_{o;t},d(t-1),c_t)\,{}^{\lfloor I}f(u_{o;t}|d(t-1)) \text{ for the industrial design}\\
{}^{\lfloor I}f(d_t|d(t-1),c_t) &\equiv f(\Delta_t|u_{o;t},d(t-1),c_t)\,{}^{\lfloor I}f(u_{o;t}|d(t-1),c_t) \text{ for the simultaneous design}\\
f(\Delta_t|u_{o;t},d(t-1),c_t) &\equiv \text{the conditional pdf computed from } c_t\text{-th estimated component}\\
{}^{\lfloor I}f(u_{o;t}|d(t-1)) &\equiv \text{the conditional pdf resulting from the industrial design}\\
{}^{\lfloor I}f(u_{o;t}|d(t-1),c_t) &\equiv \text{the conditional pdf resulting from the simultaneous design}\\
{}^{\lfloor I}f(c_t|d(t-1)) &\equiv \alpha_{c_t} \quad \text{for the industrial design}\\
{}^{\lfloor I}f(c_t|d(t-1)) &\equiv \text{the pf resulting from the academic or simultaneous design.}
\end{aligned}
$$

The exact fully probabilistic design of the optimal signaling strategy is hopeless at least because of the mixture form of the used models and the rational form needed when d^*_{p+} is non-empty. This makes us to design directly an approximate strategy. Simplicity of the action space $s^* = \{0,1\}$ allows us to use the more tight bound of the ω-type, Proposition 7.1.9.

Proposition 7.3.3 (Signaling minimizing the Jensen and ω-bound on the KL distance) *Let us consider that the academic, industrial or simultaneous design of the advisory system has provided the strategy $d^*(t-1) \to (c_t^*, u^*_{o;t})$ determining the ideal pdf*

$$
{}^{\lfloor I}f(d_t|d(t-1)) = \sum_{c_t\in c^*} {}^{\lfloor I}f(d_t|d(t-1),c_t)\,{}^{\lfloor I}f(c_t|d(t-1)).
$$

Let us assume that conditions of Proposition 7.1.4 are met.

Then, the following signaling strategy minimizes the Jensen ω-type bound, Proposition 7.1.9, on the KL distance. The recursion run for $t = \mathring{t}, \mathring{t}-1, \ldots, 1$.

$$
\begin{aligned}
f(s_t|d(t-1)) &\propto {}^{\lfloor U}f(s_t|d(t-1))\exp[-\omega_\gamma(s_t,d(t-1))], \ \textit{where} \qquad (7.49)\\
\omega_\gamma(s_t=0,d(t-1)) &\equiv \sum_{c\in c^*} \alpha_c \omega_\gamma(c,d(t-1))\\
\omega_\gamma(c,d(t-1)) &\equiv \int f(d_t|d(t-1),c)\left[\ln\left(\frac{f(d_{o;t}|d_{p+;t},d(t-1),c)}{{}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right)+\omega_\gamma(d(t))\right] dd_t\\
\omega_\gamma(s_t=1,d(t-1)) &\equiv \sum_{c\in c^*} {}^{\lfloor I}f(c_t|d(t-1))\,{}^{\lfloor I}\omega_\gamma(c,d(t-1))\\
{}^{\lfloor I}\omega(c,d(t-1)) &\equiv \int {}^{\lfloor I}f(d_t|d(t-1),c)\left[\ln\left(\frac{{}^{\lfloor I}f(d_{o;t}|d_{p+;t},d(t-1),c)}{{}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right)+\omega_\gamma(d(t))\right] dd_t\\
\omega_\gamma(d(t)) &\equiv {}^{\lfloor U}f(s_{t+1}=0|d(t))\omega_\gamma(s_{t+1}=0,d(t)) + {}^{\lfloor U}f(s_{t+1}=1|d(t))\omega_\gamma(s_{t+1}=1,d(t))\\
\omega_\gamma(d(\mathring{t})) &= 0.
\end{aligned}
$$

Proof: According to Proposition 7.1.4, the optimal strategy minimizing the KL distance

$$
\mathcal{D}\left({}^{\lfloor I}f||\,{}^{\lfloor U}f\right) \equiv \int {}^{\lfloor I}f(d(\mathring{t}),s(\mathring{t}))\ln\left(\frac{{}^{\lfloor I}f(d(\mathring{t}),s(\mathring{t}))}{{}^{\lfloor U}f(d(\mathring{t}),s(\mathring{t}))}\right) d(d(\mathring{t})s(\mathring{t}))
$$

to the user-given ideal pdf ${}^{\lfloor U}f(d(\mathring{t}), s(\mathring{t}))$, fulfilling (7.17), has the form.

$$
\begin{aligned}
{}^{\lfloor I}f(s_t|d(t-1)) &= {}^{\lfloor U}f(s_t|d(t-1))\frac{\exp[-\omega_\gamma(s_t, d(t-1))]}{\gamma(d(t-1))}, \quad \text{where} \\
\gamma(d(t)) &\equiv \sum_{s_t \in s^*} {}^{\lfloor U}f(s_{t+1}|d(t)) \exp[-\omega_\gamma(s_{t+1}, d(t))] \\
\omega_\gamma(s_t = 0, d(t-1)) &\equiv \int f(d_t|d(t-1)) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \\
\omega_\gamma(s_t = 1, d(t-1)) &\equiv \int {}^{\lfloor I}f(d_t|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_{o;t}|d_{p+;t}, d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \\
\gamma(d(\mathring{t})) &= 1.
\end{aligned}
$$

If we find an upper bounds on $\omega_\gamma(d(t-1))$ and insert them into definition of $\gamma(d(t))$ we get a lower bound on it and consequently the upper bound on the Bellman function $-\ln(\gamma(d(t)))$. Let us do that.

We use Proposition 7.1.6 and omit the term $\ln(K)$ that has not influence on optimization. It implies

$$
\begin{aligned}
&\omega_\gamma(s_t = 0, d(t-1)) \equiv \int f(d_t|d(t-1)) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \leq \\
&\leq \sum_{c \in c^*} \alpha_c \int f(d_t|d(t-1), c) \ln\left(\frac{f(d_{o;t}|d_{p+;t}, d(t-1), c)}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \equiv \sum_{c \in c^*} \alpha_c \omega_\gamma(c, d(t-1)) \text{ and} \\
&\omega_\gamma(s_t = 1, d(t-1)) \equiv \int {}^{\lfloor I}f(d_t|d(t-1)) \ln\left(\frac{{}^{\lfloor I}f(d_{o;t}|d_{p+;t}, d(t-1))}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \leq \\
&\leq \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1)) \int {}^{\lfloor I}f(d_t|d(t-1), c_t) \ln\left(\frac{{}^{\lfloor I}f(d_{o;t}|d_{p+;t}, d(t-1), c_t)}{\gamma(d(t))\, {}^{\lfloor U}f(d_{o;t}|d_o(t-1))}\right) dd_t \equiv \\
&\equiv \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|d(t-1))\, {}^{\lfloor I}\omega_\gamma(c_t, d(t-1)).
\end{aligned}
$$

Substituting this upper bound into the definition of $\gamma(d(t))$ and using the inequality between weighted arithmetic and geometric means for further bounding of $\gamma(d(t))$, we complete the last step giving the claimed result $\ln(\gamma(d(t))) \geq$

$$
\geq \sum_{c_t \in c^*} \left[{}^{\lfloor U}f(s_t = 0|d(t-1))\alpha_{c_t}\omega_\gamma(c_t, d(t-1)) + {}^{\lfloor U}f(s_t = 1|d(t-1))\, {}^{\lfloor I}f(c_t|d(t-1))\, {}^{\lfloor I}\omega_\gamma(c_t, d(t-1))\right].
$$

□

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm written for the practically used state description with the state ϕ.

Algorithm 7.3.3 (Signaling minimizing the Jensen and ω-bound on the KL distance)
Initial (off line) mode

- *Estimate the mixture model* $f(d_t|d(t-1)) = \sum_{c \in c^*} \alpha_c f(d_t|\phi_{t-1}, c)$ *of the o-system with the state* ϕ_t*, Chapter 6.*
- *Specify the user ideal pf* ${}^{\lfloor U}f(s_t|d(t-1)) = {}^{\lfloor U}f(s_t|\phi_{t-1})$ *on signaling actions* $s_t \in \{0, 1\}$.
- *Specify the user ideal pdf on the recommended pointers* c_t *and recognizable actions* $u_{o;t}$.
- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update the estimates of the model parameters if you deal with the adaptive advisory system.*

3. *Perform the academic, industrial or simultaneous design generating the ideal pdf*

$$^{\lfloor I}f(d_t|\phi_{t-1}) = \sum_{c_t \in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1})\, {}^{\lfloor I}f(d_t|\phi_{t-1}, c_t).$$

4. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $\omega_\gamma(\phi_\tau) = 0$.

   Iterative mode

   (a) *Evaluate the* *functions*

$$\begin{aligned} {}^{\lfloor I}f(c_\tau, u_{o;\tau}|d(\tau-1)) &= {}^{\lfloor I}f(c_\tau, u_{o;\tau}|\phi_{\tau-1}) \\ &\text{using the academic, industrial or simultaneous design that gives} \\ &\qquad {}^{\lfloor I}(d_\tau|\phi_{\tau-1}) = \sum_{c_\tau \in c^*} {}^{\lfloor I}f(c_\tau|\phi_{\tau-1})\, {}^{\lfloor I}f(d_\tau|\phi_{\tau-1}, c_\tau) \text{ with} \\ {}^{\lfloor I}f(d_\tau|\phi_{\tau-1}, c_\tau) &\equiv f(\Delta_\tau|u_{o;\tau}, \phi_\tau, c_\tau)\, {}^{\lfloor I}f(u_{o;\tau}|\phi_\tau, c_\tau). \end{aligned}$$

   (b) *Determine the following* *functions*

$$\begin{aligned} \omega_\gamma(c, \phi_{\tau-1}) &\equiv \int f(d_\tau|\phi_{\tau-1}, c)\left[\ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau}, \phi_{\tau-1}, c)}{{}^{\lfloor U}f(d_\tau|\phi_{\tau-1})}\right) + \omega_\gamma(\phi_\tau)\right] dd_\tau,\ c \in c^* \\ {}^{\lfloor I}\omega_\gamma(c_\tau, \phi_{\tau-1}) &\equiv \int {}^{\lfloor I}f(d_\tau|\phi_{\tau-1}, c_\tau)\left[\ln\left(\frac{{}^{\lfloor I}f(d_{o;\tau}|d_{p+;\tau}, \phi_{\tau-1}, c_\tau)}{{}^{\lfloor U}f(c_\tau, d_{o;\tau}|\phi_{\tau-1})}\right) + \omega_\gamma(\phi_\tau)\right] dd_\tau \text{ with} \\ \omega(\phi_\tau) &\equiv {}^{\lfloor U}f(s_{\tau+1} = 0|\phi_\tau)\omega_\gamma(s_{\tau+1} = 0, \phi_\tau) + {}^{\lfloor U}f(s_{\tau+1} = 1|\phi_\tau)\omega_\gamma(s_{\tau+1} = 1, \phi_\tau) \\ \omega_\gamma(s_{\tau+1} = 0, \phi_\tau) &\equiv \sum_{c_{\tau+1} \in c^*} \alpha_{c_{\tau+1}} \omega_\gamma(c_{\tau+1}, \phi_\tau) \\ \omega_\gamma(s_{\tau+1} = 1, \phi_\tau) &\equiv \sum_{c_{\tau+1} \in c^*} {}^{\lfloor I}f(c_{\tau+1}|\phi_\tau\, {}^{\lfloor I}\omega_\gamma(c_{\tau+1}, \phi_\tau). \end{aligned}$$

   (c) *Continue if* $\tau = t + 1$. *Otherwise decrease* $\tau = \tau - 1$ *and go to the beginning of* Iterative mode.

5. *Evaluate the optimal signaling strategy* ${}^{\lfloor I}f(s_{t+1}|\phi_t) \propto {}^{\lfloor U}f(s_{t+1}|\phi_t)\exp[-\omega_\gamma(s_{t+1}, \phi_t)$.

6. *Alert the user to make recommended actions if* ${}^{\lfloor I}f(s_{t+1} = 0|\phi_t)$ *is close to zero. Otherwise let him proceed as usual.*

7. *Go to the beginning of* Sequential mode.

Problem 7.3.3 (Order of presentation and signaling designs) *The design of the signaling strategy has to be done after finishing the academic, industrial or simultaneous design. It is, however, unclear whether to perform it after specification of priorities or whether priorities should be decided after the decision that the operator should make some actions.*

The former version is more realistic as it measures possible improvements limited by the restricted ability of human being to grasp information. The latter case is less conservative and more sensitive to possible problems that need not be manageable by the operator. The higher sensitivity and simplicity of the design make us to select the latter case. The alternative solution has to be inspected in future.

Problem 7.3.4 (Approximation of Bellman functions) *Evaluation of Bellman functions in all design variants has to be avoided in practical applications. It seems that there are some common approaches that could be used. This hypothesis is extrapolated from the results presented here as well as those of Chapter 9. It should be inspected in detail.*

7.4 Design validation

The full scale use of the advisory system is the only decisive validation test. Naturally, a collection of indicators is needed in the design phase for checking the chance of the final success and for an early recognition of troubles. The following, definitely incomplete, list of ideas has been collected and partially elaborated up to now.

- The model validation has to be successfully passed, see Section 6.7.
- The estimated model has to be reliable as a long horizon predictor, see Section 7.1.2. If need be, the model estimate has to be stabilized as described in Algorithm 7.1.1 and tested for quality of predictions.
- The KL distance of the estimated model of the o-system to the user ideal pdf should be close to its sample version obtained from data. The construction of such a sampled version is generally non-trivial task for continuous valued data. A plausible solution is, however, available for the normal case that is of our primary interest, see Section 9.1.4.

  The analysis should be done both for components and the overall mixture, see point 8 of Remarks 9.2.4.
- The advisory system should behave properly when an artificial closed loop is created with the estimated model in the role of the o-system. This test can be implemented only when recognizable actions are present, when the industrial or simultaneous design is tested.
- Open-loop advices, generated by the p-system but not feeded into the o-system, should be reasonably close to the good practice reflected in learning data. It has an important psychological aspect with respect to users: they surely will not be ready to change radically their current habits.
- All tests above should be robust with respect to small changes of tuning knobs used both in learning and design.
- • •

Problem 7.4.1 (Systematic design validation) *Design validation is even harder that the learning validation. Thus, it is not surprising that it still waits for a systematic solution.*

Chapter 8

Learning with normal factors and components

Factors form basic building blocks of components. Feasibility of evaluations outlined in Chapter 6 singles out *normal factors*, *Markov factors*, Chapter 10, and so called *MT factors*, Chapter 12, as suitable candidates. Chapter elaborates the operations needed for normal factors in detail. Layout of Chapter 6 is more or less copied. Operations outlined there are specialized to normal factors, to Gauss-inverse-Wishart pdf (GiW) that is their conjugate pdf and to numerically stable learning algorithms based on $L'DL$ or LDL' decompositions. Aspects related to *normal components* consisting solely of normal factors are also discussed.

The normal parameterized factor, we deal with, predicts a real-valued variable $d_t \equiv d_{i;t},\ i \in \{1,\ldots,\mathring{d}\}$ by the pdf

$$f(d_t|d_{i+1\_\mathring{d};t}, d(t-1), \Theta) = \mathcal{N}_{d_t}(\theta'\psi_t, r),\ \text{where} \tag{8.1}$$

$\Theta \ \equiv [\theta, r] \equiv$ [regression coefficients, noise variance] $\in \Theta^* \subset$ ($\mathring{\psi}$-dimensional, non-negative) real variables,

ψ is the regression vector,

$$\mathcal{N}_d(\theta'\psi, r) \equiv (2\pi r)^{-0.5} \exp\left\{-\frac{(d-\theta'\psi)^2}{2r}\right\} = (2\pi r)^{-0.5} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(\Psi\Psi'[-1,\theta']'[-1,\theta']\right)\right\}, \tag{8.2}$$

$\Psi \equiv [d, \psi']' \equiv$ data vector, $'$ denotes transposition.

8.1 Common tools

Section collects common technical tools used for learning with mixtures consisting of normal factors.

8.1.1 Selected matrix operations

We use repeatedly the following matrix operations.

Proposition 8.1.1 (Some matrix formulas) *Let the involved matrices A, B, C have compatible dimensions and the used inversions exist. Then,*

$$\begin{aligned}
\mathrm{tr}(ABC) &= \mathrm{tr}(CAB) = \mathrm{tr}(BCA) && (8.3)\\
\mathrm{tr}(A) &= \mathrm{tr}(A'),\ |A| = |A'|,\ |A^{-1}| = |A|^{-1} && (8.4)\\
(A+BCD)^{-1} &= A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} && (8.5)\\
\frac{\partial}{\partial A}\mathrm{tr}(AB) &= B',\ \frac{\partial}{\partial A}\ln(|A|) = A^{-1} && (8.6)
\end{aligned}$$

Proof:
ad (8.3): $\mathrm{tr}(ABC) \equiv \sum_{i\in i^*}(ABC)_{ii} = \sum_{i\in i^*}\sum_{k\in k^*}(AB)_{ik}C_{ki} = \sum_{k\in k^*}\left(\sum_{i\in i^*} C_{ki}(AB)_{ik}\right) = \sum_{k\in k^*}(CAB)_{kk} \equiv \mathrm{tr}(CAB)$.
ad (8.4): $\mathrm{tr}(A) \equiv \sum_{i\in i^*} A_{ii} = \sum_{i\in i^*}(A')_{ii} \equiv \mathrm{tr}(A')$.

Let λ be eigenvalue of A, i.e. there is a vector $x \neq 0$ such that $Ax = \lambda x$. It implies that x' solves the equation $x'A' = \lambda x'$. Thus, eigenvalues of A and A' coincide and consequently $|A| = |A'|$. Similarly, for non-zero eigenvalues $\lambda(A)$ belonging to the regular matrix A, we get $\lambda(A^{-1}) = 1/\lambda(A)$. Thus, $|A^{-1}| = 1/|A|$.
ad (8.5):

$$\begin{aligned} & (A+BCD)\left[A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}\right] = \\ = \; & I + BCDA^{-1} - B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} - BCDA^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} = \\ = \; & I + BCDA^{-1} - B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} - BCDA^{-1} + B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} = I. \end{aligned}$$

ad (8.6): $\frac{\partial}{\partial A_{ij}}\mathrm{tr}(AB) = \frac{\partial}{\partial A_{ij}}\left[\sum_{i\in i^*, j\in j^*} A_{ij}B_{ji}\right] = B_{ji} = (B')_{ij}$.

Let A^{kl} be the complement to the entry A_{kl}, i.e. determinant of the matrix obtained from A by cancelling the k-th row and l-th column multiplied by $(-1)^{k+l}$. Then, $|A| = \sum_{kl} A_{kl}A^{kl}$ and

$$\frac{\partial}{\partial A_{ij}} \ln |A| = \frac{A^{ij}}{|A|} = \left(A^{-1}\right)_{ij}.$$

□

8.1.2 $L'DL$ decomposition

A positive definite matrix V, called *extended information matrix*, forms the decisive part of the sufficient statistic in estimation of normal factors, see Section 8.1.5. Often, it is poorly conditioned and the use of its *$L'DL$ decomposition* is the only safe way how to counteract the induced numerical troubles. We collect the facts explicitly used in our text. The book [79] gives a broader view on the problem.

Agreement 8.1.1 ($L'DL$ decomposition) *Let V be a positive definite symmetric matrix with $\mathring{\Psi}$ rows. Let us suppose that $V = L'DL$, where L is a* lower triangular matrix *with a unit diagonal, D is a* diagonal matrix *with positive diagonal entries. This expression is called $L'DL$* decomposition *of V.*

Proposition 8.1.2 (Existence and uniqueness of the $L'DL$ decomposition) *Let V be a positive definite symmetric matrix with $\mathring{\Psi}$ rows. Then, its $L'DL$ decomposition exists and it is unique. Its construction is described by the following algorithm of the $L'DL$ decomposition*

For $j = \mathring{\Psi}, \ldots, 1$
  Set $L_{jj} = 1$
 For $i = \mathring{\Psi}, \ldots, j+1$
  Set $s = V_{ij}$
  For $k = i+1, \ldots, \mathring{\Psi}$
   Set $s = s - L_{ki}D_{kk}L_{kj}$
  end of the cycle over k
  Set $L_{ij} = s/D_{ii}$
 end of the cycle over i
  Set $s = V_{jj}$
 For $k = j+1, \ldots, \mathring{\Psi}$
  Set $s = s - L_{kj}^2 D_{kk}$
 end of the cycle over k
  Set $D_{jj} = s$
end of the cycle over j

Proof: The algorithm is directly implied by the desired identity $V = L'DL$ written entry-wise. The algorithm fails if some $D_{jj} \leq 0$. Let us assume that such D_{jj} occurs and denote ${}^{\lfloor j}L$ the part of the matrix L found up to this moment. Let ${}^{\lfloor j}x$ solves the triangular equation ${}^{\lfloor j}x' \, {}^{\lfloor j}L' = [\underbrace{1, 0, \ldots, 0}_{lenght\ j}]$. Then, the quadratic form $x'Vx$ is not positive for $x = \left[0', \, {}^{\lfloor j}x'\right]'$. This contradicts the assumed definiteness of V. □

Remark(s) 8.1.1

It is worth of stressing that the derived algorithm is useful for proving existence and uniqueness. It may fail numerically in evaluating entries D_{ii} when the matrix is poorly conditioned. We do not use it practically.

Updating $L'DL$ decomposition

The meaningful use of the $L'DL$ decomposition depends on our ability to convert the recursion $V = \lambda V + w\Psi\Psi'$ into a direct, numerically robust, recursion for matrices L and D. This recursion, with Ψ equal to the data vector and scalar positive weights λ, w, arises in estimation, see Section 8.1.5. The direct updating is based on the *algorithm dydr* [66] that makes it efficiently.

Proposition 8.1.3 (Algorithm dydr) *The positive semi-definite matrix of the rank 2 can be given the two equivalent forms defining the same kernels of a quadratic form*

$$[a, b] \begin{bmatrix} D_a & 0 \\ 0 & D_b \end{bmatrix} [a, b]' = [c, d] \begin{bmatrix} D_c & 0 \\ 0 & D_d \end{bmatrix} [c, d]'. \tag{8.7}$$

There, D_a, D_b, D_c, D_d are positive scalars and a, b, c, d are column vectors of the same length $\mathring{a} \geq 2$.
Let the following entries of c, d are assumed to be fixed at values

$$c_j \equiv 1, \ d_j \equiv 0, \quad \text{for a chosen } j \in \{1, \ldots, \mathring{a}\}. \tag{8.8}$$

Then, the right-hand side matrix in (8.7) is determined uniquely by the left-hand side through the following algorithm.

Algorithm 8.1.1 (Dyadic reduction: dydr)

1. *Set $D_c = a_j^2 D_a + b_j^2 D_b$.*
2. *Set $x = \frac{a_j D_a}{D_c}$, $y = \frac{b_j D_b}{D_c}$.*
3. *Set $D_d = \frac{D_a D_b}{D_c}$.*
4. *Evaluate remaining entries of c, d as follows $c_i = x a_i + y b_i$, $d_i = -b_j a_i + a_j b_i$ for $i = 1, \ldots, \mathring{a}$. It gives $c_j = 1$, $d_j = 0$.*

Matlab script of dydr *may look as follows*

```
function [Dc,Dd,c,d]=dydr(Da,Db,a,b,j)
%
% dydr makes dyadic reduction of the positive definite
%           quadratic form [a,b]diag(Da,Db)[a,b]' to the
%           quadratic form [c,d]diag(Dc,Dd)[c,d]' so that
%           j-th entry of c equals 1 and j-th entry of d is zero
%
aj=a(j,1);
bj=b(j,1);
Dc=aj^(2)*Da+bj^(2)*Db;
if Dc<machine_precision, 'the matrix is not positive definite', return, end
  x =aj*Da/Dc;
```

```
  y=bj*Db/Dc;
  Dd=Da*Db/Dc;
 c=x*a+y*b;
 d=bj*a+aj*b;
 c(j,1)=1;
 d(j,1)=0;
```

Proof: Let us take for simplicity $j = 1$ and let $T = \begin{bmatrix} x & -b_1 \\ y & a_1 \end{bmatrix}$ be a regular matrix determined by free scalars x, y. It has second column orthogonal to the row vector $[a_1, b_1]$. Thus, $[c, d] \equiv [a, b]T$ has $d_1 = 0$. The remaining requirement in (8.8) is fulfilled if $a_1x + b_1y = 1$. It also makes determinant of T equal to 1. The inspected decomposed form of the positive semi-definite matrix (8.7) is preserved after such a transformation of $[a, b]$ iff

$$\begin{bmatrix} D_c & 0 \\ 0 & D_d \end{bmatrix} = T^{-1} \begin{bmatrix} D_a & 0 \\ 0 & D_b \end{bmatrix} (T^{-1})',\ T^{-1} = \begin{bmatrix} a_1 & b_1 \\ -y & x \end{bmatrix}.$$

The diagonal form is preserved iff $-ya_1D_a + xb_1D_b = 0$. This together with the former condition $a_1x + b_1y = 1$ determines uniquely y, x and consequently all other elements

$$D_c = a_1^2 D_a + b_1^2 D_b,\ x = \frac{a_1 D_a}{D_c},\ y = \frac{b_1 D_b}{D_c},\ D_d = \frac{D_a D_b}{D_c},\ c = xa + yb,\ d = -b_1 a + a_1 b.$$

$\square$

With the algorithm dydr, the required rank-one updating $V = \lambda V + w\Psi\Psi'$ of $L'DL$ decomposition of the extended information matrix V is straightforward.

Algorithm 8.1.2 (Rank-one updating of $L'DL = \lambda L'DL + w\Psi\Psi'$)

$$
\begin{array}{ll}
 & \textit{Set } b = \Psi,\ D_d = w \\
\textit{For} & j = \mathring{\Psi}, \ldots, 1 \\
 & [D_j, D_d, j\textit{-th column of } L', b] = \mathsf{dydr}\,(\lambda * D_j, D_d, j\textit{-th column of } L', b, j) \\
\textit{end} & \textit{of the cycle over } j
\end{array}
$$

Conversion of $L'DL$ to LDL'

In learning, we deal mostly with the $L'DL$ decomposition, Agreement 8.1.1. For some ready structure estimation algorithms [36], it is useful to deal with the LDL' decomposition. It is given, again uniquely, by the lower triangular matrix L with unit diagonal and by the diagonal matrix D with positive diagonal entries. The algorithm dydr serves well for the needed conversion $L'DL \rightarrow LDL'$, too.

Algorithm 8.1.3 (Conversion of $L'DL$ to LDL')

$$
\begin{array}{l}
\textit{For} \quad j = \mathring{\Psi} - 1, \ldots, 1 \\
\quad \textit{For} \quad i = j + 1, \ldots, \mathring{\Psi} \\
\qquad [D_j, D_i, i\textit{-th column of } L, j\textit{-th column of } L] = \mathsf{dydr}\,(D_j, D_i, i\textit{-th column of } L, j\textit{-th column of } L, j) \\
\quad \textit{end} \quad \textit{of the cycle over } i \\
\textit{end} \quad \textit{of the cycle over } j
\end{array}
$$

Permutation of variables in quadratic forms

We permute entries of regression vectors ψ when evaluating marginal pdfs and estimating the structure of normal factors, see Section 8.6. This action induces permutations of regression coefficients θ. The decisive quadratic form $[-1, \theta']V[-1, \theta']'$ remains unchanged if the corresponding rows and columns of

V are permuted, too. It spoils, however, LDL' or $L'DL$ decompositions of V. The following propositions describe simple algorithms that recover these decompositions after permuting adjacent entries of ψ and thus of θ. More complex permutations are obtained through a sequence of such permutations.

Proposition 8.1.4 (Permutation of adjacent entries in LDL' decomposition) *Let $V = LDL'$ be the decomposition of the extended information matrix corresponding to the data $\Psi = [d, \psi']'$ and parameter $[-1, \theta']'$ vectors. Let*

$$\psi = [h', a, b, c']',\ \tilde{\psi} = [h', \underbrace{b, a}_{\text{permuted}}, c']' \text{ and } \theta = [\theta_h', \theta_a, \theta_b, \theta_c']',\ \tilde{\theta} = [\theta_h', \underbrace{\theta_b, \theta_a}_{\text{permuted}}, \theta_c']' \tag{8.9}$$

with scalars a, b, θ_a, θ_b.

Then, the LDL' decomposition of $\tilde{V} = \tilde{L}\tilde{D}\tilde{L}'$ preserving the quadratic form $[-1, \theta']V[-1, \theta']' = [-1, \tilde{\theta}']\tilde{V}[-1, \tilde{\theta}']'$ is obtained from the decomposition $V = LDL'$ by the following algorithm.

1. *Permute rows of L corresponding to regressors a and b up to the main sub-diagonal entry of the shorter one.*
2. *Store $D_{ao} = D_a =$ entry of D corresponding to the regressor a.*
3. *Re-compute $D_a = D_b + \omega^2 D_a$, where ω is the scalar being on the sub-diagonal of the longer vector of permuted rows.*
4. *Compute auxiliary quantities $x = \omega D_{ao}/D_a$, $y = D_b/D_a$.*
5. *Re-compute $D_b = D_{ao}D_b/D_a =$ entry of D corresponding to the regressor b.*
6. *Store a-th column L_a of L into L_{ao} starting from the entry below ω.*
7. *Re-compute $L_a = xL_a + yL_b$, where L_b contains entries in the adjacent entries of the column b.*
8. *Re-compute $L_b = L_{ao} - \omega L_b$.*
9. *Correct $\omega = x$.*

Proof: The permutation of entries a, b in the regression vector implies the permutation of the corresponding pair of columns and rows in the symmetric matrix V. In the equivalent LDL' decomposition, it changes

$$D = \begin{bmatrix} D_h & & & \\ & D_a & & \\ & & D_b & \\ & & & D_c \end{bmatrix},\ L = \begin{bmatrix} L_h & 0 & 0 & 0 \\ L_{ha} & 1 & 0 & 0 \\ L_{hb} & \omega & 1 & 0 \\ L_{hc} & L_a & L_b & L_c \end{bmatrix}, \text{ with scalars } D_a,\ D_b,\ \omega, \quad \text{to}$$

$$\underline{D} = D,\ \underline{L} = \begin{bmatrix} L_h & 0 & 0 & 0 \\ L_{hb} & \omega & 1 & 0 \\ L_{ha} & 1 & 0 & 0 \\ L_{hc} & L_a & L_b & L_c \end{bmatrix}. \tag{8.10}$$

We search for $\tilde{D}$, $\tilde{L}$ such that quadratic forms $[-1, \tilde{\theta}']\tilde{L}\tilde{D}\tilde{L}'[-1, \tilde{\theta}']'$ and $[-1, \theta']'LDL'[-1, \theta']'$ are equal. It is achieved when $\underline{D}$ and $\underline{L}$ coincide with $\tilde{D}$ and $\tilde{L}_h$ with exception of columns having indices $_a$, $_b$. For them, it must hold

$$\begin{bmatrix} \omega & 1 \\ 1 & 0 \\ L_a & L_b \end{bmatrix} \begin{bmatrix} D_a & \\ & D_b \end{bmatrix} \begin{bmatrix} \omega & 1 \\ 1 & 0 \\ L_a & L_b \end{bmatrix}' = \begin{bmatrix} 1 & 0 \\ \tilde{\omega} & 1 \\ \tilde{L}_a & \tilde{L}_b \end{bmatrix} \begin{bmatrix} \tilde{D}_a & \\ & \tilde{D}_b \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tilde{\omega} & 1 \\ \tilde{L}_a & \tilde{L}_b \end{bmatrix}'. \tag{8.11}$$

We proceed similarly as in derivation of dydr but we exploit the special form of entries a_j, b_j. Let us consider the regular (2,2)-matrices $T = \begin{bmatrix} x & 1 \\ y & -\omega \end{bmatrix}$ parameterized by scalars x, y. Their second column is orthogonal to the vector occurring in the first row of the first matrix in (8.11). It holds

$$\begin{bmatrix} \omega & 1 \\ 1 & 0 \\ L_a & L_b \end{bmatrix} T = \begin{bmatrix} x\omega + y & 0 \\ x & 1 \\ xL_a + yL_b & L_a - \omega L_b \end{bmatrix}.$$

The desired "tilde" form is obtained if

$$\tilde{L}_a = xL_a + yL_b,\ \tilde{L}_b = L_a - \omega L_b,\ x\omega + y = 1,\ T^{-1}\mathrm{diag}[D_a, D_b]\left(T^{-1}\right)' = \text{ diagonal matrix.}$$

The unspecified values x, y has to be chosen so that the last two restrictions are fulfilled. It gives

$$\tilde{D}_a = D_b + \omega^2 D_a,\ x = \frac{\omega D_a}{\tilde{D}_a},\ y = \frac{D_b}{\tilde{D}_a},\ \tilde{D}_b = \frac{D_a D_b}{\tilde{D}_a}.$$

□

We also need to recover $L'DL$ decomposition after permuting adjacent entries. For the sake of completeness, we summarize it.

Proposition 8.1.5 (Permutation of adjacent entries in $L'DL$ decomposition) *Let $V = L'DL$ be the $L'DL$ decomposition of the information matrix V corresponding to the data $\Psi = [d, \psi']'$ and parameter $[-1, \theta']'$ vectors. Let*

$$\psi = [h', a, b, c']',\ \tilde{\psi} = [h', b, a, c']' \text{ and } \theta = [\theta_h', \theta_a, \theta_b, \theta_c']',\ \tilde{\theta} = [\theta_h', \theta_b, \theta_a, \theta_c']' \tag{8.12}$$

with scalars a, b, θ_a, θ_b.

Then the $L'DL$ decomposition of $\tilde{V} = \tilde{L}'\tilde{D}\tilde{L}$ describing the same quadratic form $[-1, \theta']V[-1, \theta']' = [-1, \tilde{\theta}']\tilde{V}[-1, \tilde{\theta}']'$ can be obtained from the decomposition $V = L'DL$ by the following algorithm.

1. *Permute a and b columns of L up to main sub-diagonal entry of the shorter one.*
2. *Store $D_{ao} = D_a =$ entry of D corresponding to the regressor a.*
3. *Re-compute $D_a = D_b + \omega^2 D_a$, where ω is the scalar being on the sub-diagonal of the longer of vector permuted columns.*
4. *Compute auxiliary quantities $x = \omega D_{ao}/D_a,\ y = D_b/D_a$.*
5. *Re-compute $D_b = D_{ao}D_b/D_a$ entry of D corresponding to the regressor b.*
6. *Store a-th row L_a of L into L_{ao} till the entry before ω.*
7. *Re-compute $L_a = xL_a + yL_b$, where L_b contains entries in the adjacent entries of the row b.*
8. *Re-compute $L_b = L_{ao} - \omega L_b$.*
9. *Correct $\omega = x$.*

Proof: The idea is the identical one with the proof of Proposition 8.1.4. Just rows are corrected instead of columns. □

Complete squares and minimization of quadratic forms

We deal with quadratic forms $[-1, \theta']V[-1, \theta]'$ in a $\mathring{\psi}$-vector of regression coefficients θ determined by a positive definite, extended information matrix $V = L'DL$.

Let us split the information matrix V and its $L'DL$ decomposition as follows

$$\begin{aligned} V &= \begin{bmatrix} {}^{\lfloor d}V & {}^{\lfloor d\psi}V' \\ {}^{\lfloor d\psi}V & {}^{\lfloor \psi}V \end{bmatrix},\ {}^{\lfloor d}V \text{ is scalar,} \\ L &= \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L & {}^{\lfloor \psi}L \end{bmatrix},\ D = \begin{bmatrix} {}^{\lfloor d}D & 0 \\ 0 & {}^{\lfloor \psi}D \end{bmatrix},\ {}^{\lfloor d}D \text{ is scalar.} \end{aligned} \tag{8.13}$$

From here onwards, when working with quadratic forms, the *upper left index in* V, D, L indicates blocks obtained by splitting of the matrix according to the element ordering in the corresponding data vector.

Proposition 8.1.6 (Completion of squares) *It holds*

$$\begin{aligned} [-1,\theta']V[-1,\theta']' &\equiv [-1,\theta']L'DL[-1\,\theta']' = (\theta-\hat{\theta})'\,{}^{\lfloor\psi}L'\,{}^{\lfloor\psi}D\,{}^{\lfloor\psi}L(\theta-\hat{\theta}) + {}^{\lfloor d}D \qquad (8.14)\\ \hat{\theta} &\equiv {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L \equiv \text{least-squares estimate } of\,\theta. \end{aligned}$$

This quadratic form is minimized by $\theta = \hat{\theta}$ *and the minimum reached equals to* ${}^{\lfloor d}D \equiv$ least-squares remainder.

Proof: The straightforward vector-matrix multiplication for the split $L'DL$ decomposition proofs equality (8.14). It is known as *completion of squares*. Due to the positive definiteness of V, the term dependent of θ is non-negative and becomes zero for $\theta = \hat{\theta}$. □

8.1.3 GiW pdf as a conjugate prior

Normal factors belong to the exponential family, see Section 3.2, so that they possess conjugate (self-reproducing) prior. The following correspondence to the general form of the exponential family (3.5) holds

$$\begin{aligned} \mathcal{N}_d(\theta'\psi, r) &= A(\Theta)\exp[\langle B(\Psi), C(\Theta)\rangle] \ \text{ with} \qquad (8.15)\\ A(\Theta) &\equiv (2\pi r)^{-0.5},\ B(\Psi) \equiv \Psi\Psi',\ \ C(\Theta) = (2r)^{-1}[-1,\theta']'[-1,\theta'],\ \ \langle B, C\rangle \equiv \mathrm{tr}\,[B'C]. \end{aligned}$$

This correspondence determines the conjugate prior, see (3.12), in the form known as Gauss-inverse-Wishart pdf (GiW)

$$GiW_\Theta(V,\nu) \equiv GiW_{\theta,r}(V,\nu) \equiv \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(V,\nu)}\exp\left\{-\frac{1}{2r}\mathrm{tr}\,(V[-1,\theta']'[-1,\theta'])\right\}. \qquad (8.16)$$

The value of the normalization integral $\mathcal{I}(V,\nu)$ and restrictions on statistics V,ν that guarantee finiteness of $\mathcal{I}(V,\nu)$ are described below, together with other properties of this important pdf.

The $(\mathring{\Psi},\mathring{\Psi})$-dimensional *extended information matrix* V can be chosen symmetric. Its potential anti-symmetric constituents disappear in the quadratic form in which it occurs. Moreover, the extended information matrix must be *positive definite*. Otherwise, there are unbounded combinations of θ entries for which the inspected function does not fall to zero and thus is not integrable.

Using the $L'DL$ decomposition of the extended information matrix, Agreement 8.1.1 and splitting (8.13), we summarize basic properties of the GiW pdf.

Proposition 8.1.7 (Basic properties and moments of the GiW pdf)

1. *$GiW_\Theta(V,\nu)$ has the following alternative expressions* $GiW_\Theta(V,\nu) \equiv$

$$\begin{aligned} &\equiv GiW_\Theta(L,D,\nu) = \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(L,D,\nu)}\exp\left\{-\frac{1}{2r}\left[\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right)'{}^{\lfloor\psi}D\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right) + {}^{\lfloor d}D\right]\right\} \equiv\\ &\equiv \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(L,D,\nu)}\exp\left\{-\frac{1}{2r}\left[(\theta-\hat{\theta})'C^{-1}(\theta-\hat{\theta}) + {}^{\lfloor d}D\right]\right\} \equiv \qquad (8.17)\\ &\equiv \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(L,D,\nu)}\exp\left\{-\frac{1}{2r}\left[Q(\theta) + {}^{\lfloor d}D\right]\right\},\ \ \textit{where}\\ \hat{\theta} &\equiv {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L \equiv \textit{least-squares estimate of}\,\theta \qquad (8.18)\\ C &\equiv {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor\psi}D^{-1}\left({}^{\lfloor\psi}L'\right)^{-1} \equiv \textit{covariance factor of least-squares estimate} \qquad (8.19)\\ {}^{\lfloor d}D &\equiv \textit{least-squares remainder} \qquad (8.20)\\ Q(\theta) &\equiv \left(\theta-\hat{\theta}\right)'C^{-1}\left(\theta-\hat{\theta}\right) \equiv \left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right)'{}^{\lfloor\psi}D\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right). \qquad (8.21) \end{aligned}$$

2. *The normalization integral is*

$$\mathcal{I}(L,D,\nu) = \Gamma(0.5\nu)\,{}^{\lfloor d}D^{-0.5\nu}\left|{}^{\lfloor\psi}D\right|^{-0.5} 2^{0.5\nu}(2\pi)^{0.5\mathring{\psi}} \quad \textit{with} \tag{8.22}$$
$$\Gamma(x) = \int_0^\infty z^{x-1}\exp(-z)\,dz < \infty \quad \textit{for } x > 0.$$

Thus, it is finite iff $\nu > 0$ *and* V *is positive definite* $\Leftrightarrow D > 0$ *(entry-wise).*

3. *The GiW pdf has the following marginal pdfs and moments*

$$f(r|L,D,\nu) = \frac{r^{-0.5(\nu+2)}}{\mathcal{I}\left({}^{\lfloor d}D,\nu\right)}\exp\left[-\frac{{}^{\lfloor d}D}{2r}\right] \equiv iW_r\left({}^{\lfloor d}D,\nu\right) \equiv \text{inverse Wishart pdf} \tag{8.23}$$
$$\mathcal{I}\left({}^{\lfloor d}D,\nu\right) \equiv \frac{\Gamma(0.5\nu)}{\left(0.5\,{}^{\lfloor d}D\right)^{0.5\nu}} \equiv \textit{normalization integral of } f(r|L,D,\nu).$$
$$\mathcal{E}[r|L,D,\nu] = \frac{{}^{\lfloor d}D}{\nu-2} \equiv \hat{r},\quad \mathrm{var}[r|L,D,\nu] = \frac{2\hat{r}^2}{\nu-4},\quad \mathcal{E}[r^{-1}|L,D,\nu] = \frac{\nu}{{}^{\lfloor d}D}$$
$$\mathcal{E}[\ln(r)|L,D,\nu] = \ln\left({}^{\lfloor d}D\right) - \ln(2) - \frac{\partial\ln\left(\Gamma(0.5\nu)\right)}{\partial(0.5\nu)}$$
$$f(\theta|L,D,\nu) = \mathcal{I}^{-1}(D,\nu)\left[1+\left({}^{\lfloor d}D\right)^{-1}\left(\theta-\hat{\theta}\right)'\,{}^{\lfloor\psi}L'\,{}^{\lfloor\psi}D\,{}^{\lfloor\psi}L\left(\theta-\hat{\theta}\right)\right]^{-0.5(\nu+\mathring{\psi})}$$
$$\mathcal{I}(D,\nu) \equiv \frac{{}^{\lfloor d}D^{0.5\mathring{\psi}}\prod_{i=1}^{\mathring{\psi}}\Gamma(0.5(\nu+\mathring{\psi}-i))}{\left|{}^{\lfloor\psi}D\right|^{0.5}} \equiv \textit{normalization integral of } f(\theta|L,D,\nu)$$
$$\mathcal{E}[\theta|L,D,\nu] = {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L \equiv \hat{\theta},\ \mathrm{cov}[\theta|L,D,\nu] = \frac{{}^{\lfloor d}D}{\nu-2}\,{}^{\lfloor\psi}L^{-1}\,{}^{\lfloor\psi}D^{-1}\left({}^{\lfloor\psi}L'\right)^{-1} \equiv \hat{r}C.$$

Proof: Majority of evaluations can be found in [14], [80]. Here we fix the most important steps and evaluate the non-standard quantities $\mathcal{E}[\ln(r)|L,D,\nu]$ and $\mathcal{E}\left[r^{-1}|L,D,\nu\right]$ needed in Proposition 8.1.9.

1. Alternative forms rest on the completion of squares as described in Proposition 8.1.6.
2. Let us put expressions substitutions used in compound brackets within the sequence of evaluations. Then, the normalization integral is evaluated as follows

$$\begin{aligned}
\mathcal{I}(L,D,\nu) &\equiv \int r^{-0.5(\nu+\mathring{\psi}+2)}\exp\left\{-\frac{1}{2r}\left[\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right)'\,{}^{\lfloor\psi}D\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right) + {}^{\lfloor d}D\right]\right\}d\theta dr = \\
&= \left\{x \equiv r^{-0.5}\,{}^{\lfloor\psi}D^{0.5}\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right),\ dx = r^{-0.5\mathring{\psi}}\left|{}^{\lfloor\psi}D\right|^{0.5}d\theta\right\} = \\
&= (2\pi)^{0.5\mathring{\psi}}\left|{}^{\lfloor\psi}D\right|^{-0.5}\int_0^\infty r^{-0.5(\nu+2)}\exp\left[-0.5r^{-1}\,{}^{\lfloor d}D\right]dr = \\
&= \left\{x = 0.5r^{-1}\,{}^{\lfloor d}D,\ r = 0.5\,{}^{\lfloor d}Dx^{-1},\ dr = -0.5\,{}^{\lfloor d}Dx^{-2}dx\right\} = \\
&= (2\pi)^{0.5\mathring{\psi}}\left|{}^{\lfloor\psi}D\right|^{-0.5}\left(0.5\,{}^{\lfloor d}D\right)^{-0.5\nu}\int_0^\infty x^{0.5(\nu-2)}\exp(-x)\,dx \quad \underbrace{=}_{definition\ of\ \Gamma(\cdot)} \\
&= \Gamma(0.5\nu)\,{}^{\lfloor d}D^{-0.5\nu}\left|{}^{\lfloor\psi}D\right|^{-0.5}2^{0.5\nu}(2\pi)^{0.5\mathring{\psi}}.
\end{aligned}$$

3. Integrating out θ in previous step we found the marginal pdf of r proportional to $r^{-0.5(\nu+2)}\exp\left[-\frac{{}^{\lfloor d}D}{2r}\right]$. The corresponding normalization integral is obtained as follows.

$$\begin{aligned}
\mathcal{I}\left({}^{\lfloor d}D,\nu\right) &\equiv \int_0^\infty r^{-0.5(\nu+2)}\exp\left[-0.5r^{-1}\,{}^{\lfloor d}D\right]dr = \\
&= \left\{x = 0.5r^{-1}\,{}^{\lfloor d}D,\ r = 0.5\,{}^{\lfloor d}Dx^{-1},\ dr = -0.5\,{}^{\lfloor d}Dx^{-2}dx\right\} = \left(0.5\,{}^{\lfloor d}D\right)^{-0.5\nu}\Gamma(0.5\nu).
\end{aligned}$$

4. Using the definition

$$\mathcal{I}\left(\lfloor^{d}D,\nu\right)\equiv\frac{\Gamma(0.5\nu)}{\left(0.5\lfloor^{d}D\right)^{0.5\nu}}$$

of the normalizing factor in the marginal pdf of r, see (8.23), we get

(a) $\mathcal{E}\left[r^{i}\right]=\mathcal{I}\left(\lfloor^{d}D,\nu-2i\right)/\mathcal{I}\left(\lfloor^{d}D,\nu\right)$.
For $i=1$, it gives

$$\mathcal{E}[r]=\frac{0.5\lfloor^{d}D}{0.5(\nu-2)}=\frac{\lfloor^{d}D}{\nu-2}.$$

For $i=2$, it gives

$$\mathcal{E}\left[r^{2}\right]=\frac{\left(0.5\lfloor^{d}D\right)^{2}}{0.5^{2}(\nu-2)(\nu-4)}\Rightarrow\mathrm{var}(r)=\mathcal{E}[r^{2}]-(\mathcal{E}[r])^{2}=\frac{\hat{r}^{2}}{\nu-4}[\nu-2-(\nu-4)]=\frac{2\hat{r}^{2}}{\nu-4}.$$

For $i=-1$, it gives

$$\mathcal{E}\left[r^{-1}\right]=\left(0.5\lfloor^{d}D\right)^{-1}(0.5\nu)=\frac{\nu}{\lfloor^{d}D}.$$

(b)

$$\begin{aligned}\mathcal{E}[\ln(r)|L,D,\nu] &\equiv \frac{1}{\mathcal{I}\left(\lfloor^{d}D,\nu\right)}\int_{0}^{\infty}\ln(r)r^{-0.5(\nu+2)}\exp\left(-\frac{\lfloor^{d}D}{2r}\right)dr=\\ &= \frac{-2}{\mathcal{I}\left(\lfloor^{d}D,\nu\right)}\int_{0}^{\infty}\frac{\partial}{\partial\nu}r^{-0.5(\nu+2)}\exp\left(-\frac{\lfloor^{d}D}{2r}\right)dr=\\ &= \frac{-2}{\mathcal{I}\left(\lfloor^{d}D,\nu\right)}\frac{\partial}{\partial\nu}\int_{0}^{\infty}r^{-0.5(\nu+2)}\exp\left(-\frac{\lfloor^{d}D}{2r}\right)dr=-2\frac{\partial}{\partial\nu}\ln\left(\mathcal{I}\left(\lfloor^{d}D,\nu\right)\right)=\\ &= \ln\left(\lfloor^{d}D\right)-\ln(2)-\frac{\partial\ln\left(\Gamma(0.5\nu)\right)}{\partial(0.5\nu)}.\end{aligned}$$

5. Marginal pdf of θ and corresponding moments are obtained as follows.

$$\begin{aligned}f(\theta) &\propto \int r^{-0.5(\nu+\mathring{\psi}+2)}\exp\left\{-0.5r^{-1}\lfloor^{d}D[1+(\theta-\hat{\theta})'\lfloor^{d}D^{-1}\lfloor^{\psi}L'\lfloor^{\psi}D\lfloor^{\psi}L(\theta-\hat{\theta})]\right\}dr=\\ &= \left\{r=0.5\lfloor^{d}D\left[1+\left(\theta-\hat{\theta}\right)'\lfloor^{d}D^{-1}\lfloor^{\psi}L'\lfloor^{\psi}D\lfloor^{\psi}L\left(\theta-\hat{\theta}\right)\right]/x\right\}=\\ &\propto \left[1+\left(\theta-\hat{\theta}\right)'\lfloor^{d}D^{-1}\lfloor^{\psi}L'\lfloor^{\psi}D\lfloor^{\psi}L\left(\theta-\hat{\theta}\right)\right]^{-0.5(\nu+\mathring{\psi})}.\end{aligned}$$

Symmetry with respect to $\hat{\theta}$ implies that $\mathcal{E}[\theta]=\hat{\theta}$. Moreover, neither normalization integral nor covariance depends on $\hat{\theta}$, thus we can write without loss of generality the further formulas for the special case $\hat{\theta}=0$. Substitution with the unit Jacobian $x=\lfloor^{\psi}L\theta$ implies that the normalizing integral does not depend on $\lfloor^{\psi}L$ and

$$\begin{aligned}\mathcal{I}(D,\nu) &= \int\left[1+x'\lfloor^{d}D^{-1}\lfloor^{\psi}Dx\right]^{-0.5(\nu+\mathring{\psi})-2}dx=\left\{y=\left[\lfloor^{d}D^{-1}\lfloor^{\psi}D\right]^{0.5}x\right\}=\\ &= \lfloor^{d}D^{0.5\mathring{\psi}}\lfloor^{\psi}D^{-0.5}\int(1+y'y)^{-0.5(\nu+\mathring{\psi}-2)}dy\underbrace{=}_{see\ [81]}\lfloor^{d}D^{0.5\mathring{\psi}}\lfloor^{\psi}D^{-0.5}\prod_{i=1}^{\mathring{\psi}}\Gamma(0.5(\nu+\mathring{\psi}-i)).\end{aligned}$$

Using the fact that $\mathcal{E}[\theta|r]=\mathcal{E}[\theta]$, the identity $\mathrm{cov}[\theta|r]=\mathcal{E}[\theta\theta'|r]-\mathcal{E}[\theta]\mathcal{E}[\theta]'$ and the chain rule for expectations, Proposition 2.3.6, we have

$$\begin{aligned}\mathrm{cov}[\theta] &= \mathcal{E}[\theta\theta']-\mathcal{E}[\theta]\mathcal{E}[\theta]'=\mathcal{E}\left[\mathcal{E}[\theta\theta'-\mathcal{E}[\theta]\mathcal{E}[\theta]'|r]\right]=\mathcal{E}[\mathrm{cov}[\theta|r]]=\\ &= \mathcal{E}\left[r\left(\lfloor^{\psi}L'\lfloor^{\psi}D\lfloor^{\psi}L\right)^{-1}\right]=\frac{\lfloor^{d}D}{\nu-2}\left(\lfloor^{\psi}L'\lfloor^{\psi}D\lfloor^{\psi}L\right)^{-1}.\end{aligned}$$

□

We need also to evaluate marginal and conditional pdfs regression coefficients described by GiW pdf.

Proposition 8.1.8 (Low-dimensional pdfs related to GiW pdf) *Let the $L'DL$ decomposition of the extended information matrix determining the pdf $GiW_{\left[{}^{\lfloor a}\theta',\,{}^{\lfloor b}\theta'\right]',r}(L,D,\nu)$ be split in accordance with the splitting of regression coefficients $\theta = \left[{}^{\lfloor a}\theta',\ {}^{\lfloor b}\theta'\right]'$*

$$L \equiv \begin{bmatrix} 1 & & \\ {}^{\lfloor da}L & {}^{\lfloor a}L & \\ {}^{\lfloor db}L & {}^{\lfloor ab}L & {}^{\lfloor b}L \end{bmatrix},\quad D \equiv \begin{bmatrix} {}^{\lfloor d}D & & \\ & {}^{\lfloor a}D & \\ & & {}^{\lfloor b}D \end{bmatrix}.$$

Then,

$$\begin{aligned} f\left({}^{\lfloor a}\theta, r\right) &= GiW_{{}^{\lfloor a}\theta,r}\left(\begin{bmatrix} 1 & \\ {}^{\lfloor da}L & {}^{\lfloor a}L \end{bmatrix}, \begin{bmatrix} {}^{\lfloor d}D & \\ & {}^{\lfloor a}D \end{bmatrix}, \nu\right) \\ f\left({}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta, r\right) &= \mathcal{N}_{{}^{\lfloor b}\theta}\left({}^{\lfloor b}L^{-1}\left[{}^{\lfloor db}L - {}^{\lfloor ab}L\,{}^{\lfloor a}\theta\right], r\left({}^{\lfloor b}L'\,{}^{\lfloor b}D\,{}^{\lfloor b}L\right)^{-1}\right). \end{aligned} \tag{8.24}$$

Proof:

$$GiW_{\left[{}^{\lfloor a}\theta',\,{}^{\lfloor b}\theta'\right]',r}(L,D,\nu) \propto r^{-0.5(\nu+\mathring{\psi}+2)}$$

$$\exp\left\{-\frac{1}{2r}\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \\ {}^{\lfloor b}\theta \end{bmatrix}'\begin{bmatrix} 1 & & \\ {}^{\lfloor da}L & {}^{\lfloor a}L & \\ {}^{\lfloor db}L & {}^{\lfloor ab}L & {}^{\lfloor b}L \end{bmatrix}'\begin{bmatrix} {}^{\lfloor d}D & & \\ & {}^{\lfloor a}D & \\ & & {}^{\lfloor b}D \end{bmatrix}\begin{bmatrix} 1 & & \\ {}^{\lfloor da}L & {}^{\lfloor a}L & \\ {}^{\lfloor db}L & {}^{\lfloor ab}L & {}^{\lfloor b}L \end{bmatrix}\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \\ {}^{\lfloor b}\theta \end{bmatrix}\right\} =$$

$$= \underbrace{r^{-0.5(\nu+{}^{\lfloor a}\mathring{\psi}+2)}\exp\left\{-\frac{1}{2r}\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \end{bmatrix}'\begin{bmatrix} 1 & \\ {}^{\lfloor da}L & {}^{\lfloor a}L \end{bmatrix}'\begin{bmatrix} {}^{\lfloor d}D & \\ & {}^{\lfloor a}D \end{bmatrix}\begin{bmatrix} 1 & \\ {}^{\lfloor da}L & {}^{\lfloor a}L \end{bmatrix}\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \end{bmatrix}\right\}}_{A} \times$$

$$\times \underbrace{r^{-0.5\,{}^{\lfloor b}\mathring{\psi}}\exp\left\{-\frac{1}{2r}\underbrace{\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \\ {}^{\lfloor b}\theta \end{bmatrix}'\left[\ {}^{\lfloor db}L \quad {}^{\lfloor ab}L \quad {}^{\lfloor b}L\ \right]'\,{}^{\lfloor b}D\left[\ {}^{\lfloor db}L \quad {}^{\lfloor ab}L \quad {}^{\lfloor b}L\ \right]\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \\ {}^{\lfloor b}\theta \end{bmatrix}}_{Q}\right\}}_{B}.$$

Integration over ${}^{\lfloor b}\theta$, with the substitution

$$x = r^{-0.5}\left[\ {}^{\lfloor db}L \quad {}^{\lfloor ab}L \quad {}^{\lfloor b}L\ \right]\begin{bmatrix} -1 \\ {}^{\lfloor a}\theta \\ {}^{\lfloor b}\theta \end{bmatrix},$$

concerns only the factor B. The integration result is independent both of r and ${}^{\lfloor a}\theta$. Thus, the factor A, that has the GiW form, is the marginal pdf of $\left({}^{\lfloor a}\theta, r\right)$ searched for.

Consequently, the factor B has to be pdf $f\left({}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta, r\right)$. It is normal pdf. Its moments are easily found by completion squares in the quadratic form Q in the exponent

$$\begin{aligned} Q &= \left({}^{\lfloor b}\theta - \mathcal{E}\left[{}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta, r\right]\right)'\,{}^{\lfloor b}L'\,{}^{\lfloor b}D\,{}^{\lfloor b}L\left({}^{\lfloor b}\theta - \mathcal{E}\left[{}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta, r\right]\right) \text{ with} \\ \mathcal{E}\left[{}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta, r\right] &\equiv \mathcal{E}\left[{}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta\right] = {}^{\lfloor b}L^{-1}\left[{}^{\lfloor db}L - {}^{\lfloor ab}L\,{}^{\lfloor a}\theta\right]. \end{aligned}$$

The covariance matrix is $\mathrm{cov}\left[{}^{\lfloor b}\theta|\,{}^{\lfloor a}\theta, r\right] = \mathrm{cov}\left[{}^{\lfloor b}\theta|r\right] = r\left({}^{\lfloor b}L'\,{}^{\lfloor b}D\,{}^{\lfloor b}L\right)^{-1}$. □

Note that the result of Proposition is simple due to the chosen variant of $L'DL$ decomposition. The alternative version LDL' is more complex.

Repeatedly, we need to evaluate the KL distance of a pair of GiW pdfs.

Proposition 8.1.9 (KL distance of GiW pdfs) *Let $f(\Theta) = GiW_{\Theta}(L, D, \nu)$, $\tilde{f}(\Theta) = GiW_{\Theta}\left(\tilde{L}, \tilde{D}, \tilde{\nu}\right)$ be a pair of GiW pdfs of parameters $\Theta \equiv (\theta, r) =$ (regression coefficients, noise variance). Let D_{ii} stand for the diagonal element of the matrix D. Then, the KL distance of f and $\tilde{f}$ is given by the formula*

$$
\begin{aligned}
\mathcal{D}\left(f||\tilde{f}\right) &= \ln\left(\frac{\Gamma(0.5\tilde{\nu})}{\Gamma(0.5\nu)}\right) - 0.5\ln\left(\prod_{i=2}^{\mathring{\Psi}} \frac{\tilde{D}_{ii}}{D_{ii}}\right) + 0.5\tilde{\nu}\ln\left(\frac{{}^{\lfloor d}D}{{}^{\lfloor d}\tilde{D}}\right) + \qquad (8.25)\\
&+ 0.5(\nu-\tilde{\nu})\frac{\partial}{\partial(0.5\nu)}\ln\left(\Gamma(0.5\nu)\right) - 0.5\mathring{\psi} + 0.5\mathrm{tr}\left[{}^{\lfloor\psi}L^{-1}\left({}^{\lfloor\psi}D\right)^{-1}\left({}^{\lfloor\psi}L'\right)^{-1}{}^{\lfloor\psi}\tilde{L}'\,{}^{\lfloor\psi}\tilde{D}\,{}^{\lfloor\psi}\tilde{L}\right] +\\
&+ 0.5\frac{\nu}{{}^{\lfloor d}D}\left[\left(\hat{\theta}-\hat{\tilde{\theta}}\right)'{}^{\lfloor\psi}\tilde{L}'\,{}^{\lfloor\psi}\tilde{D}\,{}^{\lfloor\psi}\tilde{L}\left(\hat{\theta}-\hat{\tilde{\theta}}\right) - {}^{\lfloor d}D + {}^{\lfloor d}\tilde{D}\right] \equiv\\
&\equiv \ln\left(\frac{\Gamma\left(0.5\tilde{\nu}\right)}{\Gamma\left(0.5\nu\right)}\right) - 0.5\ln\left|C\tilde{C}^{-1}\right| + 0.5\tilde{\nu}\ln\left(\frac{{}^{\lfloor d}D}{{}^{\lfloor d}\tilde{D}}\right) +\\
&+ 0.5(\nu-\tilde{\nu})\frac{\partial}{\partial(0.5\nu)}\ln\left(\Gamma(0.5\nu)\right) - 0.5\mathring{\psi} - 0.5\nu + 0.5\mathrm{tr}\left[C\tilde{C}^{-1}\right] +\\
&+ 0.5\frac{\nu}{{}^{\lfloor d}D}\left[\left(\hat{\theta}-\hat{\tilde{\theta}}\right)'\tilde{C}^{-1}\left(\hat{\theta}-\hat{\tilde{\theta}}\right) + {}^{\lfloor d}\tilde{D}\right].
\end{aligned}
$$

Proof: A substantial part of evaluations is prepared by Proposition 8.1.7. The definition of the KL distance (2.26) reads

$$
\mathcal{D} \equiv \mathcal{D}\left(f||\tilde{f}\right) = \int f(\Theta|L, D, \nu)\ln\left(\frac{f(\Theta|L, D, \nu)}{\tilde{f}\left(\Theta|\tilde{L}, \tilde{D}, \tilde{\nu}\right)}\right) d\Theta.
$$

We write down the GiW pdf in a bit simplified form, cf. (8.19), (8.21),

$$
\begin{aligned}
f(\Theta|L, D, \nu) &\equiv GiW_{\Theta}(L, D, \nu) = \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(L, D, \nu)}\exp\left\{-\frac{1}{2r}\left[Q(\theta) + {}^{\lfloor d}D\right]\right\} \text{ with}\\
Q(\theta) &= \left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right)'{}^{\lfloor\psi}D\left({}^{\lfloor\psi}L\theta - {}^{\lfloor d\psi}L\right) =\\
&= \left(\theta - {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L\right)'{}^{\lfloor\psi}L'\,{}^{\lfloor\psi}D\,{}^{\lfloor\psi}L\left(\theta - {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L\right)\\
&= \left(\theta-\hat{\theta}\right)'C^{-1}\left(\theta-\hat{\theta}\right), \text{ where}
\end{aligned}
$$

$\hat{\theta} \equiv {}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L$ and $C^{-1} = {}^{\lfloor\psi}L'\,{}^{\lfloor\psi}D\,{}^{\lfloor\psi}L$. The same notation is used for $\tilde{f}\left(\Theta|\tilde{L}, \tilde{D}, \tilde{\nu}\right)$. If we recall that $\Theta = [\theta, r]$, we can write

$$
\begin{aligned}
\mathcal{D} &= \ln\left(\frac{\mathcal{I}\left(\tilde{L}, \tilde{D}, \tilde{\nu}\right)}{\mathcal{I}\left(L, D, \nu\right)}\right) + \mathcal{E}\left[-0.5(\nu-\tilde{\nu})\ln(r) - \frac{1}{2r}\left(Q(\theta) + {}^{\lfloor d}D - \tilde{Q}(\theta) - {}^{\lfloor d}\tilde{D}\right)|L, D, \nu\right] =\\
&= \ln\left(\frac{\mathcal{I}\left(\tilde{L}, \tilde{D}, \tilde{\nu}\right)}{\mathcal{I}(L, D, \nu)}\right) - 0.5(\nu-\tilde{\nu})\left[\ln\left({}^{\lfloor d}D\right) - \ln(2) - \frac{\partial\ln\left(\Gamma(0.5\nu)\right)}{\partial(0.5\nu)}\right] - \qquad (8.26)\\
&- \underbrace{0.5\,\mathcal{E}\left\{r^{-1}\left[Q(\theta) - \tilde{Q}(\theta)\right]|L, D, \nu\right\}}_{X} + \frac{\nu}{{}^{\lfloor d}D}\left[-{}^{\lfloor d}D + {}^{\lfloor d}\tilde{D}\right].
\end{aligned}
$$

The chain rule for expectations, Proposition 2.3.6, and the formula in (8.23) for $\mathcal{E}\left[r^{-1}\right]$ help us to evaluate the last term X.

$$
X = \mathring{\psi} - \mathrm{tr}\left[C\tilde{C}^{-1}\right] + \frac{\nu}{{}^{\lfloor d}D}\left[-\left(\hat{\theta}-\hat{\tilde{\theta}}\right)'\tilde{C}^{-1}\left(\hat{\theta}-\hat{\tilde{\theta}}\right)\right].
$$

We substitute this result and explicit formulas for $\mathcal{I}(L, D, \nu)$ and $\mathcal{I}(\tilde{L}, \tilde{D}, \tilde{\nu})$ into (8.26).

Taking into account the form of $\mathcal{I}(V, \nu)$ and $\mathcal{I}(\tilde{V}, \tilde{\nu})$, the result reads

$$\begin{aligned}
\mathcal{D} \;\; = \;\; & \ln\left(\frac{\Gamma(0.5\tilde{\nu})}{\Gamma(0.5\nu)}\right) - 0.5\ln\left(\frac{\left|{}^{\lfloor\psi}\tilde{D}\right|}{\left|{}^{\lfloor\psi}D\right|}\right) + 0.5\tilde{\nu}\ln\left(\frac{{}^{\lfloor d}D}{{}^{\lfloor d}\tilde{D}}\right) + \\
& +0.5(\nu-\tilde{\nu})\frac{\partial}{\partial(0.5\nu)}\ln\left(\Gamma(0.5\nu)\right) - 0.5\mathring{\psi} + 0.5\mathrm{tr}\left[C\tilde{C}^{-1}\right] + 0.5\frac{\nu}{{}^{\lfloor d}D}\left[\left(\hat{\theta}-\hat{\tilde{\theta}}\right)'\tilde{C}^{-1}\left(\hat{\theta}-\hat{\tilde{\theta}}\right) - {}^{\lfloor d}D + {}^{\lfloor d}\tilde{D}\right].
\end{aligned}$$

The alternative expression relates $L'DL$ decomposition to the LS quantities, see Proposition 8.1.7. □

8.1.4 KL distance of normal pdfs

The KL distance serves us as a good measure when judging distance of pairs of normal factors or components in the data space. It serves in the parameter space too when dealing with so called MT normal factors, see Chapters 12, 13. Its value can be computed as follows.

Proposition 8.1.10 (KL distance of normal pdfs) *The KL distance (2.26) of the normal pdfs $f(d) = \mathcal{N}_d(M, R)$, $\tilde{f}(d) = \mathcal{N}_d(\tilde{M}, \tilde{R})$ is given by the formula*

$$\mathcal{D}\left(f||\tilde{f}\right) = 0.5\left[\ln\left|\tilde{R}R^{-1}\right| - \mathring{d} + \mathrm{tr}\left[R\tilde{R}^{-1}\right] + \left(M-\tilde{M}\right)'\tilde{R}^{-1}\left(M-\tilde{M}\right)\right]. \tag{8.27}$$

Let us consider a pair of normal pdfs in factorized forms

$$f(d_t|d(t-1)) = \prod_{i\in i^*}\mathcal{N}_{d_i}(\theta_i'\psi_{i;t}, r_i),\;\; {}^{\lfloor U}f(d_t|d(t-1)) = \prod_{i\in i^*}\mathcal{N}_{d_i}\left({}^{\lfloor U}\theta_i'\,{}^{\lfloor U}\psi_{i;t},\, {}^{\lfloor U}r_i\right),\; i^* \equiv \{1,\ldots,\mathring{d}\},$$

with known regression vectors $\psi_{i;t}$, ${}^{\lfloor U}\psi_{i;t}$, regression coefficients θ_i', ${}^{\lfloor U}\theta_i'$ and variance r_i, ${}^{\lfloor U}r_i$, $i \in i^*$.

Then, the KL distance, conditioned on $\psi_{i;t}$, ${}^{\lfloor U}\psi_{i;t}$, is

$$\mathcal{D}\left(f(d_i|\psi_{i;t})||\,{}^{\lfloor U}f\left(d_i|\,{}^{\lfloor U}\psi_{i;t}\right)|\psi_{i;t},\,{}^{\lfloor U}\psi_{i;t}\right) = \frac{1}{2}\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right) - 1 + \frac{r_i}{{}^{\lfloor U}r_i} + \frac{\left(\theta_i'\psi_{i;t} - {}^{\lfloor U}\theta_i'\,{}^{\lfloor U}\psi_{i;t}\right)^2}{{}^{\lfloor U}r_i}\right]. \tag{8.28}$$

Proof: The definitions of the KL distance, normal pdfs and $\mathcal{E}[x'Qx] = [\mathcal{E}(x)]'Q\mathcal{E}[x] + \mathrm{tr}[\mathrm{cov}(x)Q]$ imply

$$\begin{aligned}
& \mathcal{D}(f||\tilde{f}) \equiv 0.5\ln\left(\left|\tilde{R}R^{-1}\right|\right) + \\
& + \;\; \int \mathcal{N}_d(M,R)\left[-0.5(d-M)'R^{-1}(d-M) + 0.5\left(d-\tilde{M}\right)'\tilde{R}^{-1}\left(d-\tilde{M}\right)\right] dd = \\
& = \;\; 0.5\ln\left(\left|\tilde{R}R^{-1}\right|\right) - 0.5\mathrm{tr}\left(RR^{-1}\right) + 0.5\left(M-\tilde{M}\right)'\tilde{R}^{-1}\left(M-\tilde{M}\right) + 0.5\mathrm{tr}\left(R\tilde{R}^{-1}\right).
\end{aligned}$$

The factor version is obtained by a direct substitution of moments. □

8.1.5 Estimation and prediction with normal factors

The normal factors predicting a real-valued scalar d belong to the exponential family. Consequently, their estimation and subsequent prediction reduce to algebraic operations, see Proposition 3.2.1.

Proposition 8.1.11 (Estimation and prediction of the normal factor) *Let natural conditions of decision making, Requirement 2.5.1, hold. The treated factor (8.1) is supposed to be normal and a conjugate prior (3.12) $GiW_\Theta(V_0, \nu_0)$ as well as conjugate alternative $GiW_\Theta({}^{\lfloor A}V_t, {}^{\lfloor A}\nu_t)$ in the stabilized* *forgetting, see Section 3.1, are used. Then, the posterior pdf is $GiW_\Theta(V_t, \nu_t)$ and its sufficient statistics evolve according to the recursions*

$$\begin{aligned} V_t &= \lambda(V_{t-1} + \Psi_t\Psi_t') + (1-\lambda)\,{}^{\lfloor A}V_t, \quad V_0 \text{ given by the prior pdf} \\ \nu_t &= \lambda(\nu_{t-1} + 1) + (1-\lambda)\,{}^{\lfloor A}\nu_t, \quad \nu_0 > 0 \text{ given by the prior pdf.} \end{aligned} \tag{8.29}$$

The forgetting factor $\lambda \in [0, 1]$.

The predictive pdf is known as Student pdf. *For any data vector $\Psi = [d, \psi']'$, its values can be found numerically as the ratio*

$$f(d|\psi, V, \nu) = \frac{\mathcal{I}(V + \Psi\Psi', \nu + 1)}{\sqrt{2\pi}\mathcal{I}(V, \nu)}, \quad \text{or} \tag{8.30}$$

$$f(d|\psi, L, D, \nu) = \frac{\Gamma(0.5(\nu+1))\left[{}^{\lfloor d}D(1+\zeta)\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu)\left(1 + \frac{\hat{e}^2}{{}^{\lfloor d}D(1+\zeta)}\right)^{0.5(\nu+1)}}, \quad \text{where} \tag{8.31}$$

$$\begin{aligned} \hat{e} &\equiv d - \hat{\theta}'\psi \equiv \text{prediction error} \\ \hat{\theta} &= {}^{\lfloor \psi}L^{-1}\,{}^{\lfloor d\psi}L, \ \zeta \equiv \psi'\,{}^{\lfloor \psi}L^{-1}\,{}^{\lfloor \psi}D^{-1}\left({}^{\lfloor \psi}L'\right)^{-1}\psi, \ \text{cf. Proposition 8.1.7.} \end{aligned}$$

Proof: The first form is directly implied by Proposition 2.5.2. We use it while

- taking into account that the integrand in the numerator contains one factor $(2\pi)^{-0.5}$ more than that in the denominator,
- exploiting the form of the normalizing factor $\mathcal{I}(L, D, \nu)$, Proposition 8.1.7,
- respecting relationships among various forms of statistics, and basic algebraic relationships, Proposition 8.1.1.

It gives $f(d|\psi, L, D, \nu) =$

$$\begin{aligned} &= \frac{\Gamma(0.5(\nu+1))}{\sqrt{\pi}\Gamma(0.5\nu)}\left|C^{-1}\right|^{0.5\nu}\left|C^{-1} + \psi\psi'\right|^{-0.5\nu}\left({}^{\lfloor d}D\right)^{-0.5}|V|^{0.5(\nu+1)}|V + \Psi\Psi'|^{-0.5(\nu+1)} = \\ &= \frac{\Gamma(0.5(\nu+1))}{\sqrt{\pi}\Gamma(0.5\nu)}|C|^{-0.5\nu}(1+\zeta)^{-0.5\nu}\left({}^{\lfloor d}D\right)^{-0.5}\left(1 + \Psi'L^{-1}D^{-1}\left(L^{-1}\right)'\Psi\right)^{-0.5(\nu+1)} = \\ &= \frac{\Gamma(0.5(\nu+1))\left[{}^{\lfloor d}D(1+\zeta)\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu)}\left(1 + \frac{\hat{e}^2}{{}^{\lfloor d}D(1+\zeta)}\right)^{-0.5(\nu+1)}. \end{aligned}$$

We also used the identity $\Psi'L^{-1}D^{-1}\left(L^{-1}\right)'\Psi = \hat{e}^2/\,{}^{\lfloor d}D + \zeta$. Majority of evaluations is in [14, 80]. □

Remark(s) 8.1.2

1. *The recursion (8.29) is replaced by updating the $L'DL$ decomposition of the extended information matrix according to Algorithm 8.1.2. In order to cope with stabilized forgetting,*
   (a) *the updated D is multiplied by λ,*
   (b) *the data vector Ψ is replaced by $\sqrt{\lambda}\Psi$,*
   (c) *the $L'DL$ decomposition of ${}^{\lfloor A}V_t$ is taken as the sum of weighted dyads formed by*

$$\left(k\text{-}th\_row\_of\_\,{}^{\lfloor A}L_t\right)'(1-\lambda)\,{}^{\lfloor A}D_{k;t}\left(k\text{-}th\_row\_of\_\,{}^{\lfloor A}L_t\right), \quad \text{with} \quad k = 1, \ldots, \mathring{\Psi}.$$

2. *The extensive use of the $L'DL$ decomposition is vital for numerical stability of computations. Experience confirms that without it a safe processing of real data records with target dimensions is impossible. Moreover, the evaluation of the determinants occurring in various pdfs becomes simple with the $L'DL$ decomposition.*

3. *The predictive pdf is the key element needed in approximate estimation, Section 8.5 and for comparing variants through v-likelihood, Section 6.1.2. It is thus worthwhile to check which of the versions (8.30), (8.31) is computationally simpler. The latter one has advantage that the prediction errors* $\{\hat{e}_t\}_{t\in t^*}$ *suit for an intuitive judgement of the predictive properties of the inspected factor.*
4. *The performed evaluations are tightly related to least squares and its recursive version, see e.g. [14].*
5. *The form (8.31) of the Student pdf suits for evaluating likelihood function when LS parameter estimates* $\hat{\theta}$, ${}^{\lfloor d}D$ *are fixed.*

8.1.6 Estimation and prediction with log-normal factors

Applicability of the studied model can be simply extended by modelling and predicting unknown-parameter free, one-to-one, transformations of d_t, [14]. Almost everything remains the same. Only moments of the original d_t have to be derived. *Log-normal parameterized factor* predicting $\ln(d_t)$ represents the most practical instance of this generalization. It allows us to deal with positive data having a relatively heavy-tailed pdf.

The log-normal parameterized factor models a positive scalar variable d by the pdf

$$f(d|\psi,\Theta) = \mathcal{LN}_d(\theta'\psi, r), \text{ where} \tag{8.32}$$

$$\Theta \equiv [\theta, r] \equiv [\text{regression coefficients, noise variance}] \in \Theta^* \subset (\mathring{\psi}\text{-dimensional, non-negative) reals}$$

$$\psi \equiv \text{regression vector}, \quad \Psi = [\ln(d), \psi']' \equiv \text{data vector}$$

$$\mathcal{LN}_d(\theta'\psi, r) \equiv (2\pi r)^{-0.5} d^{-1} \exp\left\{-\frac{(\ln(d) - \theta'\psi)^2}{2r}\right\} = (2\pi r)^{-0.5} d^{-1} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(\Psi\Psi'[-1,\theta']'[-1,\theta']\right)\right\}.$$

Log-normal parameterized factors belong to the exponential family, see Section 3.2, so that they possess conjugate (self-reproducing) prior. The following correspondence to (3.5) holds

$$\begin{aligned} \mathcal{LN}_d(\theta'\psi, r) &= d^{-1} A(\Theta) \exp[\langle B(\Psi), C(\Theta)\rangle] \text{ with} \\ A(\Theta) &\equiv (2\pi r)^{-0.5},\ B(\Psi) \equiv \Psi\Psi',\ C(\Theta) = 2r^{-1}[-1,\theta']'[-1,\theta'],\ \langle B, C\rangle \equiv \mathrm{tr}[B'C]. \end{aligned} \tag{8.33}$$

The additional factor $\exp(D(\Psi)) \equiv d^{-1}$ could be included into $B(\Psi)$. It has, however, no influence on estimation. For this reason, it is kept separately. The correspondence determines the conjugate prior, see (3.12), in the form of the Gauss-inverse-Wishart pdf (GiW), see (8.16). The estimation studied in connection with normal models remains obviously unchanged. The prediction is slightly influenced by the factor d^{-1} in (8.33).

Proposition 8.1.12 (Estimation and prediction of the log-normal factor) *Let natural conditions of decision making, Requirement 2.5.1, hold. The treated factor (8.32) is log-normal and a conjugate prior (3.12)* $GiW_\Theta(V_0, \nu_0)$ *as well as conjugate alternative* $GiW_\Theta({}^{\lfloor A}V_t, {}^{\lfloor A}\nu_t)$ *(see Section 3.1) in the stabilized forgetting are used. Then, the posterior pdf is* $GiW_\Theta(V_t, \nu_t)$ *and its sufficient statistics evolve according to the recursions*

$$\begin{aligned} V_t &= \lambda(V_{t-1} + \Psi_t\Psi_t') + (1-\lambda)\, {}^{\lfloor A}V_t,\ V_0 \text{ given by the prior pdf} \\ \nu_t &= \lambda(\nu_{t-1} + 1) + (1-\lambda)\, {}^{\lfloor A}\nu_t, \quad \nu_0 \text{ given by the prior pdf}. \end{aligned}$$

The predictive pdf is known as log-Student pdf. *Its values can be numerically evaluated, for any data vector* $\Psi = [\ln(d), \psi']'$, *as the ratio*

$$f(d|\psi, V, \nu) = \frac{\mathcal{I}(V + \Psi\Psi', \nu+1)}{\sqrt{2\pi}d\,\mathcal{I}(V,\nu)}, \text{ see Proposition 2.5.2, or}$$

$$f(d|\psi, V, \nu) = \frac{\Gamma(0.5(\nu+1))\left[{}^{\lfloor d}D(1+\zeta)\right]^{-0.5}}{d\sqrt{\pi}\Gamma(0.5\nu)\left(1 + \frac{\hat{e}^2}{{}^{\lfloor d}D(1+\zeta)}\right)^{0.5(\nu+1)}}, \quad \text{where} \tag{8.34}$$

$$\hat{e} \equiv \ln(d) - \hat{\theta}'\psi \equiv \text{prediction error},\ \hat{\theta} = {}^{\lfloor\psi}L^{-1}\, {}^{\lfloor d\psi}L,\ \zeta = \psi'\, {}^{\lfloor\psi}L^{-1}\, {}^{\lfloor\psi}D^{-1}\left({}^{\lfloor\psi}L'\right)^{-1}\psi.$$

Proof: It just copies Proposition 8.1.11 with the factor d^{-1} included and $\ln(d)$ replacing d. □

8.1.7 Relationships of a component to its factors

We deal systematically with the factor-based description of components $f(d_t|\phi_{t-1}, \Theta)$ using the chain rule, see Agreement 5.2.1,

$$f(d_t|\phi_{t-1}, \Theta) = \prod_{i \in i^*} f(d_{i;t}|\psi_{i;t}, \Theta_i).$$

For specification of simulation examples and for interpretation of estimation results, it is useful to relate the normal component to its factors in a detail. This task is addressed here.

It is known that conditional pdfs of a normal pdf are again normal. Thus, it is sufficient to inspect first and second moments.

Proposition 8.1.13 (Relationships of normal components and its factors)
I. Let us consider the matrix form of a normal component [82]

$$f(d_t|\phi_{t-1}, {}^{\llcorner M}\theta, {}^{\llcorner e}R) = \mathcal{N}_{d_t}({}^{\llcorner M}\theta'\phi_{t-1}, {}^{\llcorner e}L' \, {}^{\llcorner e}D \, {}^{\llcorner e}L), \quad \textit{where} \tag{8.35}$$

${}^{\llcorner M}\theta$ *is* $(\mathring{\phi}, \mathring{d})$*-matrix of unknown regression coefficients,*

${}^{\llcorner e}L$, ${}^{\llcorner e}D$ *define the* $L'DL$ *decomposition of the unknown noise-covariance matrix* ${}^{\llcorner e}R$.

Then, i*-th parameterized factor is*

$$f(d_{i;t}|\psi_{i;t}, \Theta_i) = \mathcal{N}_{d_{i;t}}(\theta_i'\psi_{i;t}, {}^{\llcorner e}D_i), \quad \textit{where} \tag{8.36}$$

$\psi_{i;t} = [d'_{(i+1)\cdots\mathring{i};t}, \phi'_{t-1}]' = [d_{i+1;t}, \psi'_{i+1;t}]'$, $i < \mathring{d}$, $\psi_{\mathring{d};t} \equiv \phi_{t-1}$ *and*

$\theta_i' = \left[({}^{\llcorner e}L')^{-1} - I, \ ({}^{\llcorner e}L')^{-1} \, {}^{\llcorner M}\theta'\right]_i$, *where the operation* $[\,]_i$ *selects* i*-th row of the matrix in its argument and omits leading zeros.*

II. Let the normal parameterized factors

$$f(d_{i;t}|\psi_{i;t}, \Theta_i) = \mathcal{N}_{d_{i;t}}(\theta_i'\psi_{i;t}, r_i) \tag{8.37}$$

be ordered so that $\psi_{i;t}$ *is a sub-vector of* $\left[d'_{(i+1)\cdots\mathring{i};t}, \phi'_{t-1}\right]'$. *Let us extend regression coefficients* θ_i *to* $\bar{\theta}_i$ *by inserting zeros at appropriate places so that* $\theta_i'\psi_{i;t} = \bar{\theta}_i' \left[d_t', \phi_{t-1}'\right]'$. *The vector* $\bar{\theta}_i'$ *has at least* i *leading zeros. Let us create*

$(\mathring{d}, \mathring{d})$*-upper triangular matrix* η_1 *having unit diagonal and* $-\bar{\theta}_{ij}$ *value on the position* $1 < i < j \leq \mathring{d}$,

$(\mathring{d}, \mathring{\phi})$*-matrix* η_2 *with the value* $\bar{\theta}_{i(j+\mathring{d})}$ *on the position* $i \in \{1, \ldots, \mathring{d}\}$, $j \in \{1, \ldots, \mathring{\phi}\}$.

Then, these factors form normal component (8.35) with the moments

$${}^{\llcorner e}L' = \eta_1^{-1}, \quad {}^{\llcorner M}\theta' = {}^{\llcorner e}L'\eta_2 \quad \textit{and} \quad {}^{\llcorner e}D_i = r_i. \tag{8.38}$$

Proof: By a direct comparison of moments. □

The derived relationships suit whenever uncertainty of parameters can be neglected. This is the case, for instance, when we specify a simulated component.

The situation is more complex when information on parameters Θ was obtained through a parameter estimation. Then, the predictive pdfs describing individual factors have Student pdf (8.31). It is known that this pdf resembles the normal pdf with the same moments whenever ν is high enough. The values above 20 make this approximation excellent. Thus, for a given factor, we get the correspondence required, see (8.23) and (8.31),

$$\hat{\theta} \leftrightarrow \theta \quad \text{and} \quad \hat{r}(1+\zeta) \leftrightarrow r. \tag{8.39}$$

This observation gives the algorithm for creating approximate normal component from incomplete estimated factors.

Algorithm 8.1.4 (Approximate normal component composed of uncertain factors)

1. *For all factors* $i \in i^*$
   - *Compute* $\theta_i \equiv {}^{\lfloor\psi}L_i^{-1}\,{}^{\lfloor d\psi}L_i$ *and* $h_i = {}^{\lfloor\psi}L_i^{-1}\psi_{i;t}$.
   - *Define* $r_i \equiv \frac{{}^{\lfloor d}D_i}{\nu_i}\left(1+\sum_{j=1}^{\mathring{d}}\frac{h_{ij}^2}{{}^{\lfloor\psi}D_{ij}}\right)$
2. *Apply the transformation described in II. of Proposition 8.1.13.*

Remark(s) 8.1.3

1. *The variance of the factor output, cf. Algorithm 8.1.4, is greater than the estimate* $\hat{r}$ *of the noise variance. The increase by the factor* $1+\zeta_{i;t}$ *reflects a projection of parameter uncertainties to the direction of the regression vector* $\psi_{i;t}$. *The increase is obviously data dependent.*
2. *The needed inversion of the triangular matrix* ${}^{\lfloor\psi}L$ *is efficiently performed by a backward elimination with two-column right-hand side of the set of linear equations* ${}^{\lfloor\psi}L\left[\hat{\theta},h\right] = \left[{}^{\lfloor d\psi}L,\psi\right]$.

8.1.8 Prediction and model selection

This subsection provides predictors based on normal mixtures. They are repeatedly used for selection of a model among several available alternatives. Proposition 8.1.11 gives immediately a normal variant of Proposition 6.1.1.

Proposition 8.1.14 (Normal mixture-based *one-step-ahead predictor*) *Let the prior and posterior estimates of the normal mixture have the form (6.3)*

$$f(\Theta|d(t)) = Di_\alpha(\kappa_t)\prod_{c\in c^*}\prod_{i\in i^*} f(\Theta_{ic}|d(t)),\ t\in\{0\}\cup t^*.$$

The Dirichlet pdf $Di_\alpha(\kappa_t)$ *is the estimate of component weights* α, *Agreement 5.2.1, and has the expected value*

$$\hat{\alpha}_{c;t} = \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}}.$$

Let the estimates of the individual normal factors be GiW pdfs (8.16)

$$f(\Theta_{ic}|d(t)) = GiW_{\theta_{ic},r_{ic}}(L_{ic;t},D_{ic;t},\nu_{ic;t}).$$

Then, the estimation within the adaptive advisory system provides the value of the one-step-ahead predictor at a possible data point d_{t+1} *in the form*

$$\begin{aligned}
f(d_{t+1}|d(t)) &= \sum_{c\in c^*}\hat{\alpha}_{c;t}\prod_{i\in i^*} f(d_{ic;t+1}|d(t),c) \quad \textit{with} \qquad (8.40)\\
f(d_{ic;t+1}|d(t),c) &= \frac{\mathcal{I}(V_{ic;t}+\Psi_{ic;t+1}\Psi'_{ic;t+1},\nu_{ic;t}+1)}{\sqrt{2\pi}\,\mathcal{I}(V_{ic;t},\nu_{ic;t})}\\
\mathcal{I}(V_{ic;t},\nu_{ic;t}) &\equiv \mathcal{I}\left(L_{ic;t},D_{ic;t},\nu_{ic;t}\right) = \Gamma\left(0.5\nu_{ic;t}\right)\left({}^{\lfloor d}D_{ic;t}\right)^{-0.5\nu_{ic;t}}\left|{}^{\lfloor\psi_{ic}}D_{ic;t}\right|^{-0.5} 2^{0.5\nu_{ic;t}}(2\pi)^{0.5\mathring{\psi}_{ic}}\\
V_{ic;t} &\equiv L'_{ic;t}D_{ic;t}L_{ic;t}\ \textit{is the extended information matrix of the factor ic at time } t\\
\nu_{ic;t} &\equiv \textit{the number of degrees of freedom of the factor ic at time } t. \qquad (8.41)
\end{aligned}$$

The predictive pdf for a factor ic can be given an alternative form (8.31). We use it explicitly in the following specification of the predictor in the fixed advisory system, *when the measured data do not enter* *the condition of the pdf on parameters ($f(\Theta|d(t)) \equiv f(\Theta)$). The mixture predictor for the fixed advisory system is*

$$
\begin{aligned}
f(d_{t+1}|d(t)) &\equiv \sum_{c\in c^*} \hat{\alpha}_{c;0} \prod_{i\in i^*} \underbrace{\int f(d_{ic;t+1}|\psi_{ic;t+1},\Theta_{ic},c) f(\Theta_{ic})\, d\Theta_{ic}}_{f(d_{ic;t+1}|\psi_{ic;t+1},c)}, \text{ where}\\
f(d_{ic;t+1}|\psi_{ic;t+1},c) &= \frac{\Gamma(0.5(\nu_{ic;0}+1))\left[{}^{\lfloor d}D_{ic;0}(1+\zeta_{ic;t+1})\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu_{ic;0})\left(1+\frac{\hat{e}^2_{ic;t+1}}{{}^{\lfloor d}D_{ic;0}(1+\zeta_{ic;t+1})}\right)^{0.5(\nu_{ic;0}+1)}}\\
\hat{e}_{ic;t+1} &= d_{ic;t+1} - \hat{\theta}'_{ic;0}\psi_{ic;t+1} \equiv \text{prediction error of the fixed predictor} \qquad (8.42)\\
\zeta_{ic;t+1} &= \psi'_{ic;t+1}\, {}^{\lfloor \psi_{ic}}L^{-1}_{ic;0}\, {}^{\lfloor \psi_{ic}}D^{-1}_{ic;0} \left({}^{\lfloor \psi_{ic}}L'_{ic;0}\right)^{-1} \psi_{ic;t+1}.
\end{aligned}
$$

In both types of advisory systems, $D = \mathrm{diag}\left[{}^{\lfloor d}D,\ {}^{\lfloor \psi}D\right]$, $L = \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L & {}^{\lfloor \psi}L \end{bmatrix}$, $\hat{\theta} = {}^{\lfloor \psi}L^{-1}\, {}^{\lfloor d\psi}L$.

Proof: Omitted. □

8.1.9 Likelihood on variants

During the design phase of the p-system, predictors serve for selecting the best initialization, forgetting, structure etc. For the adaptive p-system, the v-likelihood is the product of one-step-ahead predictors. For the fixed p-system, such a product is an approximate v-likelihood, see Section (6.1.2).

It is good news that in both cases one-step-ahead predictors of the mixture do not reflect the way how statistics were collected. Thus, their product serves well as v-likelihood even when comparing various estimation variants. Also, the involved predictors $f(d_{ic;t+1}|d(t),c)$ or $f(d_{ic;t+1}|\psi_{ic;t+1},c)$ at the factor level can be directly used for evaluating the entropy rates as defined in Agreement 6.1.2.

8.1.10 Branch and bound techniques

For normal parameterized factors with conjugate prior pdfs, various estimates of factors $GiW_{\theta,r}(V,\nu)$ are alternative descriptions inspected and compared. Thus, alternative sets of sufficient statistics V,ν, or their factorized and shifted equivalents L,D,ν, are branched and/or bounded as outlined in Section 6.1.3. The maximized functional F then becomes the function, namely, the v-likelihood $f(d(\mathring{t})|V,\nu)$ of data $d(\mathring{t})$ conditioned on the variant of sufficient statistics V,ν. These statistics determine the $GiW_{\theta,r}(V,\nu)$ pdf that are used for eliminating unknown parameters

$$f(d(\mathring{t})|V,\nu) = \int f(d(\mathring{t})|\theta,r) GiW_{\theta,r}(V,\nu)\, d\theta\, dr.$$

Evaluation of these v-likelihood values for normal mixtures is implied by Proposition 8.1.14. Specific versions of the branch and bound techniques are described in Section 8.4.

8.2 Data preprocessing

Data preprocessing is universally discussed in Chapter 6. Here, just some specific comments are made.

8.2.1 Use of physical boundaries

Normal pdfs are symmetric around expected value and positive on whole real line. The tails of this pdf fall down rather quickly. This can be exploited when respecting known physical boundaries. Individual signals should be scaled and shifted according to the physically expected range. For standardization purposes, it is useful to scale them so that the middle of this range is mapped on value 0 and 70% boundaries to value 1. It simplifies the universal outlier detection.

8.2.2 Removal of high frequency noise

Normal mixtures can be perceived as a collection of local normal regression models. Their estimation is known to be sensitive to presence of a high frequency measurement noise. It has to be removed before using these data for the normal mixture estimation. The general local filters exploiting over-sampling of data are expected to suit to this task. Use of normal *ARMAX factors* is worth of being considered. Recall that ARMAX model is an extension of the normal regression model that allows colored noise modelled by moving average process. At least the case with the known MA part, [65], is directly applicable. A promising extension to the unknown MA part is described in [64].

8.2.3 Suppression of outliers

Outliers strictly contradict the nature of the normal model that has very light tails. Thus, their removal is crucial for obtaining reliable results. It corresponds with a well known observation that least squares, that form the algorithmic core of estimation of normal mixtures, are sensitive to outlying data. Local filters [83] and simple mixtures [67] may serve for the discussed removal.

8.3 Use of prior knowledge at factor level

Quantification of prior knowledge at the factor level, Section 6.3, is specialized here to the normal factor.

8.3.1 Internally consistent fictitious data blocks

Processing of internally consistent data blocks coincides with learning normal factors, see Section 8.1.5. Thus, we have to quantify individual knowledge items only and then to specialize merging of the individual estimates $f(\Theta|K_k)$ expressing knowledge items $K(\mathring{k})$.

8.3.2 Translation of input-output characteristics into data

Quantification of the knowledge of the initial moments (6.28) of the predictive pdf $f(d|\psi)$ given by a fixed regression vector ψ is assumed

$$\hat{d} = \int d\, f(d|\psi)\, dd,\ r_d = \int \left(d - \hat{d}\right)^2 f(d|\psi)\, dd. \tag{8.43}$$

It corresponds with the knowledge, see (6.27), $h(\psi) = \left[\hat{d}, r_d\right]$, $H(\Psi) = \left[d, (d-\hat{d})^2\right]$. The function $g(\psi,\Theta)$ that determines the optimal quantification of this knowledge, Proposition 6.3.1, has the form

$$g(\psi,\Theta) = \left[\theta'\psi, (\theta'\psi - \hat{d})^2 + r\right]. \tag{8.44}$$

With this function, the pdf $f(\Theta|K)$ constructed according to Proposition 6.3.1 falls out of the *GiW* class even with the pre-prior pdf chosen within it. For this reason, we perform the optimization described in the cited proposition under the additional requirement: the optimum is searched for within the *GiW* class.

Proposition 8.3.1 (Knowledge of input-output characteristics for normal factors) *Let us consider GiW pdfs determined by least-squares statistics $\hat{\theta}, \hat{r}, C, \nu$, see Proposition 8.1.7.*

The GiW pdf that fulfills restrictions (8.43) and minimizes the KL distance to the flat pre-prior GiW – given by zero parameters estimate, noise variance $\varepsilon > 0$, covariance factor of LS estimate $1/\varepsilon^2 I$ and degrees of freedom $\nu = \varepsilon < 1$ – has the least-squares statistics

$$\hat{\theta} \ = \ \frac{\hat{d}\psi}{\psi'\psi},\quad C^{-1} = \varepsilon^2(I + x\psi\psi'),\quad \hat{r} = r_d\frac{\varepsilon^2(1+x\psi'\psi)}{\varepsilon^2(1+x\psi'\psi)+\psi'\psi},\ \textit{where} \tag{8.45}$$

x is the unique (largest) solution of the equation

$$x+\frac{\varepsilon^2(1+x\psi'\psi)}{\varepsilon^2(1+x\psi'\psi)+\psi'\psi}=\frac{\frac{\hat{d}^2}{\psi'\psi}+1/\varepsilon}{r_d}. \tag{8.46}$$

For $a\equiv(\psi'\psi)^{-1},\ b\equiv r_d^{-1}\left[\hat{d}^2a+\varepsilon^{-1}\right]$, *it has the form*

$$x=0.5\left[-(a+\varepsilon^{-2}+1-b)+\sqrt{(a+\varepsilon^{-2}+1-b)^2+4a(b+ba-1)}\right]. \tag{8.47}$$

Degrees of freedom ν should be unchanged.

Proof: The proof is straightforward but lengthy and error prone. For this reason, we present it in detail.

1. We express the optimized KL distance (8.25) in terms of least-square statistics. At the same time we re-arrange it, insert the considered specific form of $\bar{f}$ and omit terms not influencing optimization. It gives the minimized function in the form

$$\begin{aligned} F(\hat{\theta},\hat{r},C,\nu) &= \underbrace{-\ln(\Gamma(0.5\nu))+0.5(\nu-\varepsilon)\frac{\partial}{\partial(0.5\nu)}\ln\left(\Gamma(0.5\nu)\right)+0.5\varepsilon^2\ln(\nu)-0.5\nu+}_{\tilde{F}(\nu)} \\ &+ 0.5\left(-\ln|C|+\varepsilon^2\mathrm{tr}[C]\right)+0.5\varepsilon^2\ln(\hat{r})+0.5\frac{1}{\hat{r}}\left[\varepsilon^2\hat{\theta}'\hat{\theta}+\varepsilon\right]. \end{aligned}$$

2. Restrictions on moments of the predictor are

$$\hat{d} = \hat{\theta}'\psi,\ \ r_d=\hat{r}\left(1+\psi'C\psi\right)\equiv\hat{r}(1+\zeta),\ \zeta\equiv\psi'C\psi.$$

3. The minimization of $F(\cdot)$ with respect to $\hat{\theta}$ gives directly $\hat{\theta}=\frac{\hat{d}\psi}{\psi'\psi}$ irrespective of other optimized variables.

4. Inserting the found $\hat{\theta}$ into the optimized functional and using the second restriction for expressing $\hat{r}$, we get the following function to be minimized with respect to C.

$$F(C) = 0.5\left(-\ln|C|+\varepsilon^2\mathrm{tr}[C]\right)-0.5\varepsilon^2\ln(1+\zeta)+0.5\zeta\varepsilon^2B,\ \ B\equiv\frac{\frac{\hat{d}^2}{\psi'\psi}+\varepsilon^{-1}}{\lfloor d_r}.$$

Taking its derivatives with respect to C, using $\frac{\partial\ln|C|}{\partial C}=C^{-1}$, see (8.6), we get necessary condition for minimum

$$C^{-1} = \varepsilon^2\left[I+\left(B-\frac{1}{1+\zeta}\right)\psi\psi'\right].$$

Denoting $x=B-\frac{1}{1+\zeta}$, we get the optimal form of C.

5. We solve the above implicit equation ($\zeta=\psi'C\psi$). Using the matrix inversion lemma, formula (8.5), and the definition of ζ, we get

$$\zeta=\varepsilon^{-2}\frac{\psi'\psi}{1+x\psi'\psi}.$$

Definition of x in the previous step and an algebraic rearrangement give

$$x+1/(1+\zeta)=B\Rightarrow x+\frac{1}{1+\varepsilon^{-2}\frac{x\psi'\psi}{1+x\psi'\psi}}=B\Rightarrow x+\frac{\varepsilon^2(1+x\psi'\psi)}{\varepsilon^2(1+x\psi'\psi)+x\psi'\psi}=B.$$

This is the equation to be proved. It can be converted into quadratic equation for x and solved but this forms shows clearly that it has solution. Its right-hand side is always positive. Left-hand side can be made arbitrarily large for positive x and for $x<0$ is negative. Within the admissible range $x>-(\psi'\psi)^{-1}$, that makes the constructed C positive definite, the left hand side is continuous. Consequently, a real admissible solution exists. A direct inspection shows that just one formal solution guarantees positive definiteness of C, i.e. $x>-1/(\psi'\psi)$. Equation (8.47) gives the explicit form of the solution.

6. Expressing $\hat{r}$ in term of x, we get
$$\hat{r} = \frac{r_d}{1+\zeta} = \frac{r_d \varepsilon^2(1+x\psi'\psi)}{\varepsilon^2(1+x\psi'\psi)+\psi'\psi}.$$

7. It remains to find minimizing argument of the part $\tilde{F}(\nu)$ that depends on ν only. The necessary condition for extreme reads
$$1 = 0.5(\nu-\varepsilon)\frac{\partial^2 \ln(\Gamma(0.5\nu))}{\partial^2(0.5\nu)} + \frac{\varepsilon}{\nu}.$$
It is fulfilled for $\nu = \varepsilon$. The explicit evaluation of the factor at $\nu - \varepsilon$ and use of the Jensen inequality imply that the factor at $\nu = \varepsilon$ is positive. Thus, the right-hand side of the inspected equation is equal to 1 for $\nu = \varepsilon$ only.

□

8.3.3 Merging of knowledge pieces

We have got a collection of factor estimates $f(\Theta|K_k) = GiW_{\theta,r}(V_k,\nu_k)$ $k \in k^* \equiv \left\{1,\ldots,\mathring{k}\right\}$ after processing individual, internally consistent, data blocks and individual knowledge items. We have to use them for constructing of a single posterior pdf $\hat{f}(\Theta|d(\mathring{t}),K(\mathring{k})) = GiW_{\theta,r}(V_{\mathring{t}},\nu_{\mathring{t}})$ that use all of them together with available real data $d(\mathring{t})$. For this, the general Proposition 6.3.2 is applied. It leads directly to the GiW counterpart of Algorithm 6.3.2.

Algorithm 8.3.1 (Merging GiW information sources)

1. *Construct the prior factor estimates $GiW_{\theta,r}(V_k,\nu_k)$ reflecting internally consistent data blocks by applying ordinary Bayesian estimation or reflecting individual knowledge pieces and constructed according to Proposition 8.3.1. The least-square representation of the later case has to be transformed to the V,ν description using relationships (8.17).*

2. *Evaluate the likelihood function $\mathcal{L}(\Theta,d(\mathring{t})) = GiW_{\theta,r}(V_{0;\mathring{t}},\nu_{0;\mathring{t}})$ by applying Proposition 8.1.11 with zero initial conditions for V,ν.*

3. *Evaluate the posterior pdfs $f(\Theta|d(t),K_k) = GiW_{\theta,r}(V_{0;\mathring{t}}+V_k,\nu_{0;\mathring{t}}+\nu_k)$ and the v-likelihood*
$$f(d(\mathring{t})|K_k) = \frac{\mathcal{I}(V_{0;\mathring{t}}+V_k,\nu_{0;\mathring{t}}+\nu_k)}{\mathcal{I}(V_k,\nu_k)}$$
corresponding to the prior pdfs $f(\Theta|K_k) = GiW_{\theta,r}(V_k,\nu_k)$ $k \in k^*$. The integral $\mathcal{I}(V,\nu)$ is given by the formula (8.22) expressed in terms of practically used $L'DL$ decomposition of information matrix V. The regularization by a flat pre-prior pdf $GiW_{\theta,r}(\varepsilon I,\varepsilon)$, given by small $\varepsilon > 0$ has to be considered if some V_k is singular.

4. *Evaluate the weights*
$$\beta_{k|d(\mathring{t})} = \frac{f(d(\mathring{t})|K_k)}{\sum_{\tilde{k}\in k^*} f(d(\mathring{t})|K_{\tilde{k}})} = \frac{\frac{\mathcal{I}(V_{0;\mathring{t}}+V_k,\nu_{0;\mathring{t}}+\nu_k)}{\mathcal{I}(V_k,\nu_k)}}{\sum_{\tilde{k}\in k^*}\frac{\mathcal{I}(V_{0;\mathring{t}}+V_{\tilde{k}},\nu_{0;\mathring{t}}+\nu_{\tilde{k}})}{\mathcal{I}(V_{\tilde{k}},\nu_{\tilde{k}})}},\ k \in k^*.$$

5. *Determine the merger as the posterior pdf to be used*
$$\begin{aligned}\hat{f}(\Theta|d(\mathring{t}),K(\mathring{k})) &= GiW_{\theta,r}\left(V_{0;\mathring{t}} + \sum_{k\in k^*}\beta_{k|d(\mathring{t})}V_k, \nu_{0;\mathring{t}} + \sum_{k\in k^*}\beta_{k|d(\mathring{t})}\nu_k\right) = \\ &= GiW_{\theta,r}(V_{0;\mathring{t}},\nu_{0;\mathring{t}})GiW_{\theta,r}\left(\sum_{k\in k^*}\beta_{k|d(\mathring{t})}V_k, \sum_{k\in k^*}\beta_{k|d(\mathring{t})}\nu_k\right).\end{aligned}$$
Notice that the effectively used "prior pdf" is seen in the above formula.

8.4 Construction of the prior estimate

Here, relevant Gaussian counterparts of Section 6.4 are presented.

8.4.1 Iterative construction of the prior pdf

The danger of a plain repetitive use of the Bayes rule, discussed at general level in Section 6.4.1, becomes even more obvious when applied to normal factors. Within the class of conjugate GiW pdfs, $\mathring{n}$-th approximate evaluation of the posterior pdf gives

$$GiW_{\theta,r}\left(\sum_{n=1}^{\mathring{n}} V_{n;\mathring{t}}, \sum_{n=1}^{\mathring{n}} \nu_{n;\mathring{t}}\right).$$

Taking into account basic properties of the GiW pdf, Proposition 8.1.7, especially, second moments, it can be seen, that the point estimates of the unknown $\Theta = (\theta, r)$ are around non-weighted averages corresponding to the sum of the individual extended information matrices. Their apparent uncertainty, however, decreases proportionally to $\mathring{n}\mathring{t}$, i.e. quite quickly. This observation clarifies why the flattening, see Section 6.4.3 may help. It acts as forgetting that prefers newer, hopefully better, values of $V_{n;\mathring{t}}$ and prevents the cumulative degrees of freedom $\nu = \sum_{n=1}^{\mathring{n}} \nu_{n;\mathring{t}}$ to grow too quickly.

8.4.2 Common bounding mapping

As said in Section 6.4.2, the common bounding mapping just cancels the "least fit candidates". For normal mixtures, the posterior pdfs are described by the vector statistics κ, related to the component weights, and by the collection of the extended information matrices V_{ic} together with degrees of freedom ν_{ic}, $i \in i^* \equiv \{1, \ldots, \mathring{d}\}$, $c \in c^*$. These are finite-dimensional objects. They may contain, however, a huge amount entries. Taking, for instance, the simplest static case without common factors, the single alternative represents $\mathring{c}[1 + (\mathring{d} + 1)(\mathring{d} + 2)/2]$ numbers. For the target dimensions $30 - 40$, it gives about 1000 numbers even for $\mathring{c} = 2$. It indicates that we face two contradictory requirements on the bounding mapping. We have to

- preserve as much information as possible from previous iteration steps,
- store and treat as little as possible alternatives.

We found no general guideline how to reach the adequate compromise. At present, we prefer the second item and bound the number of alternatives to two, at most to three.

8.4.3 Flattening mapping

We need here just to specialize the flattening operation to the GiW pdfs that serve as estimates of normal factors.

Proposition 8.4.1 (Optimal flattening mapping for GiW pdfs) *Let $\tilde{f} = GiW_{\theta,r}\left(\tilde{V}, \tilde{\nu}\right)$ and $\bar{f} = GiW_{\theta,r}\left(\bar{V}, \bar{\nu}\right)$ be a pair of GiW pdfs defined on the common support $\Theta^* = [\theta, r]^*$. Then, the pdf $\hat{f}$ that minimizes the functional $\mathcal{D}\left(\hat{f}||\tilde{f}\right) + q\mathcal{D}\left(\hat{f}||\bar{f}\right)$, $q > 0$ is $GiW_{\theta,r}(\hat{V}, \hat{\nu})$ with*

$$\hat{V} = \Lambda\tilde{V} + (1-\Lambda)\bar{V}, \ \hat{\nu} = \Lambda\tilde{\nu} + (1-\Lambda)\bar{\nu} \ \text{ and } \ \Lambda \equiv 1/(1+q) \in (0,1). \tag{8.48}$$

Proof: Omitted. □

The application of this result to whole mixtures is straightforward as the posterior pdf of its parameters is product of the posterior estimates of weights α and of the individual factors, cf. Agreement 6.1.1.

Proposition 8.4.2 (Optimal flattening of estimates of normal factors) *Let parameters of a pair of normal mixtures be described (6.3) with factor estimates* $\tilde{f}(\theta_{ic}, r_{ic}) = GiW_{\theta_{ic},r_{ic}}\left(\tilde{V}_{ic}, \tilde{\nu}_{ic}\right)$, *and* $\bar{f}(\theta_{ic}, r_{ic}) = GiW_{\theta_{ic},r_{ic}}\left(\bar{V}_{ic}, \bar{\nu}_{ic}\right)$, $i \in i^*, c \in c^*$. *These factor estimates are assumed to have a common support* $[\theta_{ic}, r_{ic}]^*$, *i.e. the common structure of* θ_{ic}.

The corresponding estimates of the component weights are assumed to be described by $\tilde{f}(\alpha) = Di_\alpha(\tilde{\kappa})$ *and* $\bar{f}(\alpha) = Di_\alpha(\bar{\kappa})$, *again of a common structure.*

Then, the pdf $\hat{f}$ *minimizing the functional*

$$\mathcal{D}\left(\hat{f}||\tilde{f}\right) + q\mathcal{D}\left(\hat{f}||\bar{f}\right),\ q > 0,$$

preserves the original form described in Agreement 6.1.1 with GiW factors. The resulting pdf is determined by the statistics, $i \in i^*,\ c \in c^*$,

$$\begin{aligned} \hat{V}_{ic} &= \Lambda\tilde{V}_{ic} + (1-\Lambda)\bar{V}_{ic},\ \hat{\nu}_{ic} = \Lambda\tilde{\nu}_{ic} + (1-\Lambda)\bar{\nu}_{ic} \text{ with } \Lambda \equiv 1/(1+q) \in (0,1) && (8.49)\\ \hat{\kappa}_c &= \Lambda\tilde{\kappa}_c + (1-\Lambda)\bar{\kappa}_c. && (8.50)\end{aligned}$$

Proof: Omitted. Just notice that the flattening of the component-weights estimates is universal. It preserves the Dirichlet form with statistics being a convex combination of the individual statistics involved. □

The choice of the optional flattening factor is fully described by Algorithms 6.4.2, 6.4.3 and Proposition 6.4.4. Knowledge of GiW forms brings nothing new in this respect.

8.4.4 Geometric mean as branching mapping

Geometric-mean branching mapping $\mathcal{A}$ (6.17) maps a pair of arguments $\hat{f}_n(\Theta)$, $n = 1, 2$, of the maximized functional (6.40) ${}^{\lfloor h}\mathcal{L}\left(f_n(d(\mathring{t})|\Theta)\hat{f}_n(\Theta), d(\mathring{t})\right) \equiv \hat{f}_n(d(\mathring{t})) \equiv \int f_n(d(\mathring{t})|\Theta)\hat{f}_n(\Theta)\, d\Theta$ on a new, hopefully better, candidate $\hat{f}_{n+1}(\Theta)$. We apply it, assuming that Agreement 6.1.1 is met for all involved pdfs and that GiW pdfs serve as the estimates of individual parameterized factors.

Proposition 8.4.3 (Geometric mean of a pair GiW pdfs) *Let Agreement 6.1.1 be met for*

$$\hat{f}_n(\Theta|d(t)) \equiv Di_\alpha(\kappa_{n;t}) \prod_{c\in c^*}\prod_{i\in i^*} GiW_{\theta_{ic},r_{ic}}(L_{icn;t}, D_{icn;t}, \nu_{icn;t}),\ t \in 0 \cup t^*,\ n = 1, 2.$$

These pdfs are obtained from a pair of different prior pdfs when estimating approximately, Section 8.5, the normal mixture

$$f(d_t|d(t-1), \Theta) \equiv \sum_{c\in c^*} \alpha_c \prod_{i\in i^*} \mathcal{N}_{d_{i;t}}\left(\theta'_{ic}\psi_{ic;t}, r_{ic}\right).$$

Recall, $Di_\alpha(\kappa)$ *denotes Dirichlet pdf of* $\alpha \in \alpha^* \equiv \left\{\alpha_c \geq 0 : \sum_{c\in c^*}\alpha_c = 1\right\}$ *of the form (6.3)*

$$Di_\alpha(\kappa) \equiv \frac{\prod_{c\in c^*}\alpha_c^{\kappa_c - 1}}{\mathcal{B}(\kappa)},\ \mathcal{B}(\kappa) \equiv \frac{\prod_{c\in c^*}\Gamma(\kappa_c)}{\Gamma(\sum_{c\in c^*}\kappa_c)}.$$

The corresponding expected value of the component weights α *is*

$$\hat{\alpha}_{c;t} = \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}}.$$

The geometric means $\hat{f}_3(\Theta|d(t))$, $t \in t^*$, *constructed according Section 6.4.4, preserve the functional form (6.3). Its factors have statistics*

$$\begin{aligned} V_{3ic;t} &= \lambda_t V_{1ic;t} + (1-\lambda_t)V_{2ic;t} && (8.51)\\ \nu_{3ic;t} &= \lambda_t\nu_{1ic;t} + (1-\lambda_t)\nu_{2ic;t} \\ \kappa_{3c;t} &= \lambda_t\kappa_{1c;t} + (1-\lambda_t)\kappa_{2c;t}, \quad \text{where} \\ \lambda_t &= \left[1 + \prod_{\tau=1}^{t}\frac{\sum_{c\in c^*}\hat{\alpha}_{2c;\tau-1}\prod_{i\in i^*}\frac{\mathcal{I}(L_{2ic;\tau}, D_{2ic;\tau}, \nu_{2ic;\tau})}{\mathcal{I}(L_{2ic;\tau-1}, D_{2ic;\tau}, \nu_{2ic;\tau-1})}}{\sum_{\tilde{c}\in c^*}\hat{\alpha}_{1\tilde{c};\tau-1}\prod_{\tilde{i}\in i^*}\frac{\mathcal{I}(L_{1\tilde{i}\tilde{c};\tau}, D_{1\tilde{i}\tilde{c};\tau}, \nu_{1\tilde{i}\tilde{c};\tau})}{\mathcal{I}(L_{1\tilde{i}\tilde{c};\tau-1}, D_{1\tilde{i}\tilde{c};\tau}, \nu_{1\tilde{i}\tilde{c};\tau-1})}}\right]^{-1}. \end{aligned}$$

The normalization integral $\mathcal{I}(L, D, \nu)$ *is given by the formula (8.22).*

Proof: Geometric bounding is directly applied with the explicitly expressed v-likelihood taken as the product of adaptive one-step-ahead predictors. □

8.4.5 Random branching of statistics

The deterministic algorithms are prone to find just a local optimum. It is known that the global optimization might be guaranteed if a random search is used. Computational complexity prevents us to use the safe, completely random repetitive search. Thus, we support a combined use of deterministic and random searches. Here, we discuss this possibility in connection with normal factors.

According to Agreement 6.1.1, we deal with the mixture estimate $f(\Theta|d(\mathring{t}))$ that is the product of the estimate of component weights Di_α and the estimates of factors that are GiW pdfs for normal mixtures. The number of factors may be relatively large as it is mainly determined by the dimension $\mathring{d}$ of data records. Thus, random generating of variants should be and often can be reduced to their small sub-selection.

The GiW estimate of a factor is determined by the finite-dimensional, approximately sufficient, statistics L, D, ν. $L \in L^* \equiv$ set of lower triangular matrices with unit diagonal, $D \in D^* \equiv$ set of a diagonal matrices with positive diagonal entries and $\nu \in \nu^* \equiv$ set of positive real scalars.

Knowing this, we generate a random sample $\tilde{L}, \tilde{D}, \tilde{\nu}$ in L^*, D^*, ν^* for selected factors whose predictive ability is to be challenged. Even for a single factor, the possible extent of the random choice is usually prohibitive. Thus, it makes sense to exploit interpretation of parts of these statistics and change them partially only. At present, it seems to be wise, cf. Proposition 8.1.7:

- To perform estimation of the factor structure before generating new statistics.
- To preserve characteristics determining second moments, i.e. $\tilde{C}^{-1} \equiv C^{-1} \equiv {}^{\lfloor\psi}L' \, {}^{\lfloor\psi}D \, {}^{\lfloor\psi}L$ and $\tilde{\nu} \equiv \nu$.
- To change point estimate of regression coefficients

$$\begin{aligned} \tilde{\hat{\theta}} &= \hat{\theta} + \rho\sqrt{\hat{r}}C^{0.5}e, \;\; f(e) = \mathcal{N}_e(0, I) \;\; \rho \in \rho^* \equiv (1,5), \;\; I \equiv \text{unit matrix} \;\; \Leftrightarrow \\ {}^{\lfloor d\psi}\tilde{L} &= {}^{\lfloor d\psi}L + \rho\sqrt{r}\, {}^{\lfloor\psi}D^{-0.5}e \end{aligned} \quad (8.52)$$

- To change point estimate of noise covariance, using normal approximation for its estimate, possibly cut to values greater than $\underline{r}$ specifying the smallest expected noise level. The level $\underline{r}$ is always strictly positive as at least numerical noise is always present. Specifically,

$$ {}^{\lfloor d}\tilde{D} = \max\left[{}^{\lfloor d}D\left(1 + \frac{\rho\sqrt{2}}{\sqrt{\nu - 4}}e\right), (\nu-2)\underline{r}\right], \rho \in \rho^* \equiv (1,5), \;\; f(e) = \mathcal{N}_e(0,1) \quad (8.53)$$

Statistics κ determining the Dirichlet pdf of component weights should be modified so that components with $\hat{\alpha}_c \approx 0$ are given a chance to be modified. We use basic properties of the Dirichlet pdf, Proposition 10.1.1. We approximate it by a normal pdf. It is also cut so that the smallest κ_c does not decrease. It gives

$$\tilde{\kappa}_c = \max\left[\kappa_c\left(1 + \frac{\rho}{\sqrt{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c}}}}e_c\right), \min_{\tilde{c}\in c^*}\kappa_{\tilde{c}}\right], \;\; \rho \in (1,5), \;\; f(e_c) = \mathcal{N}_{e_c}(0,1). \quad (8.54)$$

Remark(s) 8.4.1

1. *The obtained result has to be flattened, Section 6.4.3.*
2. *Random generating of statistics can be used as a part of various compound algorithms for constructing the prior pdf.*
3. *The optional value ρ is tuning knob of generators. Its recommended range stems from standard properties of the normal pdf. Values of ρ in a more narrow range $\rho^* \equiv (1,2)$ seem to be preferable.*

4. *The adopted normal approximations are very simplified and can surely be improved. It does not seem worth of spending energy on this problem.*

5. *A proper selection of the factor to be randomly branched decides on efficiency of the algorithm within which the branching is used. The problem is explicitly addressed in connection with branching by splitting, see Section 6.4.8, 8.4.8.*

8.4.6 Prior-posterior branching

The prior-posterior branching, see Section 6.4.6, starts at some prior pdf, performs an approximate estimation, Section 8.5, and flatten the resulting posterior pdf so that it provides a new alternative to the prior pdf used. The specialization of the general results on flattening gives directly the following iterative Bayesian learning of normal mixtures.

Algorithm 8.4.1 (Prior-posterior branching)
Initial mode

- *Select an upper bound $\mathring{n}$ on the number n of iterations and set $n = 1$.*
- *Select a sufficiently rich structure of the mixture, i.e. specify the number of components $\mathring{c}$ and the ordered lists of factors allocated to considered components. The structure of each factor ic is determined by the structure of the corresponding data vector Ψ_{ic}.*
- *Select a flat pre-prior pdf $\bar{f}(\Theta)$ in the form (6.3) with GiW pdfs describing individual factors. It means, select the prior statistics $\bar{V}_{ic}, \bar{\nu}_{ic}, \bar{\kappa}_c$ determining GiW and Dirichlet pdfs with high uncertainties, cf. Propositions 8.1.7, 10.1.1. It serves as an alternative in flattening.*
- *Select a prior pdf $f(\Theta)$ in the form (6.3) with GiW pdfs describing individual factors. It means, select the prior statistics $V_{ic}, \nu_{ic}, \kappa_c$ determining GiW and Dirichlet pdfs, cf. Propositions 8.1.7, 10.1.1.*

  Note that generally $\bar{f}(\Theta) \neq f(\Theta)$.
- *Set the first guess of the prior pdf in n-th iteration $\hat{f}_{1n}(\Theta) \equiv f(\Theta)$, i.e. set $V_{1icn;0} = V_{ic}$, $\nu_{1icn;0} = \nu_{ic}, \kappa_{1cn;0} = \kappa_c$, with $c \in c^*$ marking component, and giving $\hat{f}_{1n}(\theta_{ic}, r_{ic}) = GiW_{\theta_{ic},r_{ic}}(V_{1icn;0}, \nu_{1icn;0})$ and $\hat{f}_{1n}(\alpha) = Di_\alpha(\kappa_{1n;0})$.*
- *Compute the posterior pdf*

$$\tilde{f}_{1n}(\Theta|d(\mathring{t})) = Di_\alpha(\kappa_{1n;\mathring{t}}) \prod_{i\in i^* c\in c^*} GiW_{\theta_{ic},r_{ic}}(V_{1icn;\mathring{t}}, \nu_{1icn;\mathring{t}})$$

  using an approximate Bayesian estimation that starts at $\hat{f}_{1n}(\Theta)$, see Section 8.5.
- *Evaluate v-likelihood l_{1n} resulting from the use of $\hat{f}_{1n}(\Theta)$.*
- *Apply flattening operation to $\tilde{f}_{1n}(\Theta|d(\mathring{t}))$ according to Algorithms 6.4.2 (on Dirichlet marginal) and 6.4.3 (on the GiW pdfs describing factors). Denote the resulting pdf $\hat{f}_{2n}(\Theta)$. For*

$$\begin{aligned}
\Lambda_D &\equiv \frac{\sum_{c\in c^*}(\kappa_c - \bar{\kappa}_c)}{\sum_{\tilde{c}\in c^*}(\kappa_{1\tilde{c}n;\mathring{t}} - \bar{\kappa}_{\tilde{c}})}, \quad \Lambda_G \equiv \frac{\sum_{c\in c^*}\sum_{i\in i^*}(\nu_{ic} - \bar{\nu}_{ic})}{\sum_{\tilde{c}\in c^*}(\nu_{1i\tilde{c}n;\mathring{t}} - \bar{\nu}_{i\tilde{c}})}, \text{ it preserves the form (6.3) with} \\
\kappa_{2cn;0} &= \Lambda_D \kappa_{1cn;\mathring{t}} + (1 - \Lambda_D)\bar{\kappa}_c, \quad V_{2icn;0} = \Lambda_G V_{1icn;\mathring{t}} + (1 - \Lambda_G)\bar{V}_{ic} \\
\nu_{2icn;0} &= \Lambda_G \nu_{1icn;\mathring{t}} + (1 - \Lambda_G)\bar{\nu}_{ic}.
\end{aligned}$$

- *Compute the posterior pdf $\tilde{f}_{2n}(\Theta|d(\mathring{t})) = Di_\alpha(\kappa_{2n;\mathring{t}}) \prod_{i\in i^*} GiW_{\theta_{ic},r_{ic}}\left(V_{2icn;\mathring{t}}, \nu_{2icn;\mathring{t}}\right)$ using an approximate Bayesian estimation that starts at $\hat{f}_{2n}(\Theta)$, see Section 8.5.*
- *Evaluate the v-likelihood l_{2n} resulting from the use of $\hat{f}_{2n}(\Theta)$.*

- *Set* $\bar{l}_n = \max(l_{1n}, l_{2n})$.

Iterative mode

1. *Apply geometric branching to the pair* $\tilde{f}_{1n}(\Theta|d(\mathring{t}))$, $\tilde{f}_{2n}(\Theta|d(\mathring{t}))$ *with v-likelihood* l_{1n}, l_{2n}, *respectively. For* $\lambda \equiv \frac{l_{1n}}{l_{1n}+l_{2n}}$, *it gives* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$ *of the form (6.3) with the GiW description of factors. It is determined by the statistics*

$$\begin{aligned} \kappa_{3cn;\mathring{t}} &= \lambda\kappa_{1cn;\mathring{t}} + (1-\lambda)\kappa_{2cn;\mathring{t}} \\ V_{3icn;\mathring{t}} &= \lambda V_{1icn;\mathring{t}} + (1-\lambda)V_{2icn;\mathring{t}},\ \nu_{3icn;\mathring{t}} = \lambda\nu_{1icn;\mathring{t}} + (1-\lambda)\nu_{2icn;\mathring{t}}. \end{aligned}$$

2. *Apply the flattening operation to* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$ *according to Algorithms 6.4.2 (on the Dirichlet marginal) and 6.4.3 (on the GiW pdfs describing factors). Denote the resulting pdf* $\hat{f}_{3n}(\Theta)$. *For*

$$\Lambda_D \equiv \frac{\sum_{c\in c^*}(\kappa_{3cn} - \bar{\kappa}_c)}{\sum_{\tilde{c}\in c^*}(\kappa_{3\tilde{c}n;\mathring{t}} - \bar{\kappa}_{\tilde{c}})},\quad \Lambda_G \equiv \frac{\sum_{c\in c^*}\sum_{i\in i^*}(\nu_{3ic} - \bar{\nu}_{ic})}{\sum_{i\in i^*}\sum_{\tilde{c}\in c^*}(\nu_{3i\tilde{c}n;\mathring{t}} - \bar{\nu}_{i\tilde{c}})}$$

it preserves the GiW version of (6.3) with

$$\begin{aligned} \kappa_{3cn;0} &= \Lambda_D\kappa_{3cn;\mathring{t}} + (1-\Lambda_D)\bar{\kappa}_c \\ V_{3icn;0} &= \Lambda_G V_{3icn;\mathring{t}} + (1-\Lambda_G)\bar{V}_{ic},\ \nu_{3icn;0} = \Lambda_G\nu_{3icn;\mathring{t}} + (1-\Lambda_G)\bar{\nu}_{ic}. \end{aligned}$$

3. *Evaluate the v-likelihood* l_{3n} *resulting from the use of* $\hat{f}_{3n}(\Theta)$ *as the prior pdf in an approximate estimation, Section 8.5. The approximation prepared for the fixed advisory system, see Section 6.1.2 may be used.*

4. *Choose among* $\tilde{f}_{in}(\Theta|d(\mathring{t}))$, $i \in \{1,2,3\}$ *the pair with the highest v-likelihood values and call them* $\tilde{f}_{1(n+1)}(\Theta|d(\mathring{t}))$, $\tilde{f}_{2(n+1)}(\Theta|d(\mathring{t}))$ *with the v-likelihood values* $l_{1(n+1)}$, $l_{2(n+1)}$.

5. *Go to the beginning of* Iterative mode *with* $n = n+1$ *if* $\bar{l}_{n+1} \equiv \max(l_{1(n+1)}, l_{2(n+1)}) > \bar{l}_n$ *or if* $\bar{l}_{n+1}, \bar{l}_n$ *are the same according to Proposition 6.1.2 and* $n < \mathring{n}$.

6. *Stop and select among* $\hat{f}_{in}(\Theta)$, $i = 1,2$ *that leading to the higher value of* l_{in} *and take it as the prior pdf constructed.*

Remark(s) 8.4.2

1. *No improvement can be expected when the value of* λ *at the beginning of the iterative mode in is around 0.5. This observation may serve as an additional stopping rule.*

2. *The structure of the estimated mixture does not change during iterations. Thus, it has to be sufficiently reflected even in the prior pdf* $f(\Theta)$.

8.4.7 Branching by forgetting

Branching by forgetting, see Section 6.4.7, makes parallel recursive estimations with no and a fixed forgetting with the forgetting factor smaller than 1. The alternative pdfs are compared according to their v-likelihood. At the moment, when one of them is a sure winner they are bounded into a single pdf and whole process is repeated. Here, the algorithm is specialized to normal mixtures.

Algorithm 8.4.2 (On line branching with forgetting for normal mixtures)
Initial mode

- *Select a sufficiently rich structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to the considered components. The structure of each factor ic is determined by the structure of the corresponding data vector* Ψ_{ic}.

- *Set a constant* $\rho \approx 3-5$ *defining the significant difference of v-log-likelihood values.*

- *Set the record counter* $t = 0$ *and select the statistics determining the prior pdf*

$$\hat{f}(\theta_{ic}, r_{ic}) = GiW_{\theta_{ic}, r_{ic}}(V_{ic;0}, \nu_{ic;0}),\ i \in i^*,\ c \in c^*$$

for factors as well as the statistics κ_0 *determining the pdf* $\hat{f}(\alpha) = Di_{\alpha}(\kappa_0)$ *of the component weights* α.

- *Choose a fixed relatively small forgetting factor* $\lambda < 1$, *say* $\lambda = 0.6$.
- *Select a fixed pre-prior alternative pdf used in the stabilized forgetting, see Proposition 3.1.1. The alternative is usually taken as a flat pre-prior pdf in the GiW version of (6.3). It is given by small* ${}^{\lfloor A}\kappa$, ${}^{\lfloor A}\nu_{ic}$ *and* ${}^{\lfloor A}V_{ic} = \varepsilon I$. *Here,* ε *is with a small positive scalar and* I *is a unit matrix of the appropriate dimension.*

Data processing mode

1. *Set* $\hat{f}_1(\Theta|d(t)) = \hat{f}_\lambda(\Theta|d(t)) = \hat{f}(\Theta|d(t))$.
2. *Initialize the v-log-likelihood values assigned to the considered alternatives* $l_{1;t} = 0$, $l_{\lambda;t} = 0$.
3. *Collect new data* d_{t+1} *and construct the data vector* Ψ_{t+1}.
4. *Update* $\hat{f}_1(\Theta|d(t))$ *to* $\hat{f}_1(\Theta|d(t+1))$ *using an approximate estimation, Section 8.5, with the forgetting factor 1.*
5. *Re-compute the v-log-likelihood* $l_{1;t}$ *to* $l_{1;t+1}$ *by adding the logarithm of the mixture prediction* $\ln(f(d(t+1)|d(t)))$ *obtained for the "prior" pdf* $\hat{f}_1(\Theta|d(t))$.
6. *Update* $\hat{f}_\lambda(\Theta|d(t))$ *to* $\hat{f}_\lambda(\Theta|d(t+1))$ *using approximate estimation with the forgetting factor* λ *and the chosen alternative. Thus, after obtaining statistics* $\kappa_{\lambda 1;t+1}$, $V_{ic\lambda 1;t+1}$, $\nu_{ic\lambda 1;t+1}$ *through updating of* $\kappa_{\lambda;t}$, $V_{ic\lambda;t}$, $\nu_{ic\lambda;t}$ *with forgetting 1, forget*

$$\begin{aligned}\kappa_{\lambda;t+1} &= \lambda\kappa_{\lambda 1;t+1} + (1-\lambda)\,{}^{\lfloor A}\kappa,\quad V_{ic\lambda;t+1} = \lambda V_{ic\lambda 1;t+1} + (1-\lambda)\,{}^{\lfloor A}V_{ic}\\ \nu_{ic\lambda;t+1} &= \lambda\nu_{ic\lambda 1;t+1} + (1-\lambda)\,{}^{\lfloor A}\nu_{ic}.\end{aligned}$$

7. *Add logarithm of the mixture prediction* $\ln(f(d(t+1)|d(t)))$, *obtained for the "prior" pdf* $\hat{f}_\lambda(\Theta|d(t))$, *to the v-log-likelihood* $l_{\lambda;t}$ *get* $l_{\lambda;t+1}$.
8. *Go to the step 3 with* $t = t+1$ *if* $|l_{1;t+1} - l_{\lambda;t+1}| < \rho$.
9. *Set* $\hat{f}(\Theta|d(t+1)) = \hat{f}_1(\Theta|d(t+1))$ *if* $l_{1;t+1} > l_{\lambda;t}$. *Otherwise set* $\hat{f}(\Theta|d(t+1)) = \hat{f}_\lambda(\Theta|d(t+1))$.

   It consists of the assignment of the appropriate sufficient statistics.
10. *Increase* $t = t+1$ *and go to the beginning of* Data processing mode *if* $t \leq \mathring{t}$ *otherwise stop and take* $\hat{f}_1(\Theta|d(\mathring{t}))$ *as the final estimate.*

Remark(s) 8.4.3

1. *Speeding up of the learning is the main expectation connected with this algorithm. The model with no forgetting is expected to be the winner in a long run. Current experience indicates that the estimator with forgetting wins only for initial several tens of data records.*

   The estimation with forgetting can be switched off when the estimation without forgetting is better for a majority of time.
2. *The technique can be directly combined with prior-posterior branching. Use of the stabilized forgetting is vital in this case. Without it, flattening rates computed recommended in Section 6.4.3 go often out of the admissible range (0,1].*

3. *Use of the general version of stabilized forgetting in Algorithm 8.4.2 is computationally expensive. The option that cares only about estimates of the noise variance and component weights seems to be acceptable to the given purpose. It has the form*

$$^{\lfloor A}V_{ic} = \begin{bmatrix} ^{\lfloor Ad}V_{ic} & 0 \\ 0 & 0 \end{bmatrix}, \ ^{\lfloor A}\nu_{ic} > 0, \ \kappa_c > 0, i \in i^*, \ c \in c^*.$$

The scalars $^{\lfloor Ad}V_{ic}$, $^{\lfloor A}\nu_{ic}$, κ_c *are chosen so that the estimate of the noise variance and component weight as well as their precision are fixed at prior values.*

4. *It is necessary to get experience with choice of the fixed forgetting factor* λ. *At present, the option* $\lambda \approx 0.6$ *seems to be satisfactory.*

5. *It would be desirable formulate the selection of the better model as sequential decision making. It should prevent the situations when the decision is made using too much uncertain probabilities of the compared hypotheses.*

8.4.8 Branching by factor splitting

Branching by factor splitting splits factors suspicious for hiding more modes and makes a new learning attempt. The general solution is described in Section 6.4.8. It covers important cases when we have no clue on structure of the mixture. Basic steps described by Algorithm 6.4.7 apply to the normal case without change. Normality adds nothing new to the selection of the very initial mixture and to the construction of a new mixture. Structure estimation aspects are specialized in Section 8.6. Thus, it remains to specialize the selection of factors to be split and the splitting algorithms. These aspects are discussed in this Section.

Optimization-based splitting of factors

The preferable hierarchical selection of split factors, Section 6.4.8, is often too costly to be directly used. Thus, we have orient ourselves on optimization-based splitting. Significant symmetry of GiW pdfs in regression coefficients θ calls for an explicit shift of factors to be split, for the application of Proposition 6.4.8. It enforces a non-zero shift of the optional function $g(\Theta)$ that determines the resulting functional form of the modified pdf as $f(\Theta)\exp[-\langle \mu, g(\Theta)\rangle]$, (6.68). It is easy to see that the result remains within GiW class if

$$g'(\Theta) \equiv g'(\theta, r) = \left[\ln(r), \frac{1}{r}, \frac{\theta'}{r}\right] M, \quad \text{where} \tag{8.55}$$

M is an arbitrary fixed matrix with $(\mathring{\Psi} + 1)$ rows. The application of Proposition 6.4.8 to this most general form does not provide a closed form solution. The problem can be avoided as follows.

First, we observe that the shifted pdf has to have $C^{-1} = {}^{\lfloor\psi}L' \, {}^{\lfloor\psi}D \, {}^{\lfloor\psi}L$ identical with its pattern as no shift of the quadratic term in θ is done. Second, we recall that splitting is done with hope to improve predictive abilities of the factor. The combination of these facts implies, that the noise variance r of the shifted factors is expected to be smaller than it appears currently. The variance of a convex mixture of a pair of normal pdfs is greater than a convex combination of individual ones. We suppose that the estimated variance resulted from such a combination and we have no reasons to assume their asymmetric role. Thus, we expect that for the shifted factors it will be smaller than one half of the original value. Use of this bound gives the rule how to replace ${}^{\lfloor d}D$ with ${}^{\lfloor d}\hat{D}$. We set

$${}^{\lfloor d}\hat{D} = s \, {}^{\lfloor d}D \ \text{ with the optional } s \in (0, 5). \tag{8.56}$$

Consequently, we can deal with a simpler, explicitly solvable case, that cares about shift of the regression coefficients only. The replacement (8.56) can be performed either before or after it.

Proposition 8.4.4 (Shifted GiW factor) *Let us consider pdf* $f = GiW_{\theta,r}(L, D, \nu)$. *Let the function determining the shift be* $g(\Theta) \equiv \theta / r$. *Let the pdf* $\hat{f}$ *minimize the KL distance* $\mathcal{D}\left(\hat{f}||f\right)$ *and have the* *prescribed norm of the shift in the expectation of* $g(\Theta)$

$$\left\langle \int g(\Theta)(\hat{f}(\Theta) - f(\Theta))\, d\Theta, \int g(\Theta)(\hat{f}(\Theta) - f(\Theta))\, d\Theta \right\rangle = \omega,\ \omega > 0. \tag{8.57}$$

Then, this pdf has the form

$$\begin{aligned} \hat{f} &= GiW_{\theta,r}(\hat{L}, \hat{D}, \hat{\nu}) \quad with \\ \hat{L} &\equiv \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L + \rho\, {}^{\lfloor\psi}Lx & L_\psi \end{bmatrix} \Leftrightarrow \hat{\hat{\theta}} = \hat{\theta} + \rho x, \quad {}^{\lfloor\psi}\hat{D} = {}^{\lfloor\psi}D,\ \hat{\nu} = \nu. \end{aligned} \tag{8.58}$$

The unit $\mathring{\psi}$*-vector* x *is eigenvector of the covariance matrix of parameters* $\frac{{}^{\lfloor d}D}{\nu}\, {}^{\lfloor\psi}L^{-1}\, {}^{\lfloor\psi}D^{-1} \left({}^{\lfloor\psi}L'\right)^{-1}$ *corresponding to the largest eigenvalue* μ. *The scalar* $\rho \neq 0$ *is a tuning knob. Its choice replaces the choice of* ω. *The value* $-\rho$ *determines the solution of the problem if the value* ρ *does.*

Proof: Proposition 6.4.8 implies functional form (6.68) of $\hat{f}$. The choice of $g(\Theta)$ guarantees that the (6.68) stay in GiW class with no changes in ν, ${}^{\lfloor\psi}L$ and D. Thus, ${}^{\lfloor d\psi}\hat{L}$ is the only optional quantity. The part of the KL distance $\mathcal{D}\left(\hat{f}||f\right)$ that depends on these quantities is proportional to

$$\left(\hat{\hat{\theta}} - \hat{\theta}\right)' {}^{\lfloor d}D^{-1}C^{-1} \left(\hat{\hat{\theta}} - \hat{\theta}\right), \quad \text{where } \hat{\hat{\theta}} \text{ are LS estimates of } \theta \text{ assigned to } \hat{f}.$$

A use of Proposition 8.1.7 implies that the restriction (8.57) defines the non-zero quadratic norm of the shift in least squares estimates of regression coefficients $\left(\hat{\hat{\theta}} - \hat{\theta}\right)' \left(\hat{\hat{\theta}} - \hat{\theta}\right) = \hat{r}^2(\omega - 1)$. We have exploited the fact that, with unchanged ν and ${}^{\lfloor d}D$, the expectation of r^{-1} is the same for both pdfs considered.

Obviously, $\hat{\hat{\theta}} - \hat{\theta} = \rho x$ is minimizing argument searched for, if x is the unit eigenvector corresponding to the smallest eigenvalue of ${}^{\lfloor\psi}L'\, {}^{\lfloor\psi}D\, {}^{\lfloor\psi}L$. It is the largest eigenvalue of its inversion. The normalizing factor ρ has to be chosen so that the considered restriction is fulfilled. The square ρ^2 enters the restriction, thus both positive and negative solution are acceptable. The claimed form of the constructed pdf is obtained when taking into account the common value of ${}^{\lfloor\psi}L$ and definition of least-squares estimates $\hat{\theta}$ (8.18) of θ. □

Remark(s) 8.4.4

1. *In an over-parameterized case, the maximizing eigenvector points to "superfluous" directions. Thus, structure estimation and corresponding reduction have to be performed before splitting the factor, cf. item 1 in Remarks 6.4.10.*
2. *The following simple iterative algorithm for determining the maximum eigenvalue and corresponding eigenvector* x *is used*
$$z_n \equiv Cx_{n-1},\ x_n = \frac{z_n}{\sqrt{\mu_n}},\ \mu_n \equiv z_n' z_n,\ x_0 \neq 0. \tag{8.59}$$
Its properties can be seen in the space rotated so that C *is a diagonal matrix. In this space,* x_n *concentrates its unit mass at the entry corresponding to the largest eigenvalue and thus it selects the searched eigenvector in the original space. The scalar* μ_n *converges to the corresponding eigenvalue. This simple case happens if the searched eigenvalue is strictly the largest one. If there is more such eigenvalues then the algorithm concentrates* x_n *to the subspace of the corresponding eigenvectors. This is, however, exactly what we need for the considered purpose.*
3. *The above iterations have to be performed at appropriate numerical level as the considered kernel* C *may be very low as it is inversely proportional to* $\mathring{t}$.
4. *Experiments indicate sensitivity of splitting quality on the optional factor* s *in (8.56). For normalized data, values* $s \approx 0.2$ *seem to be reasonable but sometimes much better results can be gained with other options. It calls for experimenting with this tuning knob. Related computational demands make this feature highly undesirable.*

We have to make a carefully selected choice of the function $g(\Theta)$ in order to stay within the GiW class after shifting. The following proposition offers a variant that requires the class preservation explicitly. It is based on the alternative formulation of the splitting problem, cf. Proposition 6.4.9.

Proposition 8.4.5 (Shift of a GiW factor to a GiW factor) *Let us consider the set of GiW pdfs $f^* \equiv \{f = GiW_{\theta,r}(L, D, \nu)\}$ and a fixed pdf $f = GiW_{\theta,r}(L, D, \nu)$ in it.*

Let scalar $\beta \in (0, 1)$ be chosen and we search for $\hat{f}$ that

1. *minimizes the KL distance $\mathcal{D}\left(\hat{f}||f\right)$,*
2. *has the prescribed shift in expectations $\hat{\theta}$, $\hat{r}$ (for f) and $\hat{\hat{\theta}}$, $\hat{\hat{r}}$ (for $\hat{f}$) of θ, r^{-1}*
$$f\left(\hat{\hat{\Theta}}\right) = \beta f\left(\hat{\Theta}\right) \quad \text{with} \quad \hat{\hat{\Theta}} = \left[\hat{\hat{\theta}}, \hat{\hat{r}}\right] \equiv \left[{}^{\lfloor\psi}\hat{L}{-1}\,{}^{\lfloor d\psi}\hat{L}, \frac{{}^{\lfloor d}\hat{D}}{\hat{\nu}}\right], \quad \hat{\Theta} = \left[\hat{\theta}, \hat{r}\right] \equiv \left[{}^{\lfloor\psi}L^{-1}\,{}^{\lfloor d\psi}L, \frac{{}^{\lfloor d}D}{\nu}\right]$$
computed for $\hat{f}(\Theta) \equiv GiW_{\theta,r}\left(\hat{L}, \hat{D}, \hat{\nu}\right)$ and $f(\Theta) \equiv GiW_{\theta,r}(L, D, \nu)$.
3. *has the prescribed ratio of peaks*
$$\hat{f}\left(\hat{\hat{\Theta}}\right) = \beta^{-1} f\left(\hat{\Theta}\right). \tag{8.60}$$
4. *has the largest Euclidean norm of the shift in $\hat{\theta}$ both as consequence of the optimization and of the appropriate choice of the factor β.*

Then, the statistics determining $\hat{f}$ are given by the formulas

$$\hat{L} \equiv \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L + s\rho\,{}^{\lfloor\psi}Lx & {}^{\lfloor\psi}L \end{bmatrix} \Leftrightarrow \hat{\hat{\theta}} = \hat{\theta} + s\rho x, \ s \in \{-1, 1\}, \tag{8.61}$$

$$\begin{aligned} x &= \text{unit-length eigenvector of } C = ({}^{\lfloor\psi}L'\,{}^{\lfloor\psi}D\,{}^{\lfloor\psi}L)^{-1} \text{ corresponding to the maximum eigenvalue } \eta, \\ {}^{\lfloor\psi}\hat{D} &= {}^{\lfloor\psi}D, \ \hat{\nu} = \nu, \ {}^{\lfloor d}\hat{D} = \hat{\hat{r}}\hat{\nu} \text{ with} \\ \hat{\hat{r}} &= \hat{r}/z. \end{aligned} \tag{8.62}$$

The scalars $z > 1, \rho > 0$ are determined by expressions

$$\begin{aligned} z &\equiv \beta^{-\frac{2}{\mathring{\Psi}+2}} \\ \rho^2 &= \gamma\left\{-1 + \frac{1 - 2/\nu\ln(\beta) + \left(1 + \frac{\mathring{\Psi}+2}{\nu}\right)\ln(z)}{z}\right\}, \quad \text{where} \\ \gamma &\equiv \nu\eta\hat{r} \text{ and } \beta = \exp\left(-0.5\frac{(\mathring{\Psi}+2)^2}{2(\mathring{\Psi}+2)+\nu}\right). \end{aligned} \tag{8.63}$$

Proof: The optimization is split in several steps using repeatedly the specific form of the GiW pdf and Propositions 8.1.7, 8.1.1.

1. The first restriction $f\left(\hat{\hat{\Theta}}\right) = \beta f\left(\hat{\Theta}\right)$, enforcing explicit shift of $\hat{\hat{\Theta}}$, has for the considered pdf $f(\Theta) = GiW(\hat{\theta}, C, \hat{r}, \nu)$ the form
$$\hat{r}^{-0.5\left(\nu+\mathring{\Psi}+2\right)}\exp[-0.5\nu] = \beta^{-1}\hat{\hat{r}}^{-0.5(\nu+\mathring{\Psi}+2)}\exp\left[-0.5\nu\frac{\left(\hat{\hat{\theta}}-\hat{\theta}\right)'(\nu C)^{-1}\left(\hat{\hat{\theta}}-\hat{\theta}\right)+\hat{r}}{\hat{\hat{r}}}\right].$$
Let us introduce auxiliary variables
$$\begin{aligned} z &\equiv \frac{\hat{r}}{\hat{\hat{r}}} > 0, \quad \rho x \equiv \hat{\hat{\theta}} - \hat{\theta}, \ x'x = 1, \\ \omega &\equiv \rho^2 x'(\hat{r}\nu C)^{-1}x > 0, \ \delta \equiv \frac{\mathring{\Psi}+2}{\nu} > 0, \quad q \equiv 1 - \frac{2}{\nu}\ln(\beta) > 1. \end{aligned}$$

Then, the above restriction gets the form $(1+\delta)\ln(z)+q = z(\omega+1)$. Note that this restriction has non-empty set of admissible solutions as the pair $z=1$ and $\omega = -2/\nu\ln(\beta)$ solves it.

2. The KL distance expressed in terms of the introduced variables has the form

$$\begin{aligned}\mathcal{D}\left(\hat{f}||f\right) &= \ln\left(\frac{\Gamma(0.5\nu)}{\Gamma(0.5\hat{\nu})}\right) - 0.5\ln\left(\left|\hat{C}C^{-1}\right|\right) - 0.5\nu\ln(\nu) - 0.5\nu\ln(z) + 0.5\nu\ln(\hat{\nu}) + \\ &+ 0.5(\hat{\nu}-\nu)\frac{\partial}{\partial(0.5\hat{\nu})}\ln(\Gamma(0.5\hat{\nu})) - 0.5\mathring{\psi} - 0.5\hat{\nu} + 0.5\mathrm{tr}\left[\hat{C}C^{-1}\right] + 0.5\nu z(\omega+1).\end{aligned}$$

3. The part of the KL distance to be optimized with respect to $0.5\hat{\nu}$ has the form

$$-\ln\left(\Gamma(0.5\hat{\nu})\right) + 0.5\nu\ln(0.5\hat{\nu}) + 0.5(\hat{\nu}-\nu)\frac{\partial}{\partial(0.5\hat{\nu})}\ln(\Gamma(0.5\hat{\nu})) - 0.5\hat{\nu}.$$

Setting its derivative with respect to $\hat{\nu}$ to zero, we get

$$1 = \nu/\hat{\nu} + 0.5(\hat{\nu}-\nu)\frac{\partial^2}{\partial(0.5\hat{\nu})^2}\ln\left(\Gamma(0.5\hat{\nu})\right).$$

Using asymptotic formula [81] (page 250) for derivatives of $\ln(\Gamma(\cdot))$,

$$\begin{aligned}&\frac{\partial^2}{\partial(0.5\hat{\nu})^2}\ln\left(\Gamma(0.5\hat{\nu})\right) \approx (0.5\hat{\nu})^{-1} + 0.5(0.5\hat{\nu})^{-2}, \text{ we get, for } y = \nu/\hat{\nu}, \\ 1 &= y + (1-y)(1+y/\nu) = 1 + (1-y)y/\nu \ \Rightarrow\ y = 1 \ \Rightarrow \hat{\nu} = \nu.\end{aligned}$$

4. The second restriction $\hat{f}\left(\hat{\Theta}\right) = \beta^{-1}f(\hat{\Theta})$ reads (note that normalizing factors differ)

$$|\hat{C}|^{-0.5}\hat{r}^{-0.5\left(\mathring{\Psi}+2\right)} = \beta^{-1}|C|^{-0.5}r^{-0.5\left(\mathring{\Psi}+2\right)}.$$

This condition provides

$$z = \left[\beta^{-2}\left|\hat{C}C^{-1}\right|\right]^{\left(\mathring{\Psi}+2\right)^{-1}}.$$

Note that $\left|\hat{C}C^{-1}\right| \geq 1$ implies that $z > 1$.

5. By minimizing the KL distance unconditionally with respect to $\hat{C}$, we get $\hat{C} = C$, so that z is uniquely determined

$$z = \beta^{-\frac{2}{\mathring{\Psi}+2}}.$$

It will be optimum value if the remaining free parameters stay within the admissible range.

6. Given z, the value of ω is uniquely determined. We require the largest shift. Thus, x must be eigenvector of C corresponding to the largest eigenvalue η of C. Then, $\omega = \rho^2/\gamma$ with $\gamma = \eta\nu\hat{r}$ and the first restriction implies

$$\rho^2 = \gamma\left[-1 + \frac{(1+\delta)\ln(z)+q}{z}\right].$$

7. It remains to maximize the right-hand side of the above equation and check whether it is positive for the maximizing $\beta \in (0,1)$. We can maximize the expression in question with respect to z as there is its one-to-one mapping to β.

The necessary condition for extreme and form of z give

$$1 - \frac{1-\frac{2}{\nu}\ln(\beta)}{1+\delta} = \ln(z) = -\frac{2}{\mathring{\Psi}+2}\ln(\beta) \Rightarrow,\ \ \beta = \exp\left(-0.5\frac{\left(\mathring{\Psi}+2\right)^2}{2\left(\mathring{\Psi}+2\right)+\nu}\right).$$

□

8.4.9 Hierarchical and optimization based splitting

As said above, the preferable hierarchical selection of the split factors, Section 6.4.8 is too costly to be directly used. Here, we try to exploit the specific form of the split factor and to combine the hierarchical and optimization based splitting.

Hierarchical splitting exploits the fact that during approximate estimation the i-th factor within the c-th component is assigned a weight $w_{ic;t} \in [0,1]$. It expresses the degree to which the current data item $d_{ic;t}$ can be though as generated by this factor. The correspondingly modified parameterized factor

$$f_w(d_{ic;t}|d_{i+1\_\mathring{d};t}, d(t-1)) \equiv f_w(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) \propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic})]^{w_{ic;t}}$$

is used in the "ordinary" Bayes rule for updating of parameter estimates of this factor. Thus, we can inspect parameter estimation of this factor without considering the other ones. Working with a fixed factor allows us so to drop temporarily the subscripts w, i, c. We can formulate hypotheses H_0, H_1 that the objective pdf is two and one component mixture, respectively,

$$\begin{aligned} {}^{\llcorner o}f(d_t|\psi_t, \Theta_0, H_0) &= \beta \mathcal{N}_{d_t}(\theta_1'\psi_t, r_1) + (1-\beta)\mathcal{N}_{d_t}(\theta_2'\psi_t, r_2),\ \beta \in (0,1), \\ {}^{\llcorner o}f(d_t|\psi_t, \Theta, H_1) &= \mathcal{N}_{d_t}(\theta'\psi_t, r). \end{aligned} \tag{8.64}$$

H_0 corresponds to the hypothesis that the factor should be split into the two-component normal mixture characterized by the parameter $\Theta_0 \equiv (\beta, \theta_1, r_1, \theta_2, r_2)$. H_1 takes the estimated factor with parameters $\Theta = (\theta, r)$ as the correct one. We assume that structures of all regression coefficient coincide, cf. Requirement 6.4.1. Moreover, we assume that the pdf $f(\psi)$ brings no information on parameters and hypotheses involved, i.e. natural conditions of decision making (2.36) apply

$$f(d_t, \psi|\Theta_0, H_0) = f(d_t|\psi, \Theta_0, H_0) f(\psi) \ \text{ and } \ f(d_t, \psi|\Theta, H_1) = f(d_t|\psi, \Theta, H_1) f(\psi).$$

According to Proposition 2.5.3, the posterior pdf $f(\Theta|d(t))$ concentrates on minimizers of the entropy rate (2.48). We assume that it converges to its expected value. Thus, asymptotically, the evaluated posterior pdf $f(\Theta|d(t))$ estimating the normal factor $\mathcal{N}_d(\theta'\psi, r)$ concentrates on

$$\lim_{\mathring{t}\to\infty} \text{supp}\left[f(\Theta|d(\mathring{t}))\right] = \text{Arg}\min_{\theta,r} \left\{ 0.5\ln(r) + 0.5\, {}^{\llcorner o}\mathcal{E}\left[(d-\underline{\theta}'\psi)^2/r\right]\right\} \equiv \text{Arg}\min_{\theta,r} \mathcal{H}_\infty\left({}^{\llcorner o}f||\theta, r\right). \tag{8.65}$$

The expectation ${}^{\llcorner o}\mathcal{E}$ is expectation assigned to the objective pdf ${}^{\llcorner o}f(d,\psi)$. Note that a constant term non-influencing the minimization is drop.

Under the hypothesis H_0 on the objective pdf (8.64), we have ${}^{\llcorner o}\mathcal{E}[\cdot] = \mathcal{E}[\cdot|\Theta_0, H_0]$ we get, for $\mathcal{H} \equiv 2\mathcal{H}_\infty\left({}^{\llcorner o}f||\Theta \equiv [\theta, r]\right)$,

$$\begin{aligned} \mathcal{H} &= \ln(r) + r^{-1}\mathcal{E}[(d-\theta'\psi)^2|H_0, \Theta_0] = \\ &= \ln(r) + \beta r^{-1}\mathcal{E}[(d-\theta'\psi)^2|\theta_1, r_1] + (1-\beta) r^{-1}\mathcal{E}[(d-\theta'\psi)^2|\theta_2, r_2] = \\ &= \ln(r) + \beta\frac{r_1}{r} + (1-\beta)\frac{r_2}{r} + r^{-1}\mathcal{E}\left\{\beta[(\theta_1-\theta)'\psi]^2 + (1-\beta)[(\theta_2-\theta)'\psi]^2\right\}. \end{aligned}$$

Let us denote $C^{-1} = \mathcal{E}[\psi\psi']$. Then, points r, θ in $\text{supp}\left[f(\Theta|d(\mathring{t}))\right]$ minimizing $2\mathcal{H}$ are

$$\begin{aligned} \theta &= \beta\theta_1 + (1-\beta)\theta_2 \\ r &= \beta r_1 + (1-\beta) r_2 + \beta(1-\beta)(\theta_1-\theta_2)' C^{-1}(\theta_1-\theta_2). \end{aligned} \tag{8.66}$$

Let us consider such a point r, θ in $\text{supp}\left[f(\Theta|d(\mathring{t}))\right]$ and introduce the variables $e_t = d_t - \theta'\psi_t = \beta(d_t - \theta_1'\psi_t) + (1-\beta)(d_t - \theta_2')\psi_t$. They are generated by the <u>static</u> mixture

$$f(e_t|d(t-1)) = \beta\mathcal{N}_{e_t}(0, r_1) + (1-\beta)\mathcal{N}_{e_t}(0, r_2).$$

The best point estimate of the unknown e_t is the *filtering error*

$$\hat{e}_t \equiv \mathcal{E}[e_t|d(t)] \equiv d_t - \hat{\theta}_t'\psi_t,\ \hat{\theta}_t \equiv \mathcal{E}[\theta|d(t)]. \tag{8.67}$$

Asymptotically, it is expected to behave as e_t. Thus, we can model filtering error $\hat{e}_t$ by a similar static mixture as e_t. In order to cope with the adopted approximations, we include constant offsets μ_1, μ_2 into the two-component static mixture describing the filtering error

$$f(\hat{e}_t|d(t-1), r_1, r_2, \mu_1, \mu_2, \beta) = \beta \mathcal{N}_{\hat{e}_t}(\mu_1, r_1) + (1-\beta)\mathcal{N}_{\hat{e}_t}(\mu_2, r_2). \tag{8.68}$$

It is estimated jointly with the factor in question. The estimation of (8.68) gives us the estimates $\hat{\beta} \equiv \mathcal{E}[\beta|d(\mathring{t})]$, $\hat{r}_1 \equiv \mathcal{E}[r_1|d(\mathring{t})]$, $\hat{r}_2 \equiv \mathcal{E}[r_2|d(\mathring{t})]$ of β, r_1, r_2 and v-likelihood $f(\hat{e}(\mathring{t}))$. The relationships (8.66) determine approximately the same relationships for expectations

$$\begin{aligned} \hat{\theta} &= \hat{\beta}\hat{\theta}_1 + (1-\hat{\beta})\hat{\theta}_2 \\ \hat{r} &= \hat{\beta}\hat{r}_1 + (1-\hat{\beta})\hat{r}_2 + \hat{\beta}(1-\hat{\beta})(\hat{\theta}_1 - \hat{\theta}_2)'C^{-1}(\hat{\theta}_1 - \hat{\theta}_2). \end{aligned} \tag{8.69}$$

In it, conditional independence of involved parameters implied by the product form of the approximately evaluated posterior pdfs (6.3) in the special case of two-component mixture is used. Moreover, covariance of $\beta(1-\beta)$ is neglected.

The second equality in (8.69) determines the value of the quadratic form

$$q'C^{-1}q = \frac{\hat{r}}{\hat{\beta}(1-\hat{\beta})} - \frac{\hat{r}_1}{1-\hat{\beta}} - \frac{\hat{r}_2}{\hat{\beta}} \equiv \rho, \; q \equiv \hat{\theta}_1 - \hat{\theta}_2. \tag{8.70}$$

Recalling that $C^{-1} = \mathcal{E}[\psi\psi']$, we see, (8.19), that we have at disposal its sampled version while estimating θ, r.

We make the splitting unique by selecting the most distant $\hat{\theta}_1$, $\hat{\theta}_2$, by selecting q that fulfills (8.70) with the largest norm $q'q$. The vector $\hat{q} = \sqrt{\zeta\rho}q_0$, where q_0 is the unit eigenvector of C corresponding to the largest eigenvalue ζ of C can be shown to be vector of this type. Having it, we can use the first equality in (8.69) and specify

$$\hat{\theta}_1 = \hat{\theta} + (1-\hat{\beta})\hat{q}, \; \hat{\theta}_2 = \hat{\theta} - \hat{\beta}\hat{q}. \tag{8.71}$$

Data attributed to originally to single factor should be distributed between newly created factors. It leads to the option

$$C_1 = \hat{\beta}^{-1}C, \; C_2 = (1-\hat{\beta})^{-1}C \text{ and } \nu_1 = \hat{\beta}\nu, \;\; \nu_2 = (1-\hat{\beta})\nu. \tag{8.72}$$

It concludes the specification of the splitting step.

In order to judge whether splitting is necessary, it is sufficient evaluate the v-likelihood under the hypothesis H_1, i.e. to estimate during estimation also one-component counterpart of model (8.64), i.e. $\mathcal{N}_{\hat{e}_t}(\mu, r)$. The test is described by the following algorithm that also generates as a byproduct quantities needed for performing the split. It is prepared for the normalized data that lie predominantly in the range $[-1, 1]$.

Algorithm 8.4.3 (Test on the need to split factor)
Initial mode

- *Select the decision level* $\gamma \in (0,1)$, $\gamma \to 1$.
- *Attach to each factor* ic, $i \in i^*$, $c \in c^*$, *the two-component static mixture given by the sufficient statistics in LS form, cf. (8.17),*

$$\begin{aligned} \hat{\theta}_{1ic;0} &= -\hat{\theta}_{2ic;0} \in (1,5), \;\; \textit{(significant shifts from the expected value 0)}, \\ \hat{r}_{1ic;0} &= \hat{r}_{2ic;0} \in (1e-6, 1e-4), \;\; \textit{(the noise level in the range (0.1,1)\%) of the data range}, \\ C_{1ic;0} &= C_{2ic;0} = 1/\hat{r}_{1ic;0}, \;\; \textit{(covariance of parameter estimates expected to be order} \approx 1\textit{)}, \\ \nu_{1ic;0} &= \nu_{2ic;0} = 2(2 + 1/\varepsilon^2), \; \varepsilon \in (0.1, 1), \\ & \qquad \left(\textit{the ratio } \frac{\sqrt{\mathrm{var}(r)}}{\hat{r}} \textit{ expected in (10,100)\%, cf. (8.23)}\right), \\ \kappa_{1ic;0} &= \kappa_{2ic;0} = 0.1\mathring{t}, \;\; \textit{(the standard option)}. \end{aligned}$$

- *Attach to each factor ic,* $i \in i^*$, $c \in c^*$, *the one-component static mixture given by the sufficient statistics in the LS form, cf. (8.17). It can be formally chosen as the two-component mixture described above but with completely overlapping components. Thus, all options of the real two-component mixture are copied but point estimates of positions are selected identical* $\hat{\theta}_{1ic;0} = -\hat{\theta}_{2ic;0} = 0$.

- *Set the v-likelihood values assigned to one- and two-component mixtures* ${}^{\lfloor 1}l_{ic;0} = 0$ *and* ${}^{\lfloor 2}l_{ic;0} = 0$.

Learning mode

For $t = 1, \ldots, \mathring{t}$

Perform a step of quasi-Bayes (QB) algorithm estimating parameters of the original mixture.

For $c = 1, \ldots, \mathring{c}$

For $i = 1, \ldots, \mathring{d}$

Weight the data vectors $[\hat{e}_{ic;t}, 1] \equiv$ *[filtering error, 1] by*

$\sqrt{w_{ic;t}} = \sqrt{\text{weight of the factor in question assigned by QB algorithm.}}$

Run in parallel QB algorithm on the weighted data

feed in the one- and two-component mixtures.

Update the v-likelihood values ${}^{\lfloor 1}l_{ic;t}$, ${}^{\lfloor 2}l_{ic;t}$.

end of the cycle over i

end of the cycle over c

end of the cycle over t

Decision mode

For $i = 1, \ldots, \mathring{d}$

For $c = 1, \ldots, \mathring{c}$

Denote the factor ic as ready for splitting if

$$\frac{{}^{\lfloor 2}l_{ic;\mathring{t}}}{{}^{\lfloor 2}l_{ic;\mathring{t}} + {}^{\lfloor 1}l_{ic;\mathring{t}}} > \gamma \text{ and store the terminal values of the LS statistics.}$$

end of the cycle over c

end of the cycle over i

Results of estimation of the two-component mixture on prediction errors allow us to summarize the overall factor-splitting algorithm. The fixed indices $ic; \mathring{t}$ are drop in it.

Algorithm 8.4.4 (Splitting of factor with LS statistics $\hat{\theta}, \hat{r}, C, \nu$)

1. *Define LS statistics of newly created factors as follows*

   $\hat{r}_1$, $\hat{r}_2$ $\equiv$ *estimates of noise covariance of respective components of the two-component mixture estimated on the weighted filtering errors,*

   $\nu_1 = \hat{\beta}\nu$, $\nu_2 = (1-\hat{\beta})\nu$, *where* $\hat{\beta}$ *is the weight corresponding of the first component in the two-component mixture on filtering errors, cf. (8.72),*

   $C_1 = \hat{\beta}^{-1}C$, $C_2 = (1-\hat{\beta})^{-1}C$, *cf. (8.72).*

2. *Reduce temporarily factor structure to significant one, see Algorithm 8.6.1.*

3. *Evaluate the unit eigenvector* q_0 *corresponding to the maximum eigenvalue* ζ *of the LS covariance factor of parameters* C*, see (8.23), using the iterations (8.59).*

4. *Compute the norm of the shift in point estimates of regression coefficient*

$$\rho = \sqrt{\nu\zeta\left(\frac{\hat{r}}{\hat{\beta}(1-\hat{\beta})} - \frac{\hat{r}_1}{1-\hat{\beta}} - \frac{\hat{r}_2}{\hat{\beta}}\right)}.$$

The factor ν respects that inversion of the LS covariance factors is a non-normalized version of the sampling covariance of regression vectors.

The factor is to be denoted as not ready for splitting if the argument of the above square root is negative.

5. *Complement the definition of the LS statistics by $\hat{\theta}_1 = \hat{\theta} + s\rho q_0, \quad \hat{\theta}_2 = \hat{\theta} - s\rho q_0, \quad s \in \{-1, 1\}$. The sign s is selected randomly.*

Problem 8.4.1 (On a systematic choice of the combined factors) *The random choice in the last step of the above algorithm could be replaced by attempting to maximize the resulting v-likelihood. Algorithm 12.3.3 used in shadow cancelling may serve to this purpose.*

8.4.10 Techniques applicable to static mixtures

Normality of the mixture brings nothing specific to this counter-part of Chapter 6. Only a conjecture that Proposition 6.4.10 is applicable even for the considered continuous-valued data is worth of mentioning. It will be judged in future.

8.5 Approximate parameter estimation

Here, results of Section 6.5 are specialized to normal components.

8.5.1 Quasi-Bayes estimation

A direct application of Algorithm 6.5.2 gives the specialized algorithm.

Algorithm 8.5.1 (Quasi-Bayes algorithm with common factors in normal components)
Initial (off line) mode

- *Select structure of the mixture, i.e. specify the number of components $\mathring{c}$ and the ordered lists of factors allocated to the considered components. The structure of each factors ic is determined by the structure of the corresponding data vector Ψ_{ic}.*
- *Select the statistics $L_{ic;0}, D_{ic;0}, \nu_{ic;0}$ determining prior pdfs of the individual factors*

$$GiW_{\theta_{ic},r_{ic}}(V_{ic;0}, \nu_{ic;0}) \equiv GiW_{\theta_{ic},r_{ic}}(L_{ic;0}, D_{ic;0}, \nu_{ic;0}).$$

- *Select the initial values $\kappa_{c;0} > 0$ determining the Dirichlet pdf $Di_\alpha(\kappa_0)$ of component weights α, say, about 0.1$\mathring{t}$.*
- *Compute initial estimates of the component weights $\hat{\alpha}_{c;0} = \frac{\kappa_{c;0}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};0}}$.*

Sequential (on line) mode, *running for $t = 1, 2, \ldots,$*

1. *Acquire the data record d_t.*
2. *Perform, for each individual factor $i \in i^* = \{1, \ldots, \mathring{d}\}$, $c \in c^*$,* a trial updating *of $L_{ic;t-1}$, $D_{ic;t-1}$ by the data vector $\Psi_{ic;t}$ using Algorithm 8.1.2 with the forgetting factor $\lambda = 1$.*

   These updates give values of the predictive pdfs $f(d_{ic;t}|d(t-1), c) = \frac{\mathcal{I}(d(t)|i,c)}{\sqrt{2\pi}\mathcal{I}(d(t-1)|i,c)}$ (2.47), with

$$\mathcal{I}(d(t)|i,c) \equiv \mathcal{I}(L_{ic;t}, D_{ic;t}, \nu_{ic;t}) = \Gamma(0.5\nu_{ic;t})\, {}^{\lfloor d}D_{ic;t}^{-0.5\nu_{ic;t}} \left| {}^{\lfloor \psi}D_{ic;t} \right|^{-0.5} 2^{0.5\nu_{ic;t}} (2\pi)^{0.5\mathring{\psi}}.$$

3. *Compute values of the predictive pdfs* $f(d_t|d(t-1),c) = \prod_{i\in i^*} f(d_{ic;t}|d(t-1),c)$ *for each component* $c \in c^*$.

4. *Compute the probabilistic weights* $w_{c;t}$, *using the formula*

$$w_{c;t} = \frac{\hat{\alpha}_{c;t-1} f(d_t|d(t-1),c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c};t-1} f(d_t|d(t-1),\tilde{c})}, \ c \in c^*.$$

5. *Update the scalars* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}$, $c \in c^*$, *cf. (6.85).*

6. *Update Bayesian parameter estimates of different factors, i.e. update the corresponding statistics*

$$L_{ic;t-1}, D_{ic;t-1}, \nu_{ic;t-1} \rightarrow L_{ic;t}, D_{ic;t}, \nu_{ic;t}.$$

The updating is equivalent to the weighted version of (8.29)

$$\begin{aligned} V_{ic;t} &= \lambda(V_{ic;t-1} + w_{ic;t}\Psi_{ic;t}\Psi'_{ic;t}) + (1-\lambda)\, {}^{\lfloor A}V_{ic;t},\ V_{ic;0} \textit{ given} \\ \nu_{ic;t} &= \lambda(\nu_{ic;t-1} + w_{ic;t}) + (1-\lambda)\, {}^{\lfloor A}\nu_{ic;t}, \qquad \nu_{ic;0} \textit{ given and} \qquad (8.73) \\ w_{ic;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c};t} \qquad (8.74) \end{aligned}$$

with $c_i^* \equiv$ *set pointers to components that contain* i*-th factor.*

Numerically, $D_{ic;t-1}$ *are first multiplied by* $\lambda = forgetting\_factor$ *and then* <u>*real updating*</u> *of* $L_{ic;t-1}$, $D_{ic;t-1}$ *by data vectors* $\Psi_{ic;t}$ *is performed using Algorithm 8.1.2 with* $\lambda = (forgetting\_factor) \times w_{ic;t}$. *The alternative is added "dyad-wise" with weights* $(1 - (forgetting\_factor))\, {}^{\lfloor A}D_{ic;t}$, *where* ${}^{\lfloor A}D_{ic;t}$ *is the corresponding diagonal of the* $L'DL$ *decomposition of* ${}^{\lfloor A}V_{ic;t}$.

7. *Evaluate the point estimates of the mixing weights*

$$\mathcal{E}[\alpha_c|d(t)] = \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t}} \equiv \hat{\alpha}_{c;t}$$

and, if need be, characteristics of the pdf $f(\Theta_{ic}|d(t))$ *describing other parameters* Θ_{ic}.

8. *Go to the beginning of* Sequential mode *while data are available, while* $t \leq \mathring{t}$.

Remark(s) 8.5.1

1. *The predictive pdf could be computed in its Student form (8.31). It makes sense only when prediction errors are explicitly needed.*

2. *The stabilized forgetting could be applied to the statistics* $\kappa_{c;t}$ *with its specific forgetting factor and alternative. This possibility is worth of considering.*

8.5.2 EM estimation

A direct application of EM algorithm 6.5.4 to normal factors looks as follows.

Algorithm 8.5.2 (EM algorithm for normal mixture estimation with common factors)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 0$.
- *Select the structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to the considered components. The structure of each factor* ic *is determined by the structure of the corresponding data vector* Ψ_{ic}.
- *Select the point estimates* $\hat{\Theta}_{icn} \equiv \left[\hat{\theta}_{icn}, \hat{r}_{icn}\right]$ *of parameters* $\Theta_{ic} \equiv [\theta_{ic}, r_{ic}]$ *characterizing the* i*-th normal factor within the* c*-th normal component.*

- *Select* $\kappa_{cn;0} = 1$ *that corresponds with the point estimates* $\hat{\alpha}_{cn} = 1/\mathring{c}$ *of components weights* α_c.

Iterative mode

1. *Use the current point estimate* $\hat{\Theta}_n$ *in the following evaluations.*
2. *Fill* $L'DL$ *decomposition of the extended information matrix related to the* i*-th factor within the* c*-th component as follows*

$$L_{icn;0} = \begin{bmatrix} 1 & \\ \hat{\theta}_{icn}\varepsilon & I \end{bmatrix}, \; D_{icn;0} = \begin{bmatrix} \hat{r}_{icn}\varepsilon & \\ 0 & \varepsilon I \end{bmatrix}, \; \varepsilon > 0, \; \varepsilon \to 0. \tag{8.75}$$

Initialize the corresponding $\nu_{icn;0} = \varepsilon$.

Sequential mode, *running for* $t = 1, 2, \ldots,$

(a) *Construct the data vectors* $\Psi_{ic;t}$.

(b) *Compute for this* $\Psi_{ic;t}$ *the values of predictive pdfs* $f\left(d_{ic;t}|\psi_{ic;t}, \hat{\Theta}_{icn}, c\right) \equiv \mathcal{N}_{d_{ic;t}}\left(\hat{\theta}'_{icn}\psi_{ic;t}, \hat{r}_{icn}\right)$ *for each individual factor* $i \in i^* = \{1, \ldots, \mathring{d}\}$ *in all components* $c \in c^*$ *using the parameter estimates* $\hat{\Theta}_{icn}$ *that are constant during the time cycle of the sequential mode.*
Thus, the approximate, fixed, one-step-ahead predictor with the certainty-equivalent approximation is used.

(c) *Compute the values of the predictive pdfs*

$$f\left(d_t|\psi_{c;t}, \hat{\Theta}_{cn}, c\right) \equiv \prod_{i \in i^*} \mathcal{N}_{d_{ic;t}}\left(\hat{\theta}'_{icn}\psi_{ic;t}, \hat{r}_{icn}\right)$$

for each component $c \in c^*$.

(d) *Compute the probabilities* $w_{cn;t}$ *approximating* δ_{c,c_t}

$$w_{cn;t} = \frac{f\left(d_t|\psi_{c;t}, \hat{\Theta}_{cn}, c\right)\hat{\alpha}_{cn}}{\sum_{\tilde{c} \in c^*} f\left(d_t|\psi_{\tilde{c};t}, \hat{\Theta}_{cn}, \tilde{c}\right)\hat{\alpha}_{\tilde{c}n}}.$$

(e) *Update the statistics determining the log-likelihood functions describing different factors*

$$\begin{aligned} V_{icn;t} &= V_{icn;t-1} + w_{icn;t}\Psi_{ic;t}\Psi'_{ic;t}, \; \nu_{icn;t} = \nu_{icn;t-1} + w_{icn;t} \\ w_{icn;t} &= \sum_{\tilde{c} \in c_i^*} w_{\tilde{c}n;t}. \end{aligned} \tag{8.76}$$

The set $c_i^* \equiv$ *includes pointers to components that contain* i*-th factor.*
The recursive step (8.76) is numerically performed by updating $L'DL$ *decompositions of extended information matrices with data vectors* $\Psi_{ic;t}$ *weighted by* $w_{icn;t}$ *(using Algorithm 8.1.2).*

(f) *Update* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}, \; c \in c^*$.

(g) *Go to the beginning of* Sequential mode *if* $t \leq \mathring{t}$. *Otherwise continue.*

3. *Find the new point estimates* $\hat{\Theta}_{ic(n+1)} = \left[{}^{\lfloor\psi}L^{-1}_{icn;\mathring{t}} \, {}^{\lfloor d\psi}L_{icn;\mathring{t}}, \, {}^{\lfloor d}D_{icn;\mathring{t}}/\nu_{icn;\mathring{t}}\right]$ *of* $\Theta_{ic} \equiv [\theta_{ic}, r_{ic}]$ *and* $\hat{\alpha}_{c(n+1)} \propto \kappa_{cn;\mathring{t}}$.
These values are maximizing arguments of the n*-th approximate likelihood.*
4. *Stop the log-likelihood value*

$$\sum_{c \in c^*} \kappa_{cn;\mathring{t}} \left[\ln(\hat{\alpha}_{cn;\mathring{t}}) - 0.5 \sum_{i \in i^*} \ln({}^{\lfloor d}D_{icn;\mathring{t}}/\nu_{icn;\mathring{t}})\right] \tag{8.77}$$

is not increasing any more. Otherwise set $n = n + 1$ *and go to the beginning of* Iterative mode.

8.5.3 Batch quasi-Bayes estimation

Here, Algorithm 6.5.5 is specialized to normal mixtures. In this way, processing-order-independent, batch quasi-Bayes estimation of normal mixtures is gained.

Algorithm 8.5.3 (BQB algorithm for normal mixture estimation with common factors)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 0$.
- *Select the structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to the considered components. The structure of each factor* ic *is determined by the structure of the corresponding data vector* Ψ_{ic}.
- *Set the maximum of the v-log-likelihood* $\bar{l} = -\infty$.
- *Select the statistics* $\bar{L}_{ic}, \bar{D}_{ic}, \bar{\nu}_{ic}$ *determining (flat) pre-prior pdfs of the individual factors*

$$GiW_{\theta_{ic},r_{ic}}\left(\bar{V}_{ic}, \bar{\nu}_{ic}\right) \equiv GiW_{\theta_{ic},r_{ic}}\left(\bar{L}_{ic}, \bar{D}_{ic}, \bar{\nu}_{ic}\right).$$

  Select the values $\bar{\kappa}_c > 0$ *determining a flat pre-prior Dirichlet pdf on component weights.*

  These values serve for flattening.
- *Select the statistics* $L_{icn}, D_{icn}, \nu_{icn}$ *determining prior pdfs of the individual factors*

$$GiW_{\theta_{ic},r_{ic}}(V_{icn}, \nu_{icn}) \equiv GiW_{\theta_{ic},r_{ic}}(L_{icn}, D_{icn}, \nu_{icn}).$$

- *Select the initial values* $\kappa_{cn} > 0$ *determining a prior Dirichlet pdf on component weights.*
- *Make copies* $L_{ic;0} = L_{icn}$, $D_{ic;0} = D_{icn}$, $\nu_{ic;0} = \nu_{icn}$ *and* $\kappa_{c;0} = \kappa_{cn}$.

Iterative mode

1. *Use the current prior pdf* $f_n(\Theta) = Di_\alpha(\kappa_n) \prod_{c\in c^*} \prod_{i\in i^*} GiW_{\theta_{ic},r_{ic}}(L_{icn}, D_{icn}, \nu_{icn})$ *in the following evaluations.*
2. *Set the value of v-log-likelihood* $l_{n;0} = 0$.
3. *Compute the point estimates of the components weights* $\hat{\alpha}_{cn} = \frac{\kappa_{cn}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c}n}}$.

   Sequential mode, *running for* $t = 1, 2, \ldots,$

   (a) *Construct the data vectors* $\Psi_{ic;t}$.

   (b) *Compute for these data vectors the values of the predictive pdfs*

$$f_n(d_{ic;t}|\psi_{ic;t}, c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c) f_n(\Theta_{ic})\, d\Theta_{ic} =$$

$$= \frac{\Gamma(0.5(\nu_{icn}+1)) \left[{}^{\lfloor d}D_{icn}(1+\zeta_{icn;t})\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu_{icn}) \left(1 + \frac{\hat{e}^2_{icn;t}}{{}^{\lfloor d}D_{icn}(1+\zeta_{ic;t})}\right)^{0.5(\nu_{icn}+1)}},$$

$$\hat{e}_{icn;t} \equiv d_t - {}^{\lfloor d\psi}L'_{icn} \left({}^{\lfloor \psi}L'_{icn}\right)^{-1} \psi_{ic;t}, \quad \zeta_{icn;t} = \psi'_{ic;t}\, {}^{\lfloor \psi}L^{-1}_{icn}\, {}^{\lfloor \psi}D^{-1}_{icn} \left({}^{\lfloor \psi}L'_{icn}\right)^{-1} \psi_{ic;t}.$$

   for each factor $i \in i^*$ *in all components* $c \in c^*$ *using the prior pdfs* $GiW_{\theta_{ic},r_{ic}}(L_{icn}, D_{icn}, \nu_{icn})$ *that are constant during time cycle.*

   (c) *Compute the values of predictive pdfs* $f_n(d_t|\psi_{c;t}, c) \equiv \prod_{i\in i^*} f_n(d_{ic;t}|\psi_{ic;t}, c)$ *for each component* $c \in c^*$.

   (d) *Update the v-log-likelihood* $l_{n;t} = l_{n;t-1} + \ln\left(\sum_{c\in c^*} \hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)\right)$.

(e) *Compute the probabilistic weights* $w_{cn;t}$ *approximating* δ_{c,c_t}

$$w_{cn;t} = \frac{\hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}n} f_n(d_t|\psi_{\tilde{c};t}, \tilde{c})}.$$

(f) *Update the statistics determining the posterior pdfs evolving from copies of the prior statistics* $L_{ic;0}$, $D_{ic;0}$, $\nu_{ic;0}$ *and* $\kappa_{c;0}$. *The factorized equivalent of the update of the extended information matrix is performed*

$$\begin{aligned} V_{icn;t} &= V_{icn;t-1} + w_{icn;t}\Psi_{ic;t}\Psi'_{ic;t}, \quad V_{icn;t} \equiv L'_{icn;t}D_{icn;t}L_{icn;t} \\ \kappa_{c;t} &= \kappa_{c;t-1} + w_{cn;t} \\ \nu_{icn;t} &= \nu_{icn;t-1} + w_{icn;t} \text{ with} \\ w_{icn;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c}n;t}. \end{aligned} \tag{8.78}$$

The set $c_i^* \equiv$ *includes the pointers to components that contain* i*-th factor.*

(g) *Go to the beginning of* Sequential mode *if* $t \leq \mathring{t}$. *Otherwise continue.*

4. *Stop if the v-likelihood of the mixture does not increase* $l_{n;\mathring{t}} < \bar{l}$, *or the compared valued are perceived as the same in the vein of Proposition 6.1.2, or* $n \geq \mathring{n}$. *Otherwise set* $\bar{l} = l_{n;\mathring{t}}$, *increase the iteration counter* $n = n + 1$, *apply flattening operation to* $f(\Theta_{ic}|d(\mathring{t}))$ *and* $Di_\alpha(\kappa_{c;\mathring{t}})$

$$\begin{aligned} D_{icn} &= \Lambda_n D_{ic;\mathring{t}} + (1-\Lambda_n)\bar{D}_{ic}, \quad \nu_{icn} = \Lambda_n \nu_{ic;\mathring{t}} + (1-\Lambda_n)\bar{\nu}_{ic} \\ \kappa_{cn} &= \Lambda_n \kappa_{c;\mathring{t}} + (1-\Lambda_n)\bar{\kappa}_c, \quad \Lambda_n \to 0.5 \text{ according to Proposition 6.4.4.} \end{aligned}$$

5. *Go to the beginning of* Iterative mode.

Remark(s) 8.5.2

As discussed in Remarks 6.5.4, the BQB algorithm uses for estimating δ_{c,c_t} *Bayesian predictors. They respect uncertainty of the current estimates of unknown parameters. Predictions become too cautious if this uncertainty is initialized to high values. This may break down the algorithm completely. In the context of normal pdfs, it happens if the factor* $\zeta = \psi' C \psi$ *becomes too high. Then, the influence of prediction errors* $\{\hat{e}\}$ *that predominantly indicates membership of the data vector to the specific components is suppressed too much.*

Knowing the danger, the remedy is simple. Essentially, predictions used in the EM algorithm that ignore these uncertainties have to be used in several initial iterative steps of the algorithm, i.e. the value $\zeta = 0$ *is enforced.*

8.5.4 Universal approximate estimation

The universal approximate estimation Algorithm 6.5.6 specializes to normal mixtures as follows.

Algorithm 8.5.4 (Universal mixture estimation of normal mixtures with common factors)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 0$.
- *Select the structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to the considered components. The structure of each factor* ic *is determined by the structure of the corresponding data vector* Ψ_{ic}.
- *Set the maximum of the v-log-likelihood* $\bar{l} = -\infty$.

- *Select the statistics $\bar{L}_{ic}, \bar{D}_{ic}, \bar{\nu}_{ic}$ determining (flat) pre-prior pdfs of the individual factors*

$$GiW_{\theta_{ic},r_{ic}}\left(\bar{V}_{ic}, \bar{\nu}_{ic}\right) \equiv GiW_{\theta_{ic},r_{ic}}\left(\bar{L}_{ic}, \bar{D}_{ic}, \bar{\nu}_{ic}\right).$$

  Select the values $\bar{\kappa}_c > 0$ determining a flat pre-prior Dirichlet pdf on the component weights.
- *Select the statistics $\bar{L}_{icn}, \bar{D}_{icn}, \bar{\nu}_{icn}$ determining flat pre-prior pdfs of the individual factors*

$$GiW_{\theta_{ic},r_{ic}}\left(\bar{V}_{ic}, \bar{\nu}_{ic}\right) \equiv GiW_{\theta_{ic},r_{ic}}\left(\bar{L}_{ic}, \bar{D}_{ic}, \bar{\nu}_{ic}\right).$$

- *Select the initial values $\bar{\kappa}_{cn} > 0$ determining a flat pre-prior Dirichlet pdf on the component weights.*
- *Select the statistics $L_{icn}, D_{icn}, \nu_{icn}$ determining the prior pdfs of the individual factors*

$$GiW_{\theta_{ic},r_{ic}}(V_{ic}, \nu_{ic}) \equiv GiW_{\theta_{ic},r_{ic}}(L_{ic}, D_{ic}, \nu_{ic}).$$

- *Select the initial values $\kappa_{cn} > 0$ determining a prior Dirichlet pdf on the component weights.*
- *Select the algorithmic variant* **av** $\in$ {**quasi-Bayes, quasi-EM, batch-quasi-Bayes, batch-EM**}.

Iterative mode

1. Make copies $L_{oic;0} = L_{icn}$, $D_{oic;0} = D_{icn}$, $\nu_{oic;0} = \nu_{icn}$ *and* $\kappa_{oc;0} = \kappa_{cn}$ *if **av** is a* batch variant.
2. *Compute the point estimates of the components weights* $\hat{\alpha}_{cn} = \frac{\kappa_{cn}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c}n}}$ *and point estimates* $\hat{\Theta}_{icn} = \left[{}^{\lfloor\psi}L_{icn}^{-1}\, {}^{\lfloor d\psi}L_{icn},\, {}^{\lfloor d}D_{icn}/\nu_{icn}\right]$ *of parameters Θ_{ic} of the individual parameterized factors.*
3. *Set the value of v-log-likelihood $l_{n;0} = 0$.*
   Sequential mode, *running for $t = 1, 2, \ldots,$*
   (a) *Use data d_t to construct the data vectors $\Psi_{ic;t}$.*
   (b) *Compute values of the predictive pdfs*

$$f_n(d_{ic;t}|\psi_{ic;t}, c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c) f_n(\Theta_{ic})\, d\Theta_{ic} =$$
$$= \frac{\Gamma(0.5(\nu_{icn}+1))\left[{}^{\lfloor d}D_{icn}(1+\zeta_{icn;t})\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu_{icn})\left(1+\frac{\hat{e}^2_{icn;t}}{{}^{\lfloor d}D_{icn}(1+\zeta_{ic;t})}\right)^{0.5(\nu_{icn}+1)}},$$
$$\hat{e}_{icn;t} \equiv d_t - \hat{\theta}'_{icn}\psi_{ic;t}, \quad \zeta_{icn;t} = \psi'_{ic;t}\, {}^{\lfloor\psi}L_{icn}^{-1}\, {}^{\lfloor\psi}D_{icn}^{-1}\left({}^{\lfloor\psi}L'_{icn}\right)^{-1}\psi_{ic;t},$$

   for each individual factor $i \in i^* = \{1, \ldots, \mathring{d}\}$ in all components $c \in c^*$, using the prior pdfs $GiW_{\theta_{ic},r_{ic}}(L_{icn}, D_{icn}, \nu_{icn})$.[1]
   (c) *Compute values of the predictive pdfs $f_n(d_t|\psi_{c;t}, c) \equiv \prod_{i\in i^*} f_n(d_{ic;t}|\psi_{ic;t}, c)$ for each component $c \in c^*$.*[1]
   (d) *Update v-log-likelihood*[1] $l_{n;t} = l_{n;t-1} + \ln\left(\sum_{c\in c^*} \hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)\right)$.
   (e) *Compute the probabilistic weights $w_{cn;t}$ approximating δ_{c,c_t} according to the variant **av** used*
   av= *-Bayes

$$w_{cn;t} \equiv \frac{\hat{\alpha}_{cn} f_n(d_t|\psi_{c;t}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}n} f_n(d_t|\psi_{\tilde{c};t}, \tilde{c})},$$

   av=*-EM

$$w_{cn;t} = \frac{\hat{\alpha}_{cn}\prod_{i\in i^*}\mathcal{N}_{d_{ic;t}}(\hat{\theta}'_{icn}\psi_{ic;t}, \hat{r}_{icn})}{\sum_{\tilde{c}\in c^*}\hat{\alpha}_{\tilde{c}n}\prod_{i\in i^*}\mathcal{N}_{d_{i\tilde{c};t}}(\hat{\theta}'_{i\tilde{c}n}\psi_{i\tilde{c};t}, \hat{r}_{i\tilde{c}n})}.$$

[1] This evaluation is not necessary for EM versions but suits for a use in a wider Bayesian context.

(f) *Perform data updating according to the variant* ***av*** *used*

av=quasi-*

$$\begin{aligned}
V_{icn;t} &= V_{icn;t-1} + w_{icn;t}\Psi_{ic;t}\Psi'_{ic;t}, \quad V_{icn;t} \equiv L'_{icn;t}D_{icn;t}L_{icn;t} \\
\kappa_{c;t} &= \kappa_{c;t-1} + w_{cn;t}, \ \nu_{icn;t} = \nu_{icn;t-1} + w_{icn;t} \ \textit{with} \\
w_{icn;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c}n;t}.
\end{aligned}$$

Correct the point estimates of parameters

$$\hat{\alpha}_{cn} = \frac{\kappa_{cn}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c}n}}, \quad \hat{\theta}_{icn} = {}^{\lfloor\psi}L^{-1}_{icn;t}\,{}^{\lfloor d\psi}L_{icn;t}, \quad \hat{r}_{icn} = \frac{{}^{\lfloor d}D_{icn;t}}{\nu_{icn;t}}.$$

av=batch-* *Update copies*

$$\begin{aligned}
V_{oic;t} &= V_{oic;t-1} + w_{oic;t}\Psi_{ic;t}\Psi'_{ic;t}, \quad V_{oic;t} \equiv L'_{oic;t}D_{oic;t}L_{oic;t} \\
\kappa_{oc;t} &= \kappa_{oc;t-1} + w_{cn;t}, \ \nu_{oic;t} = \nu_{oic;t-1} + w_{icn;t}, \quad \textit{with} \\
w_{icn;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c}n;t}.
\end{aligned}$$

In all cases, the set c_i^* includes the pointers to the components that contain i-th factor.

(g) *Go to the beginning of* Sequential mode *if $t \le \mathring{t}$. Otherwise continue.*

4. *Stop $l_{n;\mathring{t}} < \bar{l}$ or $l_{n;\mathring{t}}, \bar{l}$ are not equal in the sense of Proposition 6.1.2 or $n \ge \mathring{n}$.*

5. *Set $\bar{l} = l_{n;\mathring{t}}$, increase the iteration counter $n = n + 1$.*
   For ***av=batch-****, set*
   $$V_{icn} = V_{oic;\mathring{t}}, \ \nu_{icn} = \nu_{oic;\mathring{t}}, \ \kappa_{cn} = \kappa_{oc;\mathring{t}}$$
   and refresh the point estimates
   $$\hat{\Theta}_{icn} = \left[{}^{\lfloor\psi}L^{-1}_{icn}\,{}^{\lfloor d\psi}L_{icn}, \ {}^{\lfloor d}D_{icn}/\nu_{icn}\right].$$

6. *Apply the flattening operation to $Di_\alpha(\kappa_{cn})\prod_{c\in c^*}\prod_{i\in i^*} f_n(\Theta_{ic})$*
   $$\begin{aligned}
   D_{icn} &= \Lambda_n D_{ic;\mathring{t}} + (1-\Lambda_n)\bar{D}_{ic}, \quad \nu_{icn} = \Lambda_n \nu_{ic;\mathring{t}} + (1-\Lambda_n)\bar{\nu}_{ic} \\
   \kappa_{icn} &= \Lambda_n \kappa_{ic;\mathring{t}} + (1-\Lambda_n)\bar{\kappa}_c, \quad \Lambda_n \to 0.5 \ \textit{according to Proposition 6.4.4.}
   \end{aligned}$$

7. *Go to the beginning of* Iterative mode.

Remark(s) 8.5.3

Note that the flattening factor has to be almost zero if the standard EM algorithm should be obtained.

8.6 Structure estimation

Efficiency of the advisory system depends strongly on the quality of the model of the o-system. A good choice of the used structure of the mixture , see Agreement 5.2.1, is quite important in this respect. It includes an appropriate selection of

- variables among those measured on the o-system that should be used by the p-system,
- structure of individual factors, it means mostly data entries and delayed data entries used in the state vector in the phase form,

- structure of components, i.e. the order of factors used in them,
- structure of the mixture determined by the number of components and specification of the common factors in them.

All these tasks can be formally solved as the Bayesian estimation of discrete-valued pointers to alternative structures. The excessive number of possible alternatives makes the problem non-trivial. Details needed for implementing the strategy outlined in Section 6.6 are presented here.

8.6.1 Estimation of factor structure

Here, we summarize facts that make basis for the tailored choice of the neighborhood for normal factors. Publications [36, 76, 84] serve as references to details.

Agreement 8.6.1 (Nested normal factors) *Let regression vectors $\psi, \tilde{\psi}$ of a pair of normal factors predicting the same data item d fulfill*

$$\psi' = \left[\tilde{\psi}', \bullet\right] \tag{8.79}$$

with $\bullet$ marking here and below arbitrary arrays of appropriate dimensions. Let also the statistics V_0, $\tilde{V}_0$, ${}^{\lfloor A}V$, ${}^{\lfloor A}\tilde{V}$ of conjugate prior pdfs $f(\Theta) = GiW_{\Theta}(V_0, \nu_0)$, $f\left(\tilde{\Theta}\right) = GiW_{\tilde{\Theta}}\left(\tilde{V}_0, \nu_0\right)$ and alternative pdfs ${}^{\lfloor A}f(\Theta) = GiW_{\Theta}\left({}^{\lfloor A}V, {}^{\lfloor A}\nu\right)$, ${}^{\lfloor A}f(\tilde{\Theta}) = GiW_{\tilde{\Theta}}\left({}^{\lfloor A}\tilde{V}, {}^{\lfloor A}\tilde{\nu}\right)$ (8.16) fulfill

$$V_0 = \begin{bmatrix} \tilde{V}_0 & \bullet \\ \bullet & \bullet \end{bmatrix}, \quad {}^{\lfloor A}V = \begin{bmatrix} {}^{\lfloor A}\tilde{V} & \bullet \\ \bullet & \bullet \end{bmatrix}. \tag{8.80}$$

Then, we say that the factor $f(d|\tilde{\psi}, \tilde{\Theta})$ is nested into the factor $f(d|\psi, \Theta)$.

Proposition 8.6.1 (Estimation and prediction with nested factors) *Let natural conditions of decision making, Requirement 2.5.1, hold. The considered parameterized factor (8.1) is normal, given by the regression vector ψ, and a conjugate prior (3.12) as well as conjugate alternative, see Section 3.1, in the stabilized forgetting are used. Let another normal factor $f\left(d|\tilde{\psi}, \tilde{\Theta}\right)$, determined by the regression vector $\tilde{\psi}$, be nested into the factor $f(d|\psi, \Theta)$. Then, the statistic V_t describing estimates of the factor $f(d|\psi, \Theta)$, that evolves according to the recursion*

$$V_t \;=\; \lambda(V_{t-1} + \Psi\Psi') + (1-\lambda)\, {}^{\lfloor A}V_t, \; V_0 \text{ given.}$$

At the same time, it performs updating of the statistics $\tilde{V}_t$ as

$$V_t = \begin{bmatrix} \tilde{V}_t & \bullet \\ \bullet & \bullet \end{bmatrix}, \; \forall t \in t^*. \tag{8.81}$$

Let us decompose all symmetric positive definite V-matrices into LDL' decomposition (it differs from the decomposition $L'DL$ used otherwise in learning!)

$$\begin{aligned} V &= LDL', \; L \text{ is lower triangular matrix with unit diagonal} \\ & \; D \text{ is diagonal matrix with non-negative entries} \\ L &= \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L & {}^{\lfloor \psi}L \end{bmatrix}, \; D = \begin{bmatrix} {}^{\lfloor d}D & 0 \\ 0 & {}^{\lfloor \psi}D \end{bmatrix}, \; {}^{\lfloor d}D \text{ is scalar.} \end{aligned} \tag{8.82}$$

Then,

$$L = \begin{bmatrix} \tilde{L} & \bullet \\ \bullet & \bullet \end{bmatrix}, \; D = \begin{bmatrix} \tilde{D} & 0 \\ 0 & \bullet \end{bmatrix}. \tag{8.83}$$

The normalization integral defining the likelihood function of data for the given structure is

$$\begin{aligned} \mathcal{I}(V, \nu) &= \Gamma(0.5\nu) \left({}^{\lfloor d}D\right)^{-0.5\nu} \left|{}^{\lfloor \psi}D\right|^{-0.5} 2^{0.5(\nu-2)} (2\pi)^{0.5\mathring{\psi}} \\ \Gamma(x) &= \int_0^{\infty} z^{x-1} \exp(-z)\, dz < \infty \text{ for } x > 0. \end{aligned} \tag{8.84}$$

The v-likelihood corresponding to the considered structure is

$$f(d(\mathring{t})) = \frac{\mathcal{I}(V_{\mathring{t}}, \nu_{\mathring{t}})}{\mathcal{I}(V_0, \nu_0)}. \tag{8.85}$$

Proof: Detailed evaluations can be found in [80, 84]. Nesting of the extended information matrix follows directly from assumptions on nesting of the initial and alternative values together with the nesting of data vectors and the form of updating. The definition of LDL' (!) decomposition implies its nesting. The form of the normalizing integral $\mathcal{I}(V, \nu)$ given in Proposition 8.1.7 is preserved in spite of the change of the decomposition type as it depends on determinant of matrices V and ${}^{\lfloor\psi}V$ that can be unambiguously computed from diagonals D and ${}^{\lfloor\psi}D$ only. The formula (8.85) results from (2.47) and the chain rule for pdfs, Proposition 2.3.4. □

Remark(s) 8.6.1

1. *The version with $L'DL$ decomposition is prepared, too. It is nested in the tail position of ψ and L, D.*

2. *Note that the normal pdf belongs to the exponential family, see (3.5), with $B(\Psi) = \Psi\Psi'$. The nested normal model is obtained by cancelling entries in regression vector at its tail. This can obviously be interpreted as a linear operator defining a nesting mapping on $B(\Psi)$, see Proposition 3.3.1. Thus, Proposition 8.6.1 is mostly implied by Agreement 8.6.1. The cancellation at tails only and the choice of the decomposition LDL' make elements of LDL' nested, too.*

3. *The nesting property can be and will be used for an efficient evaluation of $f(d(\mathring{t})|s)$ for a rich neighborhood of some structure s_0. It contains*

   - *all regression vectors that can be gained by deleting one entry from that determined by s_0 or by adding one regressor from the richest regression vector ψ_r;*

   - *all regression vectors that are nested into regression vectors specified under the previous item.*

   By evaluating about $\mathring{\psi}_r$ different LDL' decompositions of V we evaluate about $0.5\mathring{\psi}_r\mathring{\psi}_{s_0}(\mathring{\psi}_{s_0} + 1)$ values of likelihood function. Moreover, re-computation of these LDL' decomposition that guarantee appropriate nesting is computationally cheap. It can be performed by the repetitive permutation of adjacent entries in the regression vector and corresponding restoration of the LDL' decomposition of the extended information matrix V, see Proposition 8.1.4.

4. *Nesting of the prior statistics is guaranteed for weekly informative prior pdfs. The nesting is violated when a non-trivial physical knowledge is used, see Section 8.3. An efficient algorithm coping with the induced problems is described in [85].*

8.6.2 Structure estimation in factor splitting

Here, we address the problem that arisen in Section 8.4.8 in connection with branching by factor shifting. Essentially, we need to perform splitting in the space of significant parameters and to extend it to the original parameter space.

Particular steps and their solutions are summarized in the following algorithm.

Algorithm 8.6.1 (Steps of factor modification in a significant subspace)

1. *Estimation of the factor structure $f(\theta, r) = GiW_{\theta,r}(L, D, \nu)$ by analysis of the given $L'DL$ decomposition of the extended information matrix.*

   Using Algorithm 8.1.3, we transform $L'DL$ to $\tilde{L}\tilde{D}\tilde{L}'$ and apply structure estimation algorithm outlined in Section 8.6.1.

2. *Analysis of the marginal* $f({}^{\lfloor a}\theta, r) = GiW_{{}^{\lfloor a}\theta,r}(\cdot)$ *pdf on significant regression coefficients* ${}^{\lfloor a}\theta$.
   The needed marginal pdf

$$f\left({}^{\lfloor a}\theta, r\right) = GiW_{{}^{\lfloor a}\theta,r}\left(\begin{bmatrix} 1 & \\ {}^{\lfloor da}L & {}^{\lfloor a}L \end{bmatrix}, \begin{bmatrix} {}^{\lfloor d}D & \\ & {}^{\lfloor a}D \end{bmatrix}, \nu\right)$$

   is obtained directly from the original pdf whose statistics are split as follows $f(\theta, r) = GiW_{\theta,r}(L, D, \nu)$

$$L \equiv \begin{bmatrix} 1 & & \\ {}^{\lfloor da}L & {}^{\lfloor a}L & \\ {}^{\lfloor db}L & {}^{\lfloor ab}L & {}^{\lfloor b}L \end{bmatrix}, \quad D \equiv \begin{bmatrix} {}^{\lfloor d}D & & \\ & {}^{\lfloor a}D & \\ & & {}^{\lfloor b}D \end{bmatrix}.$$

   This simple relationships holds if ${}^{\lfloor a}\theta$ *are at leading position of all regression coefficients, see Proposition 8.1.8. This configuration is achieved by their permutation with the corresponding restoration of* $L'DL$ *according to Proposition 8.1.5.*

   The conditional pdf of insignificant coefficients $f\left({}^{\lfloor b}\theta | \, {}^{\lfloor a}\theta, r\right)$ *is obtained as byproduct. The statistics determining it are in the unexploited part of the permuted extended information matrix.*

   The mentioned analysis deals with the least-squares version of the found marginal pdf. One-to-one relationships (8.18), (8.19), (8.20) provide it.

3. *Modification (splitting) of the factor reduced on* ${}^{\lfloor a}\theta, r$.
   Solution is presented in Section 8.4.8 and modifies the reduced decomposition of the extended information matrix to the other one given by, say,

$$\begin{bmatrix} 1 & \\ {}^{\lfloor da}\underline{L} & {}^{\lfloor a}\underline{L} \end{bmatrix}, \begin{bmatrix} {}^{\lfloor d}\underline{D} & \\ & {}^{\lfloor a}\underline{D} \end{bmatrix}.$$

4. *Extension of the modified pdf*

$$\underline{f}({}^{\lfloor a}\theta, r) = GiW_{{}^{\lfloor a}\theta,r}\left(\begin{bmatrix} 1 & \\ {}^{\lfloor da}\underline{L} & {}^{\lfloor a}\underline{L} \end{bmatrix}, \begin{bmatrix} {}^{\lfloor d}\underline{D} & \\ & {}^{\lfloor a}\underline{D} \end{bmatrix}, \nu\right)$$

   to a new pdf on the original space of parameters. It is done according to the chain rule

$$\begin{aligned} \underline{f}(\theta, r) &\equiv f\left({}^{\lfloor b}\theta | \, {}^{\lfloor a}\theta, r\right) \underline{f}({}^{\lfloor a}\theta, r) = GiW_{\theta,r}(\underline{L}, \underline{D}, \nu), \quad \textit{where} \\ \underline{L} &\equiv \begin{bmatrix} 1 & & \\ {}^{\lfloor da}\underline{L} & {}^{\lfloor a}\underline{L} & \\ {}^{\lfloor db}L & {}^{\lfloor ab}L & {}^{\lfloor b}L \end{bmatrix}, \quad \underline{D} \equiv \begin{bmatrix} {}^{\lfloor d}\underline{D} & & \\ & {}^{\lfloor a}\underline{D} & \\ & & {}^{\lfloor b}D \end{bmatrix}. \end{aligned}$$

   In words, the original ${}^{\lfloor a}$*-parts are simply replaced by the new* ${}^{\lfloor a}$*-parts.*

8.6.3 Estimation of component structure

A practical universal search for the best order of factors forming the component is unsolved even for the normal components. Their specific form allows us to get a deeper insight that will hopefully lead to practical algorithms.

Normal component can be written in the matrix "equation form"

$$d_t = {}^{\lfloor M}\theta' \phi_{t-1} + G' e_t, \quad \text{where} \tag{8.86}$$

${}^{\lfloor M}\theta$ is a matrix of regression coefficients complemented by zeros so that we can use the state vector ϕ_{t-1} that is common for all entries of d_t. $G' \equiv {}^{\lfloor e}L' \, {}^{\lfloor e}D^{0.5}$ is *Choleski square root* of the noise covariance and e_t is normal white zero-mean noise with non-correlated entries and common unit variance.

The chosen triangular form of G defines uniquely decomposition of this component into factors as well as their parameterizations, see Proposition 8.1.13. If we permute two entries of d_t, we have to permute corresponding columns of ${}^{\lfloor M}\theta$ and G. In this way, triangular form of G is spoiled and has to be recovered by using invariance of the noise covariance to orthogonal transformation applied from right to G. Such a recovered G has generally different number of zero elements. Its inversion combines columns of ${}^{\lfloor M}\theta$ in a different way, too. Thus, the number of zero entries in parameters of new factors differs from the former one.

The result depends in a complex way on linear dependencies of modified rows of ${}^{\lfloor M}\theta$, rows of G and G^{-1}. Up to now, we found no operational way how to describe them. Thus, we have no guideline how to search for "permutation" candidates.

8.6.4 Merging and cancelling of components

Normality of components brings just specific form of statistics and of the normalizing integral $\mathcal{I}(V,\nu)$. It allows us to present just the final algorithms of merging and cancelling.

Merging of a group of normal components

Here, normal counterpart of Algorithm 6.6.5 is presented.

Algorithm 8.6.2 (Merging of a group of normal components)
Initial mode

- *Estimate the mixture with a sufficient amount of components $\mathring{c}$ so that statistics* ${}^{\lfloor\mathring{c}}\kappa_t,\ {}^{\lfloor\mathring{c}}\nu_{ic;t},\ {}^{\lfloor\mathring{c}}L_{ic;t},\ {}^{\lfloor\mathring{c}}D_{ic;t}\ t \in \{0,\mathring{t}\},\ c = 1,\ldots,\mathring{c},\ i = 1,\ldots,\mathring{d}$ *are at disposal.*
- *Set pointers $c = 1, \tilde{c} = 2$ to trial components to be merged.*

Evaluation mode

Set the indicator of the common structure $cs = 0$.

For $i = 1,\ldots,\mathring{d}$

Set $cs = -1$ and break the cycle over i if the structures of θ_{ic} and $\theta_{i\tilde{c}}$ differ.

end of the cycle over i

Do if $cs = 0$

Evaluate the common part of the trial merger

$\tilde{\kappa}_{\mathring{t}} = \kappa_{c;\mathring{t}} + \kappa_{\tilde{c};\mathring{t}}) - 1,\ \tilde{\kappa}_0 = \kappa_{c;0} + \kappa_{\tilde{c};0} - 1.$

Evaluate and store the factor-related parts of the trial merger

For $i = 1,\ldots,\mathring{d}$

use Proposition 10.1.2 and (8.25)

$\tilde{\nu}_{i;\mathring{t}} = \nu_{ic;\mathring{t}} + \nu_{i\tilde{c};\mathring{t}}$

$\tilde{L}'_{i;\mathring{t}}\tilde{D}_{i;\mathring{t}}\tilde{L}_{i;\mathring{t}} = L'_{ic;\mathring{t}}D_{ic;\mathring{t}}L_{ic;\mathring{t}} + L'_{i\tilde{c};\mathring{t}}D_{i\tilde{c};\mathring{t}}L'_{i\tilde{c};\mathring{t}}.$

end of the cycle over i

Evaluate the change $\tilde{l}$ of log-v-likelihood expected after the definite merging

$$\tilde{l} = +\left\{-\ln\left(\Gamma(\kappa_{c;\mathring{t}})\right) - \ln\left(\Gamma(\kappa_{\tilde{c};\mathring{t}})\right) + \ln\left(\Gamma(\tilde{\kappa}_{\mathring{t}})\right) - \ln\left(\left(\sum_{c=1}^{\mathring{c}-1}\kappa_{c;\mathring{t}}\right) - 1\right)\right\} -$$

$$-\left\{-\ln\left(\Gamma(\kappa_{c;0})\right) - \ln\left(\Gamma(\kappa_{\tilde{c};0})\right) + \ln\left(\Gamma(\tilde{\kappa}_0)\right) - \ln\left(\left(\sum_{c=1}^{\mathring{c}}\kappa_{c;0}\right) - 1\right)\right\}.$$

For $i = 1,\ldots,\mathring{d}$

(factor parts)

$$\tilde{l} = \tilde{l} + \left\{\ln(\mathcal{I}(\tilde{L}_{i;\mathring{t}},\tilde{D}_{i;\mathring{t}},\tilde{\nu}_{i;\mathring{t}})) - \ln(\mathcal{I}(L_{ic;\mathring{t}},D_{ic;\mathring{t}},\nu_{ic;\mathring{t}})) - \ln(\mathcal{I}(L_{i\tilde{c};\mathring{t}},D_{i\tilde{c};\mathring{t}},\nu_{i\tilde{c};\mathring{t}}))\right\} -$$

$$-\left\{\ln(\mathcal{I}(\tilde{L}_{i;0},\tilde{D}_{i;0},\tilde{\nu}_{i;0})) - \ln(\mathcal{I}(L_{ic;0},D_{ic;0},\nu_{ic;0})) - \ln(\mathcal{I}(L_{i\tilde{c};0},D_{i\tilde{c};0}\nu_{i\tilde{c};0}))\right\}.$$

end of the cycle over i

end of the condition $cs = 0$

Do if $\tilde{l} \le 0$ *or* $cs < 0$

Set $\tilde{c} = \tilde{c} + 1$.

Go to the beginning of Evaluation mode *if* $\tilde{c} \le \mathring{c}$. *Otherwise continue.*

Set $c = c + 1$ *and* $\tilde{c} = c + 1$.

Go to the beginning of Evaluation mode *if* $c < \mathring{c}$. *Otherwise stop.*

else replace statistics related to the component c by

$\tilde{\kappa}_{\mathring{t}},\ \tilde{\kappa}_0,\ \left\{\tilde{L}_{i;\mathring{t}},\ \tilde{D}_{i;\mathring{t}},\ \tilde{L}_{i;0},\ \tilde{D}_{i;0},\ \tilde{\nu}_{i;\mathring{t}},\ \tilde{\nu}_{i;0}\right\}_{i=1}^{\mathring{d}}$.

Swap the components $\mathring{c}$ *and* $\tilde{c}$.

Decrease $\mathring{c} = \mathring{c} - 1$, *i.e. omit the component* $\tilde{c}$.

Set $\tilde{c} = c + 1$ *if* $\tilde{c} > \mathring{c}$.

end of the test on improvement of v-likelihood and of $cs < 0$

Stop if $\mathring{c} = 1$. *Otherwise go to the beginning of* Evaluation mode.

Normal factor-based merging

The final Algorithm 6.6.8 is specialized only.

Algorithm 8.6.3 (Systematic merging of normal factors)
Initial mode

- *Estimate a mixture with normal factors. The mixture estimate is described by the collection of statistics*

$$\{L_{ic;t}, D_{ic;t}, \nu_{ic;t}\}_{c\in c^*, i=1,\ldots,\mathring{d}, t\in\{0,\mathring{t}\}}.$$

  The factors with the common i are supposed to describe the same entry of $d_{i;t}$ *irrespective of the component number.*

- *Initialize the list with rows* $\rho = (i, c, \tilde{c}) \equiv i$*-th factor is common for components* $c, \tilde{c}$. *Usually, the list* ρ *is initialized as the empty one.*

- *Evaluate the individual normalization factors,* $\forall c \in c^*, i = 1,\ldots,\mathring{d}, t \in \{0, \mathring{t}\}$,

$$\mathcal{I}(L_{ic;t}, D_{ic;t}\nu_{ic:t}) = \Gamma(0.5\nu_{ic;t})\,{}^{\lfloor d}D_{ic;t}^{-0.5\nu_{ic;t}} \left|{}^{\lfloor\psi}D_{ic;t}\right|^{-0.5} 2^{0.5\nu_{ic;t}}(2\pi)^{0.5\mathring{\psi}}.$$

Evaluation mode

For $i = 1,\ldots,\mathring{d}$

Set pointers $c = 1, \tilde{c} = 2$ *to trial components to be merged.*

Test of the common structure

Set the indicator of the common structure $cs = 0$.

Set $cs = -1$ *if the structures of* θ_{ic} *and* $\theta_{i\tilde{c}}$ *differ.*

Do if $cs = 0$

Create $L'DL$ *decomposition of the trial merger*

$\tilde{L}'_{i;\mathring{t}}\tilde{D}_{i;\mathring{t}}\tilde{L}_{i;\mathring{t}} = L'_{ic;\mathring{t}}D_{ic;\mathring{t}}L_{ic;\mathring{t}} + L'_{i\tilde{c};\mathring{t}}D_{i\tilde{c};\mathring{t}}L_{i\tilde{c};\mathring{t}}$

$\tilde{L}'_{i;0}\tilde{D}_{i;0}\tilde{L}_{i;0} = L'_{ic;0}D_{ic;0}L_{ic;0} + L'_{i\tilde{c};0}D_{i\tilde{c};0}L_{i\tilde{c};0}$

using Algorithm 8.1.2 on columns of the added matrices.

Set $\tilde{\nu}_{i;\mathring{t}} = \nu_{ic;\mathring{t}} + \nu_{i\tilde{c};\mathring{t}},\ \tilde{\nu}_{i;0} = \nu_{ic;0} + \nu_{i\tilde{c};0}$.

Evaluate increment $\tilde{l}$ *of the log-v-likelihood using prepared values and formula (8.22)*

$$\tilde{l} = + \left\{ \ln(\mathcal{I}(\tilde{L}_{i;\mathring{t}}, \tilde{D}_{i;\mathring{t}}, \tilde{\nu}_{i;\mathring{t}})) - \ln(\mathcal{I}(L_{ic;\mathring{t}}, D_{ic;\mathring{t}}, \nu_{ic;\mathring{t}})) - \ln(\mathcal{I}(L_{i\tilde{c};\mathring{t}}, D_{i\tilde{c};\mathring{t}}, \nu_{i\tilde{c};\mathring{t}})) \right\}$$
$$- \left\{ \ln(\mathcal{I}(\tilde{L}_{i;0}, \tilde{D}_{i;0}, \tilde{\nu}_{i;0})) - \ln(\mathcal{I}(L_{ic;0}, D_{ic;0}, \nu_{ic;0})) - \ln(\mathcal{I}(L_{i\tilde{c};0}, D_{i\tilde{c};0}, \nu_{i\tilde{c};0})) \right\}.$$

end of the test on $cs = 0$

Do if $\tilde{l} \leq 0$ *or* $cs < 0$

Set $\tilde{c} = \tilde{c} + 1$.

Go to the Test of the common structure *if* $\tilde{c} \leq \mathring{c}$. *Otherwise continue.*

Set $c = c + 1$ *and* $\tilde{c} = c + 1$.

Go to the beginning of Test of the common structure *if* $c < \mathring{c}$.

Otherwise go to the end of cycle over i.

else replace prior and posterior estimates of factors with indices ic *and* $i\tilde{c}$ *by the trial merger.*

Extend the list of common factors by $\rho = [\rho; (i, c, \tilde{c})]$.

end of the test on improvement of v-likelihood and of $cs < 0$

end of the cycle over i

Merging of components

For $c = 1, \ldots, \mathring{c} - 1$

For $\tilde{c} = c + 1, \ldots, \mathring{c}$

Set $\kappa_{\tilde{c};\mathring{t}} = \kappa_{\tilde{c};\mathring{t}} + \kappa_{c;\mathring{t}},\quad \kappa_{\tilde{c};0} = \kappa_{\tilde{c};0} + \kappa_{c;0}$ *and cancel the component* c *if the components consist of common factors only.*

end of the cycle over $\tilde{c}$

end of the cycle over c

Component cancelling

Cancelling is based on the specialized version of Algorithm 6.6.9. For it, it is necessary to find parameter values fulfilling (6.114). It is straightforward in the normal case. The choice $\theta_{ic} = 0$, $r_{ic} = 1$, $c \in c^*$, $i = 1, \ldots, \mathring{d}$, guarantees that all parameterized factors in all components coincide.

The following values are the specific items needed in the algorithm cancelling algorithm

$$\ln\left(\frac{{}^{\lfloor \mathring{c}}f\left({}^{\lfloor \mathring{c}}\Theta_{\mathring{c}} | d(\mathring{t}) \right)}{{}^{\lfloor \mathring{c}}f\left({}^{\lfloor \mathring{c}}\Theta_{\mathring{c}} | d(0) \right)} \right) = 0.5 \sum_{i=1}^{\mathring{d}} \left({}^{\lfloor d_{i}, \mathring{c}}V_{i\mathring{c};0} - {}^{\lfloor d_{i}, \mathring{c}}V_{i\mathring{c};\mathring{t}} \right)$$

Note that normalizing factors $\mathcal{I}(\cdot)$ are treated as independent terms.

Algorithm 8.6.4 (Systematic cancelling of normal components)
Initial mode

- *Estimate of the mixture with normal factors. The mixture estimate is described by the collection of statistics* $\{L_{ic;t}, D_{ic;t}, \nu_{ic;t}\}_{c \in c^*, i=1,\ldots,\mathring{d}, t \in \{0,\mathring{t}\}}$. *The factors with the common* i *are supposed to describe the same entry of* $d_{i;t}$ *irrespective of the component number.*
- *Evaluate the individual normalization factors* $\mathcal{I}(L_{ic;t}, D_{ic;t}, \nu_{ic:t})$, $\forall c \in c^*, i = 1, \ldots, \mathring{d}, t \in \{0, \mathring{t}\}$, *using formula (8.22).*

Evaluation mode

Do while $c \leq \mathring{c}$ *and* $\mathring{c} > 1$

Set $s = s - \kappa_{c;\mathring{t}},\ s_0 = s_0 - \kappa_{c;0}$.

If $\kappa_{c;\mathring{t}} > \kappa_{c;0}$

$$l = \ln\left(\mathcal{B}(s_0, \kappa_{c;0})\right) - \ln\left(\mathcal{B}(s, \kappa_{c;\mathring{t}})\right) - \ln\left(\mathcal{B}(s - s_0, \kappa_{c;\mathring{t}} - \kappa_{c;0})\right)$$

For $i = 1, \ldots, \mathring{d}$

$$l = l + 0.5\left({}^{\lfloor d}D_{ic;0} - {}^{\lfloor d}D_{ic;\mathring{t}} + {}^{\lfloor d\psi}L'_{ic;0}\,{}^{\lfloor \psi}D_{ic;0}\,{}^{\lfloor d\psi}L_{ic;0} - {}^{\lfloor d\psi}L'_{ic;\mathring{t}}\,{}^{\lfloor \psi}D_{ic;\mathring{t}}\,{}^{\lfloor d\psi}L_{ic;\mathring{t}}\right) +$$

$$+ \ln\left(\mathcal{I}(L_{ic;0}, D_{ic;0}, \nu_{ic;0})\right) - \ln\left(\mathcal{I}(L_{ic;\mathring{t}}, D_{ic;\mathring{t}}, \nu_{ic;\mathring{t}})\right)$$

end of the cycle over i

else set $l = 1$

end of the test on increase of κ_c

If $l > 0$

Swap c *with* $\mathring{c}$ *and set* $\mathring{c} = \mathring{c} - 1$, *i.e. cancel the component*

Stop if $\mathring{c} = 1$

else set $s = s + \kappa_{c;\mathring{t}},\ s_0 = s_0 + \kappa_{c;0},\ c = c + 1$

end of the test on v-log-likelihood increase

8.7 Model validation

Section applies the general model validation presented in Section 6.7 to normal components.

8.7.1 Learning with labelled data

This part specializes Section 6.7.1.

Construction of the advisory system is simplified if learning data are qualified by experts. It provides an important additional data item, say $e \in e^* \equiv \{1, \ldots, \mathring{e}\}$, $\mathring{e} < \infty$. If the discussed labels are directly measured then they can be treated as other discrete data, i.e. modelled by Markov type factors and require no special discussion.

A different situation arises with labels that classify data ex post. In this case, a finer modelling is desirable at least because of inherently unbalanced scenario of the considered learning. It stems from the fact that any reasonably managed process leads to a few types of outcomes. Deviations from a good state are exceptional. As such, they could get too small weight or be completely overlooked when treated without taking into account their specific role. Quasi-Bayes estimation increases this danger. These labels are constant over whole data blocks and thus approximate processing-order-dependent estimation may provide quite misleading results.

In Section 6.7.1, the following hypothesis are formulated.

$H_s \equiv$ The difference in observed consequences is due to the inseparable influence of external conditions and management way.

Single mixture describes the standard $d(\mathring{t}_s)$ as well as other block of the labelled $d(\mathring{t}_e)$ data, i.e. for $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e)$

$$f(d(\mathring{t})|H_s) = \int f(d(\mathring{t})|\Theta_s) f(\Theta_s)\, d\Theta_s. \tag{8.87}$$

$H_e \equiv$ The difference in observed consequences is caused by different ways of management. Thus, different mixtures should be used for the standard $d(\mathring{t}_s)$ and the labelled $d(\mathring{t}_e)$ data, i.e. for $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$

$$f(d(\mathring{t})|H_e) = \int f(d(\mathring{t}_s)|\Theta_s) f(\Theta_s)\, d\Theta_s \int f(d(\mathring{t}_e)|\Theta_e) f(\Theta_e)\, d\Theta_e. \tag{8.88}$$

Structures of both mixtures may differ.

With these elements and no prejudice, $f(H_s) = f(H_e)$, the Bayes rule provides the posterior pf $f(H_s|d(\mathring{t}))$. The common model can be accepted as good one if this probability is high enough.

If H_s is accepted then the labelling just help in recognition of the factors active on $d(\mathring{t}_e)$ as potentially dangerous.

If H_e is accepted both models should be used separately. The factors of the predictor obtained from $f(d(\mathring{t}_s)|\Theta_s, H_s)$ near to the predictive factors obtained from $f(d(\mathring{t}_e)|\Theta_e, H_e)$ should be treated as potentially dangerous.

The conceptual algorithm is as follows.

Algorithm 8.7.1 (Test of normal data homogeneity)

1. *Run complete model estimation on standard* $d(\mathring{t}_s)$*, labelled* $d(\mathring{t}_e)$ *and concatenated* $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$ *data. This provides the posterior pdfs*
$$f(\Theta|d(\mathring{t}_\iota)) = Di_\alpha(\kappa_{\mathring{t}_\iota}) \prod_{c\in c^*} \prod_{i\in i^*} GiW_{\theta_{ic}, r_{ic}}(L_{ic;\mathring{t}_\iota}, D_{ic;\mathring{t}_\iota} \nu_{ic;\mathring{t}_\iota}), \ \iota \in \iota^* \equiv \{s, e, \emptyset\}.$$

2. *The corresponding v-likelihood values indexed by* $\iota \in \iota^*$ *are obtained as by product of approximate estimations. They have the form, see (8.40),*
$$\begin{aligned} f(d(\mathring{t}_\iota)|\iota) &= \prod_{t\in t^*_\iota} \sum_{c\in c^*} \prod_{i\in i^*} \frac{\mathcal{I}(L_{ic;t}, D_{ic;t}, \nu_{ic;t})}{\mathcal{I}(L_{ic;t-1}, D_{ic;t-1}, \nu_{ic;t-1})} \frac{\kappa_{c;t-1}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t-1}} \quad \textit{with} \\ \mathcal{I}(L_{ic;t}, D_{ic;t}, \nu_{ic;t}) &= \Gamma(0.5\nu_{ic;t})\, {}^{\lfloor d}D_{ic;t}^{-0.5\nu_{ic;t}} \left| {}^{\lfloor \psi_{ic}}D_{ic;t} \right|^{-0.5} 2^{0.5\nu_{ic;t}} (2\pi)^{0.5\mathring{\psi}_{ic}}. \end{aligned}$$

3. *Determine probability of that a single standard model should be used*
$$f(standard|d(\mathring{t})) \equiv f(H_s|d(\mathring{t})) = \frac{f(d(\mathring{t}))}{f(d(\mathring{t})) + f(d(\mathring{t}_s))f(d(\mathring{t}_e))}. \tag{8.89}$$

4. *Use the single model further on if* $f(standard|d(\mathring{t}))$ *is close to 1. The factors that were active on* $f(d(\mathring{t}_e))$ *denote as potentially dangerous.*

5. *Use both mixtures independently if* $f(standard|d(\mathring{t}))$ *is close to 0. A danger of arising the situation labelled by e should be signaled whenever the model fitted to* $d(\mathring{t}_e)$ *makes better predictions than the model fitted to the standard data.*

8.7.2 Learning results

Let us divide data available in off line mode $d(\mathring{t})$ into learning $d(\mathring{t}_l)$ and validation data sets $d(\mathring{t}_v)$. We want to test whether the model fitted on learning data suits for the validation set, too. We test the hypotheses

$H_l \equiv$ All recorded data $d(\mathring{t}) \equiv (d(\mathring{t}_l), d(\mathring{t}_v))$ are described by a single mixture model.

$H_v \equiv$ The learning data $d(\mathring{t}_l)$ set and the validation data set $d(\mathring{t}_v)$ should be described by individual models.

With no prior prejudice, the probability of the tested hypothesis can be evaluated by copying evaluations of Algorithm 8.7.1 with different interpretation and consequences of results.

Algorithm 8.7.2 (Model validation on a homogenous validation data)

1. *Run complete mixture estimations on learning* $d(\mathring{t}_l)$*, validation* $d(\mathring{t}_v)$ *and concatenated* $d(\mathring{t}) \equiv (d(\mathring{t}_l), d(\mathring{t}_v))$ *data.*

2. *Evaluate the corresponding v-likelihood* $f(d(\mathring{t}_l))$*,* $f(d(\mathring{t}_v))$*,* $f(d(\mathring{t}))$ *as in Algorithm 8.7.1.*

3. *Determine the probability of successful learning*

$$f(success|d(\mathring{t})) \equiv f(H_l|d(\mathring{t})) = \frac{f(d(\mathring{t}))}{f(d(\mathring{t})) + f(d(\mathring{t}_l))f(d(\mathring{t}_v))}. \tag{8.90}$$

4. *Accept this test as an evidence of successful learning if $f(success|d(\mathring{t}))$ is close to 1.*

5. *Continue in learning if $f(success|d(\mathring{t}))$ is close to 0. Typically, care about a correction of the mixture structure. In any case, analyze what group data made this probability small and what are their common features poorly reflected in the tested model.*

Remark(s) 8.7.1

The current experience indicates that this test is extremely sharp giving the resulting probabilities either zero or one. For a real data, some under-modelling is always expected so that the learning results will hardly pass this test. This motivates search for more "soft" alternatives, see Section 8.7.3.

8.7.3 Forgetting-based validation

Application of forgetting-based validation to normal mixtures is straightforward. We present it here for completeness only.

Algorithm 8.7.3 (Forgetting-based validation)
Initial mode

- *Estimate both parameters and structure of the normal mixture model.*
- *Apply flattening in the branching version so that good prior pdf*

$$f(\Theta) \quad = \quad Di_\alpha(\kappa_0)\prod_{c\in c^*}\prod_{i\in i^*} GiW_{\theta_{ic},r_{ic}}(L_{ic;0}, D_{i;0}, \nu_{ic;0}), \quad \textit{is obtained.}$$

- *Select several forgetting factors $0 \approx \lambda_1 < \lambda_2 < \ldots \lambda_{\mathring{i}-1} < \lambda_{\mathring{i}} = 1,\ 1 < \mathring{i} < \infty$.*
- *Set $f_i(\Theta) = f_{\lambda_i}(\Theta) = f(\Theta)$.*

Validation mode

1. *Perform estimation with the stabilized forgetting, Section 3.1 for all λ_i, using $f(\Theta|d(\mathring{t}))$ as the alternative.*

2. *Evaluate values of v-likelihood $l_{i;\mathring{t}} = f(d(\mathring{t})|\lambda_i)$ as the product of the one-step-ahead adaptive predictors (8.40). Compute MAP estimate $\hat{\lambda}$ of λ .*

3. *Take the model as successful one if $\lambda_1 = \hat{\lambda}$. Otherwise search for its improvements.*

8.7.4 Inspection by human designer

Low-dimensional projections of the estimated pdfs are computed using Proposition 8.1.8. Evaluation of the low-dimensional projections of normal predictors is described by Propositions 9.1.2, 9.1.3. Using these projections, it is possible to exploit human ability to grasp features hardly accessible in an algorithmic way.

8.7.5 Operating modes

This validation, described generally in Section 6.7.5, judge the KL distance of sufficiently probable components to the user ideal ${}^{\lfloor U}f(d(\mathring{t}))$. The (worst) best of them should coincide with those operation modes that according to the expert judgement are taken as the (worst) best ones.

We assume that the estimated mixture is precise enough – it passed successfully tests described above – when making this test. Then, we can assume that components are normal with known parameters. Thus, it makes sense to specify the user ideal as a normal pdf, too. Then, we can use Proposition 8.1.10.

Let us assume that the data record d can be split into the following parts

d_m variables determining directly markers that consist of the decisive outputs of the o-system and recognizable actions u_o of the operator,

d_e external and/or informative variables that could not or could be influenced by the operator but no specific values are required.

Often, the user ideal can be expressed in terms of d_m. We assume this case and moreover we assume the ideal has the form of a normal pdf. Then, Proposition 8.1.10 allows us to apply the following algorithm.

Algorithm 8.7.4 (Analysis of normal operating modes)

1. *Split data records d into marker-defining d_m and remaining external variables d_e.*
2. *Determine the user ideal $\prod_{i=1}^{\mathring{d}_m} \mathcal{N}_{d_i}\left({}^{\lfloor U}\theta_i' \, {}^{\lfloor U}\psi_i, \, {}^{\lfloor U}r_i\right)$ on d_m.*
3. *Take the estimated normal mixture that passed successfully learning tests.*
4. *Select a lower bound $\underline{\alpha} \in [0, 1)$ on non-negligible component weights.*
5. *Take gradually all components with weights $\alpha_c \geq \underline{\alpha}$.*
6. *Evaluate marginal pdfs $f(d_m|d(t-1))$ from the inspected component using Proposition 7.1.2 and if need be use the results of Section 9.1.1.*
7. *Evaluate the KL distance of the resulting normal components to the user ideal using Proposition 8.1.10.*
8. *Check whether the KL distances are decreasing function of quality assigned by an expert to operating modes around the respective components.*
9. *The model passes this test successfully if no visible discrepancy is recognized in the previous step. Otherwise model has to be improved.*

Remark(s) 8.7.2

1. *The algorithm can be easily extended to markers that are one-to-one known image of d_m. The corresponding substitution in the integral defining the KL distance does not change its value.*
2. *The conditional version of distance is evaluated. It gives a local view on the achieved quality. If we make the user ideal of external variables d_e equal to the pdf $f(d_e|d_m, d(t-1))$ then the overall distance*

$$\mathcal{D}\left(f|| \, {}^{\lfloor U}f\right) = \mathcal{E}\left[\sum_{t \in t^*} \int f(d_m|d(t-1)) \ln\left(\frac{f(d_m|d(t-1))}{{}^{\lfloor U}f(d_m|d(t-1))}\right) dd_m\right].$$

   Thus, the sample version coincides with sum of these local distances.
3. *This validation part can be substantially extended by using results of the design, see Chapter 9. Other tests, like comparison of the optimal recommendations and real actions, can be and will be added.*

Chapter 9

Design with normal mixtures

Normal models with known parameters and state in the phase form, in conjunction with unrestricted optimization of quadratic or exponential-quadratic loss function, lead to feasible Bellman functions, Agreement 2.4.3. For them, the optimal design reduces to the numerically tractable manipulations with quadratic forms. Here, we inspect how this property can be extended to normal mixtures. For this, we restrict ourselves to mixtures with the state in the phase form that is (almost) necessary prerequisite of the discussed tractability. Thus, we consider *normal components*

$$f(d_t|d(t-1),\Theta_c,c) = \mathcal{N}_{d_t}(\theta_c'\phi_{c;t-1}, r_c), \text{ with} \tag{9.1}$$

parameters $\Theta_c = [\theta_c, r_c]'$ consisting of <u>matrix</u> regression coefficients θ_c and covariance matrix r_c. The state vector $\phi_{c;t-1}$ has the phase form $\phi_{c;t-1} = [d'_{(t-1)\ldots(t-\partial_c)}, 1]'$, $\partial_c \geq 0$. When recognizable actions $u_{o;t}$ are at disposal, the innovations Δ_t available to the p-system are modelled by the components

$$f(\Delta_t|u_{o;t},d(t-1),\Theta_c,c) = f\left(\Delta_t|u_{o;t},\phi_{c;t-1},\, {}^{\lfloor\Delta}\Theta_c, c\right) = \mathcal{N}_{\Delta_t}\left({}^{\lfloor\Delta}\theta_c'\psi_t,\, {}^{\lfloor\Delta}r_c\right), \text{ where} \tag{9.2}$$

${}^{\lfloor\Delta}\Theta_c = \left[{}^{\lfloor\Delta}\theta_c,\, {}^{\lfloor\Delta}r_c\right]$, ${}^{\lfloor\Delta}\theta_c$ contains <u>matrix</u> regression coefficients and ${}^{\lfloor\Delta}r_c$ is the corresponding covariance matrix. The regression vector $\psi_{c;t}$ is defined $\psi'_{c;t} = \left[u'_{o;t}, d'_{(t-1)\ldots(t-\partial_c)}, 1\right] \equiv \left[u'_{o;t}, \phi'_{c;t-1}\right]$, $\partial_c \geq 0$. The recognizable actions are generated by

$$f\left(u_{o;t}|d(t-1),\, {}^{\lfloor u}\Theta_c, c\right) = f\left(u_{o;t}|\phi_{c;t-1},\, {}^{\lfloor u}\Theta_c, c\right) = \mathcal{N}_{u_{o;t}}\left({}^{\lfloor u}\theta_c'\phi_{c;t-1},\, {}^{\lfloor u}r_c\right), \text{ where} \tag{9.3}$$

${}^{\lfloor u}\Theta_c = \left[{}^{\lfloor u}\theta_c,\, {}^{\lfloor u}r_c\right]$, ${}^{\lfloor u}\theta_c$ contains <u>matrix</u> regression coefficients and ${}^{\lfloor u}r_c$ is the corresponding covariance matrix. The state vector $\phi_{c;t-1}$ is defined $\phi'_{c;t-1} = \left[d'_{(t-1)\ldots(t-\partial_c)}, 1\right]$, $\partial_c \geq 0$.

The matrices of coefficients $\theta_c \equiv \left[{}^{\lfloor\Delta}\theta_c,\, {}^{\lfloor u}\theta_c\right]$ as well as covariance matrices ${}^{\lfloor\Delta}r_c$, ${}^{\lfloor u}r_c$ are obtained from the factorized version of the normal mixture as described in Section 8.1.7. For the design, the individual regression coefficients θ_{ic} are, moreover, complemented by zeros so that the common state vector $\phi_{t-1} \equiv \phi_{c;t-1}$ is used. For reference purposes, let us fix the agreements adopted.

Agreement 9.0.1 (Design conditions) *The system is modelled by a mixture with normal components (9.1) having known parameters Θ and the state ϕ_t in the phase form. The regression coefficients are complemented by zeros so that all factors within a single component have a common state vector. In the academic design, the data record $d_t = \Delta_t \equiv$ innovations and regression vectors of individual factors are nested as follows*

$$\begin{aligned}
\psi_{i;t} &\equiv [d'_{i+1\ldots\mathring{d};t}, \phi'_{t-1}]' \equiv [d_{i+1;t}; \psi'_{i+1;t}]',\ i = 1,\ldots,\mathring{d}-1, \\
\psi_{\mathring{d};t} &\equiv \phi_{t-1} \equiv [d'_{(t-1)\ldots(t-\partial)}, 1]',\ \partial \geq 0,\quad \psi_{0;t} \equiv \Psi_t \equiv [d'_{t\ldots(t-\partial)}, 1]'.
\end{aligned} \tag{9.4}$$

In the industrial or simultaneous design, the data record $d_t = (\Delta, u_{o;t}) \equiv$(innovations, recognizable actions) and regression vectors are nested as follows

$$\begin{aligned}
\psi_{i;t} &\equiv [\Delta'_{i+1\ldots\mathring{\Delta};t}, u'_{o;t}, \phi'_{t-1}]' \equiv [\Delta_{i+1;t}, \psi'_{i+1;t}]' = \Psi_{i+1;t},\ i = 1,\ldots,\mathring{\Delta}-1, \\
\psi_{\mathring{\Delta};t} &\equiv [u'_{o;t}, \phi'_{t-1}]' \equiv [u'_{o;t}, d'_{(t-1)\ldots(t-\partial)}, 1]',\ \partial \geq 0,\quad \psi_{0;t} \equiv \Psi_t \equiv [d'_{t\ldots(t-\partial)}, 1]'.
\end{aligned} \tag{9.5}$$

Graphical expression of this nesting looks as follows.

$$\Psi_t \equiv \psi_{0;t} \left\{ \phi_{t-1} \equiv \psi_{\mathring{d};t} \equiv \left\{ \begin{bmatrix} d_t \\ --- \\ d_{t-1} \\ --- \\ \dots \\ --- \\ d_{t-\partial} \\ --- \\ 1 \end{bmatrix} \equiv \begin{bmatrix} d_{1;t} \\ d_{2;t} \\ \dots \\ d_{\mathring{d};t} \\ --- \\ d_{1;t-1} \\ d_{2;t-1} \\ \dots \\ d_{\mathring{d};t-1} \\ --- \\ \dots \\ --- \\ d_{1;t-\partial} \\ d_{2;t-\partial} \\ \dots \\ d_{\mathring{d};t-\partial} \\ --- \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \Delta_{1;t} \\ \Delta_{2;t} \\ \dots \\ \Delta_{\mathring{\Delta};t} \\ u_{o1;t} \\ u_{o2;t} \\ \dots \\ u_{o\mathring{u}_o;t} \\ --- \\ \Delta_{1;t-1} \\ \Delta_{2;t-1} \\ \dots \\ \Delta_{\mathring{\Delta};t-1} \\ u_{o1;t-1} \\ u_{o2;t-1} \\ \dots \\ u_{o\mathring{u}_o;t-1} \\ --- \\ \dots \\ --- \\ \Delta_{1;t-\partial} \\ \Delta_{2;t-\partial} \\ \dots \\ \Delta_{\mathring{\Delta};t-\partial} \\ u_{o1;t-\partial} \\ u_{o2;t-\partial} \\ \dots \\ u_{o\mathring{u}_o;t-\partial} \\ --- \\ 1 \end{bmatrix} \right. \right\} \equiv \psi_{\mathring{\Delta};t} \equiv [u'_{o;t}, \phi'_{t-1}]'. \tag{9.6}$$

$$\left. \begin{bmatrix} d_{1;t} \\ \dots \\ d_{i;t} \\ d_{i+1;t} \\ \dots \\ d_{\mathring{d};t} \\ --- \\ d_{1;t-1} \\ d_{2;t-1} \\ \dots \\ d_{\mathring{d};t-1} \\ --- \\ \dots \\ --- \\ d_{1;t-\partial} \\ d_{2;t-\partial} \\ \dots \\ d_{\mathring{d};t-\partial} \\ --- \\ 1 \end{bmatrix} \right\} \equiv \psi_{i;t} \equiv \begin{bmatrix} d_{i+1;t} \\ --- \\ \psi_{i+1;t} \end{bmatrix} \equiv \begin{bmatrix} d_{i+1;t} \\ \dots \\ d_{\mathring{d};t} \\ --- \\ \phi_{t-1} \end{bmatrix}. \tag{9.7}$$

Chapter starts with common tools, Section 9.1, that cover model projections, discussion of practical presentation aspects, evaluations of feasible upper bounds of a Jensen type on the KL distance and evaluation of expected quadratic forms. Then, academic, industrial and simultaneous designs of the advisory system are elaborated in detail, Section 9.2. Designs of presentation and signaling strategies conclude Chapter, Section 9.3.

9.1 Common tools

9.1.1 Model projections in design

Steady state pdfs

Steady state pdfs of the observed data are needed for recognizing dangerous components, Agreement 5.4.6. The large amount of data available justifies the assumption that the uncertainty of parameters is negligible after estimation. It implies that we can evaluate steady-state pdf of data assuming that the parameters $\Theta = [\theta, r]$ characterizing the normal component (9.1) with

$$\text{matrix regression coefficients } \theta' \equiv [\tilde{\theta}', \mu] \equiv [A_1, \ldots, A_{\partial}, \mu],\ A_i \text{ denotes square matrix, and}$$
$$\text{noise covariance } r = L_e' D_e L_e \tag{9.8}$$

are known. The column $\mathring{d}$-vector μ in θ describes the data offset. The modelled o-system is supposed to be stable without a significant loss of generality. Thus, a good model has to have A's describing a stable auto-regression.

The steady state pdf of the state ϕ is normal as the relationships among the observed data are linear and the driving white noise is normal. Thus, it is sufficient to evaluate the steady state expected value $\phi_\infty \equiv \mu_\infty \otimes \mathbf{1}_\partial$ and the steady state covariance matrix $C_\infty = L_\infty' D_\infty L_\infty$. Recall that $\otimes$ denotes Kronecker product and $\mathbf{1}_\partial$ is the ∂-vector of units.

The value μ_∞ can be computed in the iterative way. It is the limiting value of the sequence $\{\mu_n\}_{n=1-\partial}^{\infty}$. The sequence starts from zero initial conditions $\mu_n = 0$, for $n \le 0$, and its other members are generated as follows

$$\mu_n = \sum_{i=1}^{\partial} A_i \mu_{n-i} + \mu. \tag{9.9}$$

This evaluation is preferable against an explicit solution. Convergence of (9.9) tests stability of the estimated auto-regressive model, it tests whether the component is dangerous.

The $L'DL$ decomposition of the steady state covariance C_∞ of ϕ without constant entry 1 can be found also in the iterative way

$$L_{n+1}' D_{n+1} L_{n+1} = \begin{bmatrix} L_n \tilde{\theta} & \tilde{L}_n \end{bmatrix}' D_n \begin{bmatrix} L_n \tilde{\theta} & \tilde{L}_n \end{bmatrix} + [L_e\ 0]' D_e [L_e\ 0], \quad \text{where} \tag{9.10}$$

$L_1 = D_1 = I =$ unit matrix, $\tilde{\theta}$ means θ with the offset μ omitted and $\tilde{L}_n$ coincides with initial $\mathring{d}(\partial - 1)$ columns of L_n. We summarize the overall algorithm.

Algorithm 9.1.1 (Moments of the steady state normal pdf)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 0$.
- *Select a small* $\varepsilon > 0$ *used for stopping.*
- *Set* $[\mu_{-1}, \ldots, \mu_{-\partial}] = 0$.
- *Set* $L_n = D_n = I = (\mathring{d}\partial, \mathring{d}\partial)$*-unit matrix.*

Iterative mode (expectation)

1. *Update* $\mu_n = \sum_{i=1}^{\partial} A_i \mu_{n-i} + \mu$.

2. *Stop completely and announce instability of the model (label the component as dangerous one) if* $||\mu_n|| > 1/\varepsilon$ *for some norm* $||\cdot||$.

3. *Set* $n = 0$ *and go to the beginning of* Iterative mode (covariance) *if* $n > \mathring{n}$ *or* $||\mu_n - \mu_{n-1}|| < \varepsilon$.

4. *Increase the counter* $n = n + 1$ *and go to the beginning of* Iterative mode (expectation).

Iterative mode (covariance)

1. *Make the updating (9.10) using a variant of Algorithm 8.1.2.*

2. *Stop if* $n > \mathring{n}$ *or* $||D_n - D_{n-1}|| < \varepsilon$ *and take* μ_n *and* $L_n' D_n L_n$ *as steady-state moments.*

3. *Increase the counter* $n = n + 1$ *and go to the beginning of* Iterative mode (covariance).

The non-stable components are obvious candidates for dangerous ones but those with stationary moments far from the target area may be classified as dangerous, too, Agreement 5.4.6.

Remark(s) 9.1.1

Presence of non-stable components does not automatically mean that the mixture with such components is unstable. The following simple example indicates it. Let the scalar d_t *be generated by the normal two-component mixture*

$$f(d_t|d(t-1)) = \alpha \mathcal{N}_{dt}(\theta_1 d_{t-1}, 1) + (1-\alpha)\mathcal{N}_{dt}(\theta_2 d_{t-1}, 1).$$

Then, $r_t = \mathcal{E}[d_t^2] \geq 0$ *evolves according to the recursion* $r_t = \underbrace{[\alpha\theta_1^2 + (1-\alpha)\theta_2^2]}_{\lfloor m \theta^2}r_{t-1} + 1$ *and has the finite solution if* ${}^{\lfloor m}\theta < 1$. *This condition is met even for combinations of stable and non-stable components, for instance,* $\alpha = 0.8$, $\theta_1 = 1.1$ *(!) and* $\theta_2 = 0.3$ *giving* ${}^{\lfloor m}\theta = 0.993$. *With it, the second and consequently first moments are finite even for* $t \to \infty$.

The conclusion can be expressed in the appealing form presence of, even rarely active, stable component has stabilizing effect on the overall mixture.

Marginal and conditional pdfs

Low dimensional marginal and conditional pdfs provide the main technical tool for presenting results of the design of the p-system. The algorithmic description of their evaluation is given here for a known (well estimated) normal mixture.

In the industrial design with quadratic loss function and in simultaneous design, the pdf of the optimal recognizable actions is also normal. Thus, the results of this Section serve to all versions of advisory systems. Essentially, Proposition 7.1.2 is specialized here. The specialization is split in Proposition 9.1.2 dealing with marginal pdfs and Proposition 9.1.3 addressing the conditioning. Both of them rely on a specific structure of components, Agreement 5.2.1. Thus, we need a tool for changing it.

Proposition 9.1.1 (Marginal pdf of adjacent normal factors) *Let us consider a pair of* adjacent normal factors, *Agreement 5.2.1, with known parameters*

$$f(\Delta_1, \Delta_2|\psi_1, \psi_2) = \mathcal{N}_{\Delta_1}\left([\beta, \theta_1'][\Delta_2, \psi_1']', r_1\right)\mathcal{N}_{\Delta_2}\left(\theta_2'\psi_2', r_2\right). \tag{9.11}$$

Then,

$$f(\Delta_1, \Delta_2|\psi_1, \psi_2) = \mathcal{N}_{\Delta_2}\left([\tilde{\beta}, \tilde{\theta}_2'][\Delta_1, \psi']', \tilde{r}_2\right)\mathcal{N}_{\Delta_1}\left(\tilde{\theta}_1'\psi, \tilde{r}_1\right), \quad where$$

ψ *contains union of entries in* ψ_1, ψ_2. *The entry, which is at least in one of them, is put on the corresponding position of* ψ.

The obtained normal factor for Δ_1 *in the second version coincides with the marginal pdf of* Δ_1. *The quantities marked by* ˜ *are generated by the following algorithm.*

- *Extend vectors* θ_i *by zeros to the vectors* $\bar{\theta}_i$ *of a common length so that* $[\beta, \theta_1'][\Delta_2, \psi_1']' = [\beta, \bar{\theta}_1'][\Delta_2, \psi']'$ *and* $\theta_2'\psi_2 = \bar{\theta}_2'\psi$.

- *Permute the entries* Δ_1, Δ_2 *in the quadratic form* Q

$$Q \equiv [\Delta_1, \Delta_2, \psi'] \begin{bmatrix} -1 & \beta & \bar{\theta}_1 \\ 0 & -1 & \bar{\theta}_2 \end{bmatrix}' \mathrm{diag}[r_1^{-1}, r_2^{-1}] \begin{bmatrix} -1 & \beta & \bar{\theta}_1 \\ 0 & -1 & \bar{\theta}_2 \end{bmatrix} [\Delta_1, \Delta_2, \psi']' =$$

$$[\Delta_2, \Delta_1, \psi'] \begin{bmatrix} 0 & -1 & \bar{\theta}_2 \\ -1 & \beta & \bar{\theta}_1 \end{bmatrix}' \mathrm{diag}[r_1^{-1}, r_2^{-1}] \begin{bmatrix} 0 & -1 & \bar{\theta}_2 \\ -1 & \beta & \bar{\theta}_1 \end{bmatrix} [\Delta_2, \Delta_1, \psi']'$$

and recover $L'DL$ *decomposition with -1 (!) on the diagonal of* L *of its kernel using Proposition 8.1.4*

$$Q = [\Delta_2, \Delta_1, \psi'] \begin{bmatrix} -1 & \tilde{\beta} & \tilde{\theta}_2 \\ 0 & -1 & \tilde{\theta}_1 \end{bmatrix}' \mathrm{diag}[\tilde{r}_2^{-1}, \tilde{r}_1^{-1}] \begin{bmatrix} -1 & \tilde{\beta} & \tilde{\theta}_2 \\ 0 & -1 & \tilde{\theta}_1 \end{bmatrix} [\Delta_2, \Delta_1, \psi']'.$$

Proof: Omitted. □

Any desired change of the component structure can be achieved by a sequence of pair-wise *permutations on adjacent factors.* If the component consists of normal factors only then it can be written in the matrix form (9.1) and the discussed permutation can be made on it.

Proposition 9.1.2 (Marginal predictors for known normal mixtures) *Under Agreement 9.0.1, let us consider the known normal mixture model in the factorized form*

$$f(\Delta_t | u_{o;t}, d(t-1)) = \sum_{c \in c^*} \alpha_c \prod_{i \in i^*} \mathcal{N}_{\Delta_{ic;t}}(\theta'_{ic}\psi_{ic;t}, r_{ic}), \quad \textit{where}$$

$\psi_{ic;t} = [\Delta'_{(i+1)\ldots\mathring{\Delta}c;t}, u'_{o;t}, \phi'_{c;t-1}]' \equiv [\Delta'_{(i+1)\ldots\mathring{\Delta}c;t}, \psi'_{\mathring{\Delta}c;t}]'$ *are regression vectors,* $i \in i^* \equiv \{1, \ldots, \mathring{\Delta} - 1\}$, $c \in c^*$,

$\psi'_{\mathring{\Delta}c;t}$ *is the common part of regression vectors forming the component* c,

$\Delta_{ic;t}$ *are entries of innovations* $\Delta_{c;t}$ *and* $u_{o;t}$ *are recognizable o-actions,*

θ_{ic} *are regression coefficients, with entries ordered accordingly to the corresponding regression vector* ψ_{ic}, *i.e.* $\theta_{ic} = \left[\theta_{1ic}, \ldots, \theta_{(\mathring{\Delta}-i)ic}, {}^{\lfloor\psi}\theta'_{ic}\right]'$, *where* θ_{jic}, $j = 1, \ldots, \mathring{\Delta}-i$, *are scalars and* ${}^{\lfloor\psi}\mathring{\theta}_{ic} \equiv \mathring{\psi}_{\mathring{\Delta}c;t}$; *zeros are inserted into the coefficients* θ_{ic} *in order to get the common regression vectors* $\psi_{\mathring{\Delta}c;t} = \psi_{\mathring{\Delta};t}$,

r_{ic} *is the noise variance of the corresponding factor.*

The considered normal component is determined by the matrices G_c, H_c *and* F_c, *Proposition 8.1.13,*

$$F_c \equiv \mathrm{diag}\left[r_{1c}^{-1}, \ldots, r_{\mathring{\Delta}c}^{-1}\right] \tag{9.12}$$

$$\left[G_c, \vdots H_c\right] \equiv \left[\begin{array}{c} \underbrace{\begin{array}{cccccc} -1 & \theta_{11c} & \theta_{21c} & \ldots & \theta_{(\mathring{\Delta}-1)1c} \\ 0 & -1 & \theta_{12c} & \ldots & \theta_{(\mathring{\Delta}-2)2c} \\ & & \ddots & \ddots & \\ 0 & 0 & \ldots & -1 & \theta_{1(\mathring{\Delta}-1)c} \\ 0 & 0 & \ldots & 0 & -1 \end{array}}_{G_c} \end{array} \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{array} \underbrace{\begin{array}{c} {}^{\lfloor\psi}\theta'_{1c} \\ {}^{\lfloor\psi}\theta'_{2c} \\ \vdots \\ {}^{\lfloor\psi}\theta'_{(\mathring{\Delta}-1)c} \\ {}^{\lfloor\psi}\theta'_{\mathring{\Delta}c} \end{array}}_{H_c} \right], \ G_c \textit{ is square matrix.}$$

Let the marginal pdf be computed for the selected entries of Δ_t, *say,* $\Delta_{j;t}$, $j \in j^* \equiv \{k_1, \ldots, k_{\mathring{j}}\} \subset i^* \equiv j^* \cup \overline{j^*}$, $j^* \cap \overline{j^*} = \emptyset$, *i.e. the marginal pdf of entries* $\Delta_{j^*;t} \equiv [\Delta_{k_1;t}, \ldots, \Delta_{k_{\mathring{j}};t}]'$ *is computed. Let the permutation matrix* $\mathcal{T}'_c$ *permute* $\Delta_{c;t}$ *so that the entries* $\Delta_{j^*;t}$ *are placed at the end of the permuted vector of innovations, i.e.* $\mathcal{T}'_c\Delta_{c;t} = \left[\bullet, \Delta'_{j^*;t}\right]'$.

Let us transform matrices F_c and $[G_c\mathcal{T}_c,\, H_c]$ to $\tilde{F}_c$ and $[\tilde{G}_c,\, \tilde{H}_c]$ so that $\tilde{F}_c$ is diagonal matrix, $\tilde{G}_c$ has the same form as G_c and the following quadratic form is preserved

$$[G_c\mathcal{T}_c,\, H_c]' F_c [G_c\mathcal{T}_c,\, H_c] \quad = \quad [\tilde{G}_c,\, \tilde{H}_c]' \tilde{F}_c [\tilde{G}_c,\, \tilde{H}_c]. \tag{9.13}$$

Then, the predictor of the variables of interest is the mixture

$$f(\Delta_{j^*;t}|u_{o;t}, d(t-1)) \quad \equiv \quad f(\Delta_{k_1;t}, \ldots, \Delta_{k_{\mathring{j}};t}|u_{o;t}, d(t-1)) = \sum_{c \in c^*} \alpha_c \prod_{j=1}^{\mathring{j}} \mathcal{N}_{\Delta_{k_j;t}}(\theta'_{k_j c}\psi_{k_j c;t}, r_{k_j c}), \ \textit{where}$$

the regression coefficients $\theta'_{k_j c}$ are in the last $\mathring{j}$ rows of $[\tilde{G}_c,\, \tilde{H}_c]$ (with -1 omitted) and variance inversions $r^{-1}_{k_j c}$ are placed at the corresponding positions in $\tilde{F}_c$.

Proof: Let us take a fixed component c. Then, for $\Delta_{c;t} = [\Delta_{1c;t}, \ldots, \Delta_{\mathring{\Delta}c;t}]'$, its pdf is

$$f(\Delta_{c;t}|u_{o;t}, d(t-1), c) = \prod_{i \in i^*} \mathcal{N}_{\Delta_{ic;t}}(\theta'_{ic}\psi_{ic;t}, r_{ic}) \propto$$

$$\exp\left\{ -\frac{X'}{2} \underbrace{\begin{bmatrix} r^{-1}_{1c} & & & \\ & r^{-1}_{2c} & & \\ & & \ddots & \\ & & & r^{-1}_{\mathring{\Delta}c} \end{bmatrix}}_{F_c} \underbrace{\begin{bmatrix} -1 & \theta_{11c} & \theta_{21c} & \ldots & \vdots & \theta_{(\mathring{\Delta}-1)1c} & {}^{\lfloor\psi}\theta'_{1c} \\ 0 & -1 & \theta_{12c} & \ldots & \vdots & \theta_{(\mathring{\Delta}-2)2c} & {}^{\lfloor\psi}\theta'_{2c} \\ & & \ddots & \ddots & & \vdots & \vdots \\ 0 & 0 & \ldots & -1 & \vdots & \theta_{1(\mathring{\Delta}-1)c} & {}^{\lfloor\psi}\theta'_{(\mathring{\Delta}-1)c} \\ 0 & 0 & \ldots & 0 & -1 \ \vdots & & {}^{\lfloor\psi}\theta'_{\mathring{\Delta}c} \end{bmatrix}}_{X} \begin{bmatrix} \Delta_{c;t} \\ \psi_{\mathring{\Delta};t} \end{bmatrix} \right\} \equiv$$

$$\equiv \quad \exp\left\{ -0.5[\Delta'_{c;t}, \psi'_{\mathring{\Delta};t}] \begin{bmatrix} G'_c \\ H'_c \end{bmatrix} F_c [G_c \ H_c] \begin{bmatrix} \Delta_{c;t} \\ \psi_{\mathring{\Delta};t} \end{bmatrix} \right\} \equiv \exp\{-0.5Q\}, \ \text{where}$$

F_c is diagonal $(\mathring{d}, \mathring{d})$-matrix, G_c is upper triangular matrix with -1 on its diagonal. The rectangular $(\mathring{\Delta}, \mathring{\psi}_{\mathring{\Delta}c})$ matrix H_c contains regression coefficients corresponding to the common part $\psi_{\mathring{\Delta};t}$ of regression vectors.

Let J' be a permutation matrix, which places entries $\Delta_{j^*;t}$ whose marginal pdfs should be computed to the end of the transformed vector of innovations Δ_t. Taking another matrix T_c that performs the permutation $T_c\Delta_t = \Delta_{c;t} \Leftrightarrow \Delta_t = T'_c\Delta_{c;t}$, the following equality is obtained

$$[\bullet, \Delta'_{j^*;t}]' \equiv J'\Delta_t = J'T'_c\Delta_{c;t} \equiv \mathcal{T}'_c\Delta_{c;t}.$$

Orthogonality of permutation matrices implies that the quadratic form Q in the exponent of the normal component can be expressed

$$Q = \left[[\bullet, \Delta'_{j^*;t}], \psi'_{\mathring{\Delta}c;t}\right] [G_c\mathcal{T}_c,\, H_c]' F_c [G_c\mathcal{T}_c,\, H_c] \left[[\bullet, \Delta'_{j^*;t}], \psi'_{\mathring{\Delta}c;t}\right]'.$$

Using non-uniqueness of the kernel decomposition, we find diagonal $\tilde{F}_c$, upper triangular $\tilde{G}_c$ and rectangular $\tilde{H}_c$ so that $[G_c\mathcal{T}_c,\, H_c]'F_c[G_c\mathcal{T}_c,\, H_c] = [\tilde{G}_c,\, \tilde{H}_c]'\tilde{F}_c[\tilde{G}_c,\, \tilde{H}_c]$. In algorithmic terms, LDL' decomposition (with -1 on diagonal L) is re-created. With it, the integration of superfluous variables reduces to the omission of leading rows and columns of the matrices $\tilde{F}_c$, $\tilde{G}_c$ together with leading rows of $\tilde{H}_c$. □

Proposition 9.1.3 (Conditional pdfs of normal mixture model) *Under Agreement 9.0.1, let us consider the known normal mixture model in the factorized form*

$$f(\Delta_t|u_{o;t}, d(t-1)) = \sum_{c \in c^*} \alpha_c \prod_{i \in i^*} \mathcal{N}_{\Delta_{ic;t}}(\theta'_{ic}\psi_{ic;t}, r_{ic}), \quad \textit{where}$$

$\psi_{ic;t} = [\Delta'_{(i+1)\ldots\mathring{\Delta}c;t}, u'_{o;t}, \phi'_{c;t-1}]' \equiv [\Delta_{(i+1)c;t}, \psi'_{(i+1)c;t}]'$ *are regression vectors,* $i \in \{1,\ldots,\mathring{\Delta}-1\}$, $c \in c^*$, *with the common part* $\psi_{\mathring{\Delta};t} \equiv [u'_{o;t}, \phi'_{t-1}]'$,

$\Delta_{ic;t}$ *are entries of innovations* $\Delta_{c;t}$, $u_{o;t}$ *are recognizable o-actions and* $\phi_{t-1} \equiv \phi_{c;t-1}$ *is the common observable state of individual components,*

$\theta_{ic} = \left[\theta_{1ic},\ldots,\theta_{(\mathring{\Delta}-i)ic}, {}^{\lfloor\psi}\theta'_{ic}\right]'$, ${}^{\lfloor\psi}\mathring{\theta}_{ic} \equiv \mathring{\psi}_{\mathring{\Delta}c;t}$, *are regression coefficients split in accordance with the corresponding regression vector; if need be, zeros are inserted to get the common part* $\psi'_{\mathring{\Delta};t}$,

r_{ic} *is variance of noise of the factor ic.*

Let, for simplicity, the order of factors for all components be common. In other words, common structure of components Agreement 5.2.1 implying $\Delta_{c;t} = \Delta_t$ *is considered. Suppose, that* $j \in j^* \equiv \{\underline{j},\ldots,\mathring{i}\}$, $\underline{j} \le \mathring{i}$, *point to selected entries of* Δ_t. *Their marginal pdfs are already computed, see Proposition 9.1.2, while index* $\underline{k} \in (\underline{j},\mathring{i})$ *determines the splitting of* $\Delta_{j^*;t}$ *into two parts.*

Then, the predictor of $\Delta_{\underline{j}\ldots(\underline{k}-1);t}$ *conditioned on* $u_{o;t}, d(t-1)$ *and the remaining* $\Delta_{\underline{k}\ldots\mathring{i};t}$ *is the* ratio of normal mixtures

$$f\left(\Delta_{\underline{j}\ldots(\underline{k}-1);t}|\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)\right) = \frac{\sum_{c\in c^*}\alpha_c\prod_{j=\underline{j}}^{\mathring{i}}\mathcal{N}_{\Delta_{j;t}}(\theta'_{jc}\psi_{j;c}, r_{jc})}{\sum_{c\in c^*}\alpha_c\prod_{j=\underline{k}}^{\mathring{i}}\mathcal{N}_{\Delta_{j;t}}(\theta'_{jc}\psi_{j;c}, r_{jc})}. \tag{9.14}$$

When all values in the condition are fixed, the resulting predictor is the normal mixture

$$f\left(\Delta_{\underline{j}\ldots(\underline{k}-1);t}|\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)\right) = \sum_{c\in c^*}\tilde{\alpha}_c(\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1))\prod_{j=\underline{j}}^{\underline{k}-1}\mathcal{N}_{\Delta_{j;t}}(\theta'_{jc}\psi_{j;c}, r_{jc}), \text{ with}$$

$$\tilde{\alpha}_c\left(\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)\right) \propto \alpha_c\prod_{j=\underline{k}}^{\mathring{i}}\mathcal{N}_{\Delta_{j;t}}(\theta'_{jc}\psi_{j;c}, r_{jc}). \tag{9.15}$$

Proof: It is implied by Proposition 9.1.2 and by the formula $f(\beta|\gamma) = \frac{f(\beta,\gamma)}{f(\gamma)} = \frac{f(\beta,\gamma)}{\int f(\beta,\gamma)\,d\beta}$. □

Remark(s) 9.1.2

1. *The dependence of weights (9.15) on data indicates that the model obtained through the estimation of the full predictor followed by marginalization is richer than the directly estimated, low dimensional predictor. This very practical observation is easily overlooked.*
2. *Permutations of adjacent factors according to Proposition 9.1.1 have to be made if the considered entries are not at the assumed positions. This need also implies that the assumption of the c-invariant ordering of* $\Delta_{c;t} = \Delta_t$ *is not restrictive: we have to reach this situation anyway, at least for the innovation entries of interest.*

9.1.2 Dynamic predictors in advising

Here, general results of Section 7.1.2 are specialized to normal mixtures. For them, individual components are multi-variate auto-regression models (AR). Their stability is the key property we need for good long-horizon predictors.

Stable auto-regression

Normal multi-variate AR models serve well as short-horizon predictors in a range of applications including those covered by the presented project. They are often unsuitable for long-term predictions as the dynamic system represented by estimated coefficients is too close to the stability boundary or even unstable.

Data preprocessing, Chapter 6, estimation of underlying continuous-time-model parameter [86, 87] or use models with moving average part (ARMA, [88]) offer possible remedy. None of them, however, can guarantee a full success. Then, the reduction of the support of prior pdf on the set, cf. (9.8),

$$\Theta_s^* = \left\{ ([A_1, \ldots, A_\partial, \mu], r)\,,\ \mathcal{A} = \begin{bmatrix} A_1 & \ldots & A_{\partial-1} & A_\partial \\ I & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & I & 0 \end{bmatrix} \text{has } \textit{spectral radius } < 1 \right\} \tag{9.16}$$

as discussed in Section 7.1.2, is the only possibility. Technically, Monte Carlo evaluation of the expectation over the set of stable AR models is applied. The necessary samples of Θ are generated from the estimated factorized version of the posterior pdf

$$f(\Theta|d(\mathring{t})) \propto \prod_{i=1}^{\mathring{d}} GiW_{\theta_i, r_i}\left(L_{i;\mathring{t}}, D_{i;\mathring{t}}, \nu_{i;\mathring{t}}\right) \chi_{\Theta_s^*}(\Theta). \tag{9.17}$$

The involved statistics are updated irrespective of the restriction used. For the approximate mixture estimation 6.5, weights of individual data vectors should be modified comparing to the unrestricted case. They are formed, however, by one-step-ahead predictors that are weakly influenced by the information about stability. Thus, this dependence can be and will be neglected. Consequently, the evaluation of the posterior pdf is uninfluenced by the stability restriction even for mixtures. Having the posterior pdf, we need to evaluate a point estimate $\hat{\Theta}$ of the parameter Θ. The expected value

$$\hat{\Theta} \equiv \mathcal{E}\left[\Theta|d(\mathring{t})\right] = \frac{\int \Theta \chi_{\Theta_s^*}(\Theta) \prod_{i=1}^{\mathring{d}} GiW_{\theta_i, r_i}\left(L_{i;\mathring{t}}, D_{i;\mathring{t}}, \nu_{i;\mathring{t}}\right) d\Theta}{\int \chi_{\Theta_s^*}(\Theta) \prod_{i=1}^{\mathring{d}} GiW_{\theta_i, r_i}\left(L_{i;\mathring{t}}, D_{i;\mathring{t}}, \nu_{i;\mathring{t}}\right) d\Theta} \tag{9.18}$$

is the required estimate. Its analytical evaluation is impossible. A straightforward Monte Carlo evaluation can be used. It has to take into account the non-convex shape of the integration domain. It gives the following algorithm that specializes Algorithm 7.1.1 to a normal component. For a mixture, it is applied component-wise.

Algorithm 9.1.2 (Point estimation of stable AR systems)
Initial mode

- *Perform the standard estimation, Chapter 8, giving the collection of statistics* $L_{i;\mathring{t}}, D_{i;\mathring{t}}, \nu_{i;\mathring{t}},\ i = 1, \ldots, \mathring{d}$.
- *Convert the collected statistics into their least-squares (LS) counterparts*

$$\begin{aligned} \hat{\theta}_{i;\mathring{t}} &\equiv {}^{\lfloor\psi}L_{i;\mathring{t}}^{-1}\, {}^{\lfloor d\psi}L_{i;\mathring{t}} \equiv \textit{LS estimate of regression coefficients} \\ \hat{r}_{i;\mathring{t}} &\equiv \frac{{}^{\lfloor d}D_{i;\mathring{t}}}{\nu_{i;\mathring{t}} - 2} \equiv \textit{LS estimate of noise variance} \\ C_{i;\mathring{t}} &\equiv {}^{\lfloor\psi}L_{i;\mathring{t}}^{-1}\, {}^{\lfloor\psi}D_{i;\mathring{t}}^{-1} \left({}^{\lfloor\psi}L'_{i;\mathring{t}}\right)^{-1} \equiv \textit{LS covariance factor.} \end{aligned}$$

- *Stop and take the unrestricted LS point estimates as the approximation of (9.18) if they define a stable AR model: if they belong to* Θ_s^* *(9.16).*
- *Select the upper bound* $\mathring{n}$ *(of the order 1e4 - 1e6) on the number* n *of iterations and set* $n = 0$.
- *Select the bound* $\mathring{m}$ *(of the order 1e1 - 1e3) on the necessary number* m *of stable samples and set* $m = 0$.
- *Initialize the generated stable point estimates* $\hat{\theta}_{is} = 0,\ \hat{r}_{is} = 0,\ i = 1, \ldots, \mathring{d}$.

Iterative mode

1. *Do while* $n < \mathring{n}$ & $m < \mathring{m}$.
2. *Set* $n = n + 1$.
3. *Generate, for* $i = 1, \ldots \mathring{d}$, *random samples* $\tilde{\theta}_i,\ \tilde{r}_i \sim GiW_{\theta_i, r_i}\left(\hat{\theta}_{i;\mathring{t}}, \hat{r}_{i;\mathring{t}}, C_{i;\mathring{t}}, \nu_{i;\mathring{t}}\right)$ *as follows*

$$\begin{aligned}
\tilde{r}_i &= \hat{r}_{i;\mathring{t}}(1+e_r),\ e_r \sim \mathcal{N}_{e_r}\left(0, \sqrt{\frac{2}{\nu_{i;\mathring{t}} - 4}}\right),\ cf.\ (8.23),\\
\tilde{r}_i &= \hat{r}_{i;\mathring{t}} \text{ if the sample } \tilde{r}_i \leq 0\\
\tilde{\theta}_i &= \hat{\theta}_{i;\mathring{t}} + \sqrt{\tilde{r}_i}\, {}^{\lfloor\psi}L^{-1}_{i;\mathring{t}}\, {}^{\lfloor\psi}D^{-0.5}_{i;\mathring{t}} e,\quad e \sim \mathcal{N}_e(0, I_{\mathring{\theta}_i}).
\end{aligned}$$

The normal, moments-preserving, approximation of the marginal pdf $f(r_i|d(\mathring{t}))$ *is adopted above.*

4. *Go to the beginning of* Iterative mode *if* $\tilde{\Theta} \notin \Theta^*_s$, *if the generated AR model, given by* $\tilde{\theta}_i,\ i = 1, \ldots, \mathring{d}$, *is unstable.*
5. *Set* $\tilde{m} = m + 1$ *and* $\tilde{\theta}_{is} = \hat{\theta}_{is} + \frac{1}{\tilde{m}}\left(\hat{\theta}_{is} - \tilde{\theta}_i\right)$, $\tilde{r}_{is} = \hat{r}_{is} + \frac{1}{\tilde{m}}\left(\hat{r}_{is} - \tilde{r}_i\right)$, $i = 1, \ldots, \mathring{d}$.
6. *Set* $m = \tilde{m}$ *and* $\hat{\Theta}_s = \tilde{\Theta}_s$ *if* $\tilde{\Theta}_s \in \Theta^*_s$, *if* $\tilde{\theta}_{is},\ i = 1, \ldots, \mathring{d}$, *define a stable AR model.*
7. *Go to the beginning of* Iterative mode.

Accept $\hat{\Theta}_s$ *as the final estimate if* $m = \mathring{m}$. *Otherwise take the search as unsuccessful.*

Remark(s) 9.1.3

1. *More sophisticated stopping rules can be created using simple estimates of covariance matrices of the constructed expected value and inequality of the Chebyshev type, [89].*
2. *It is useful to remember that* Θ^*_s *may be restricted by another prior knowledge, for instance, by an expected range of static gains.*

The restricted set Θ^*_s is much smaller than Θ^*. Consequently, the number of discarded samples is sometimes prohibitive. This makes us to design an alternative sampling strategy. Essentially, samples are taken from Θ^*_s and the expected value is computed as their weighted mean value. The weights express the degree how much the given sample can be attributed to the pdf $f(\Theta|d(\mathring{t}))$. The need to preserve structure of individual factors is the key obstacle faced.

Three steps have to be discussed to design such an improved sampling strategy. They use more or less text-book results. Their presentation complexity stems from the effort to stay with all evaluations at factor level.

1. Let us inspect the eigenstructure of the state matrix

$$\mathcal{A} \equiv \begin{bmatrix} A_1 & A_2 & \ldots & A_{\partial-1} & A_\partial \\ I & 0 & \ldots & 0 & 0 \\ 0 & I & \ldots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \ldots & I & 0 \end{bmatrix}$$

corresponding to multi-variate AR model without offset. Let ρ be an eigenvalue of $\mathcal{A}$ corresponding to an eigenvector ϕ, i.e. $\mathcal{A}\phi = \rho\phi$. Let us split ϕ into ∂ $\mathring{d}$-blocks $\phi' = [\phi'_1, \ldots, \phi'_\partial]$ corresponding to the size $\mathring{d}$ of data records d_t, i.e. to the size of the multi-variate regression coefficients A_i. The blocks ϕ_i are related by the recursion $\phi_i \rho = \phi_{i-1}$ valid for $i > 1$. This recursion and the identity for the block ϕ_1 give, for $\rho \neq 0$ of interest,

$$\begin{aligned}
\phi_i &= \rho^{-i+1}\phi_1,\ i = 2,\ldots,\partial, \text{ and } \rho\phi_1 = \sum_{i=1}^{\partial} A_i\phi_i = \sum_{i=1}^{\partial} A_i\rho^{-i+1}\phi_1 \quad \Rightarrow \\
0 &= \left(I - \sum_{i=1}^{\partial} A_i\rho^{-i}\right)\phi_1 \ \Rightarrow \rho \text{ is root of } \left|I - \sum_{i=1}^{\partial} A_i\rho^{-i}\right| = 0. \qquad (9.19)
\end{aligned}$$

Let us consider the <u>modified</u> multi-variate AR model with matrix coefficients $\tilde{A}_i = x^i A_i,\ i = 1,\ldots,\partial$, where x is an optional positive scalar. Obviously, if ρ is and an eigenvalue corresponding to the coefficients A_i then $x\rho$ is eigenvalue corresponding to the coefficients $\tilde{A}_i$. Thus, if ρ is eigenvalue with the largest module greater than one then the AR model modified by the factor $x = g/|\rho|$, with an arbitrary $g \in (0,1)$, is stable.

2. Let us consider iterative process, $n \in n^* \equiv \{1,2,\ldots,\mathring{n}\}$

$$\begin{aligned}
U\tilde{\phi}_n &\equiv \sum_{i=1}^{\partial} UA_i\tilde{\phi}_{n-i} \qquad (9.20) \\
U &\equiv \text{an upper triangular matrix with unit diagonal} \\
\tilde{\phi}_n &\equiv \mathbf{1}_{\mathring{d}} \equiv [1,\ldots,1]' \text{ for } n = 0,-1,\ldots,\partial-1.
\end{aligned}$$

Note that the inclusion of the upper triangular matrix U allow us to evaluate spectral radius iteratively using the plain factorized form of the AR model.

The solution has the generic form

$$\tilde{\phi}_n = \sum_{k=1}^{\partial} \rho_k^n x_k = \sum_{k=1}^{l-1} \rho_1^n \exp(i\beta_k n)x_k + \sum_{k=l}^{\partial} \rho_1^n \left(\frac{\rho_k}{\rho_1}\right)^n x_k, \text{ where}$$

i in exponent denotes here complex unit, ρ_k are roots of (9.19) and β_k differences in phases of ρ_1 and ρ_k. We have to respect the fact that they are generally complex numbers. Similarly, even for the chosen real initial conditions, entries of the $\mathring{d}$-vectors x_k reflecting them may be complex. The individual ρ_k are numbered so that $|\rho_1| = |\rho_2| = \ldots = |\rho_l| > |\rho_{l+1}| \geq \ldots, |\rho_\partial|$ with $l \in \{1,\ldots,\partial\}$. Let us take $j_n \equiv \arg\max_j |\tilde{\phi}_{j(n-1)}|$. Let us consider the ratio

$$\zeta_{j_n n} \equiv \frac{\tilde{\phi}_{j_n n}}{\tilde{\phi}_{j_n(n-1)}} = \rho_1 \frac{\sum_{k=1}^{l} \exp(i\beta_k n)x_{j_n k} + \mathcal{O}(n)}{\sum_{k=1}^{l} \exp(i\beta_k (n-1))x_{j_n k} + \mathcal{O}(n-1)} \qquad (9.21)$$

with $\mathcal{O}(n) \to 0$, for $n \to \infty$.

Let us check that $\phi_{j_{n-1}(n-1)} \neq 0$. Then, $\zeta_{j_n n} \to_{n\to\infty} \rho_1$ if ρ_k are real for $k \leq l$.

Entries of $\tilde{\phi}$ are real. Thus, the coefficients at their powers have to coincide if ρ_1, ρ_2 is the conjugate pair. Then, $|\zeta_{j_n n}| = |\rho_1|(1 + \mathcal{O}(n))$. Thus, we get the needed information even in this case. More general cases are highly improbable in the addressed problem and we resign on them. The general case should be treated as *polynomial eigenvalue problem* and solved by the corresponding technique, using either software Laplace or its Matlab version.

Values of $\tilde{\phi}_n$ become unbounded in the unstable cases of interest and go to zero in other cases. Both makes their evaluation numerically sensitive. We need, however, the ratios $\zeta_{j_n n}$ only. Thus, we can normalize them by the same number, say η_n.

Let us assume that we know

$$\begin{aligned}
\bar{\phi}_{n-1,n-i} &\equiv \tilde{\phi}_{n-i}/\eta_{n-1},\ i = 1,\ldots,\partial. \quad \text{Then,} \\
\bar{\phi}_{n,n-i} &= \frac{\eta_{n-1}}{\eta_n}\bar{\phi}_{n-1,n-i} \equiv \gamma_n\bar{\phi}_{n-1,n-i},\ i = 1,\ldots,\partial, \quad \text{and}
\end{aligned}$$

$$
\begin{aligned}
U\bar{\phi}_{n,n} &= \gamma_n \sum_{i=1}^{\partial} UA_i\bar{\phi}_{n-1,n-i}. \quad \text{Then, the choice} \\
\eta_n &\equiv \max_{j\in\{1,\ldots,\mathring{d}\}} \left| \sum_{i=1}^{\partial} UA_i\bar{\phi}_{j,n-1,n-i} \right|
\end{aligned}
$$

concludes preparatory steps for describing the overall algorithm. It converges if the spectral radius coincides with module of a real, possibly multiple, eigenvalue or module of a single pair of complex eigenvalue.

Algorithm 9.1.3 (Spectral radius for factorized AR model)
Initial mode

- *Specify regression coefficient* θ_i, $i = 1, \ldots, \mathring{d}$, *corresponding to factors of the multi-variate AR model.*
- *Select an upper bound* $\mathring{n}$ *on the number of iterations* n *and a small* $\varepsilon > 0$ *used for stopping.*
- *Set the old estimate* $\tilde{\rho} = -1$ *of spectral radius* ρ.
- *Select the data vectors* $\Psi = \mathbf{1}_{\mathring{d}(\partial+1)}$.

Iterative mode

$$
\begin{array}{l}
\textit{For} \quad n = 1, \ldots, \mathring{n} \\
\qquad\qquad \textit{Set } j_n = 0,\ \eta_n = -1. \\
\quad \textit{For} \quad i = \mathring{d}, \ldots, 1 \\
\qquad\qquad \Psi_i = 0 \\
\qquad \textit{For} \quad k = i+1, \ldots, \partial\mathring{\Psi} \\
\qquad\qquad \Psi_i = \Psi_i + \theta_{ki}\Psi_k \\
\qquad \textit{end} \quad \textit{of the cycle over } k \\
\qquad\qquad \textit{Set } j_n = i \ \&\ \eta_n = |\Psi_i| \ \textit{if} \ |\Psi_i| > \eta_n \\
\quad \textit{end} \quad \textit{of the cycle over } i \\
\qquad\qquad \textit{Set } \rho = \left| \dfrac{\Psi_{j_n}}{\Psi_{j_n+\mathring{d}}} \right| \\
\quad \textit{For} \quad j = \partial\mathring{d}, \ldots, 1 \\
\qquad\qquad \Psi_j = \Psi_{j-\mathring{d}}/\eta_n \\
\quad \textit{end} \quad \textit{of the cycle over } j \\
\qquad\qquad \textit{Break } \text{Iterative mode } \textit{if } |\rho - \tilde{\rho}| < \varepsilon \\
\qquad\qquad \textit{Set } \tilde{\rho} = \rho \\
\textit{end} \quad \textit{of the cycle over } n
\end{array}
$$

3. Let us inspect individual components. Using Algorithm 9.1.3, the component is tested whether the expected value of its regression coefficients define a stable state matrix. It is taken as the point estimate searched for, if the answer is positive. Otherwise, we select $g < 1-\varepsilon$, see Algorithm 9.1.3, and transform parameters of i-th factor, $i = 1, \ldots, \mathring{d}$,

$$
\theta_i \to \tilde{\theta}_i = \begin{bmatrix} I_{i-1} & & & \\ & \frac{g}{\rho} I_{\mathring{d}} & & \\ & & \ddots & \\ & & & \left(\frac{g}{\rho}\right)^{\partial} I_{\mathring{d}} \end{bmatrix} \theta_i \equiv T_i \theta_i. \tag{9.22}
$$

It has distribution $GiW_{\tilde{\theta}_i,r_i}\left(T_i\hat{\theta}_i, T_iC_iT_i, \hat{r}_i, \nu_i\right)$. The expected value of this transformed component has spectral radius equal to $g < 1$.

Using a direct inspection or formalism of the Radon-Nikodým derivative [17], we find that the expected value $\hat{\theta}_i$ of the original θ_i can be expressed in terms of the expectation $\tilde{E}$ assigned to the pdf $GiW_{\tilde{\theta}_i,r_i}\left(T_i\hat{\theta}_i, T_iC_iT_i, \hat{r}_i, \nu_i\right)$. It is given by the formula

$$\hat{\theta}_i = \mathcal{E}[\chi_{\theta_s^*}(\theta_i)\theta_i] = \frac{\tilde{\mathcal{E}}[\chi_{\theta_s^*}(\theta_i)\theta_i q(\theta_i)]}{\tilde{\mathcal{E}}[\chi_{\theta_s^*}(\theta_i)q(\theta)]} \quad \text{with} \quad q(\theta_i) \equiv \tag{9.23}$$

$$\equiv \frac{GiW_{\theta_i,r_i}(\hat{\theta}_i, C, \hat{r}_i, \nu_i)}{GiW_{\theta_i,r_i}(T_i\hat{\theta}_i, T_iC_iT_i',\ \hat{r}_i, \nu_i)} = \exp\left[-\frac{\left(\theta_i'\left(C_i^{-1} - T_i^{-1}C_i^{-1}T_i^{-1}\right)\theta_i + 2\theta_i'C^{-1}\left(I - T_i^{-1}\right)\hat{\theta}_i\right)}{2r_i}\right] \equiv$$

$$\equiv \quad \exp\left[-\frac{1}{2r_i}\left(\theta_i'\tilde{C}_i^{-1}\theta_i + 2\theta_i'\tilde{l}_i\right)\right].$$

Thus, we can draw samples from the GiW pdf having a stable AR model as the expected value.

4. Let us apply Monte Carlo evaluation of both expectations discussed above. For each $i \in i^*$, we take samples $\Theta_{ik} \sim GiW_{\Theta_i}\left(T_i\hat{\theta}_i, T_iC_iT_i, \hat{r}_i, \nu_i\right)$ and compute

$$\hat{\theta}_{is} \quad \approx \quad \frac{\sum_k \chi_{\theta_s^*}(\theta_{ik})\theta_{ik}q(\Theta_{ik})}{\sum_k \chi_{\Theta_s^*}(\theta_{ik})q(\Theta_{ik})}. \tag{9.24}$$

This leads to the following modification of Algorithm 9.1.2.

Algorithm 9.1.4 (Modified point estimation of a stable AR system $\equiv$ a fixed component)
Initial mode

- *Perform the standard estimation, Chapter 8, giving the collection of statistics* $L_{i;\mathring{t}}, D_{i;\mathring{t}}, \nu_{i;\mathring{t}},\ i = 1, \ldots, \mathring{d}$.
- *Convert the collected statistics into their least-squares (LS) counterparts*

$$\begin{aligned}
\hat{\theta}_{i;\mathring{t}} &\equiv {}^{\lfloor\psi}L_{i;\mathring{t}}^{-1}\, {}^{\lfloor d\psi}L_{i;\mathring{t}} \equiv \textit{LS estimate of the regression coefficients}\\
\hat{r}_{i;\mathring{t}} &\equiv \frac{{}^{\lfloor d}D_{i;\mathring{t}}}{\nu_{i;\mathring{t}} - 2} \equiv \textit{LS estimate of the noise variance}\\
C_{i;\mathring{t}} &\equiv {}^{\lfloor\psi}L_{i;\mathring{t}}^{-1}\, {}^{\lfloor\psi}D_{i;\mathring{t}}^{-1}\left({}^{\lfloor\psi}L_{i;\mathring{t}}'\right)^{-1} \equiv \textit{LS covariance factor.}
\end{aligned}$$

- *Evaluate module ρ of the maximum singular value of the state matrix $\mathcal{A}$, using Algorithm 9.1.3.*
- *Stop and take the unrestricted LS point estimates as the approximation of (9.18) if $\rho < 1$ (if they define a stable AR model: if they belong to Θ_s^* (9.16)).*
- *Select a number $g_1 \in (0, 1)$, say $g_1 = 0.99$, used for stabilization and set $g = g_1/\rho$.*
- *Define, for $i = 1, \ldots, \mathring{d}$, matrices T_i (9.22) and transform moments $\bar{\theta}_i = T_i\hat{\theta}_{i;\mathring{t}}$, $\bar{C}_i = T_iC_{i;\mathring{t}}T_i$.*
- *Evaluate, for $i = 1, \ldots, \mathring{d}$, the kernels $\tilde{C}_i^{-1}$ and linear terms $\tilde{l}_i$ in (9.23).*
- *Select the upper bound $\mathring{n}$ (of the order 1e2 – 1e4) on the number n of iterations and set $n = 0$.*
- *Select the bound $\mathring{m}$ (of the order 1e1 - 1e3) on the necessary number m of stable samples and set $m = 0$.*

- *Initialize the generated stable point estimates* $\hat{\theta}_{is} = 0$, $\hat{r}_{is} = 0$, *and denominators* $m_i = 0$, *for* $i = 1, \ldots, \mathring{d}$.

Iterative mode

1. *Do while* $n < \mathring{n}$ & $m < \mathring{m}$.
2. *Set* $n = n + 1$.
3. *Generate, for* $i = 1, \ldots \mathring{d}$, *random samples* $\tilde{\theta}_i$, $\tilde{r}_i \sim GiW_{\theta_i, r_i}\left(\bar{\theta}_i, \hat{r}_{i;\mathring{t}}, \bar{C}_i, \nu_{i;\mathring{t}}\right)$

$$\begin{aligned}
\tilde{r}_i &= \hat{r}_{i;\mathring{t}}(1+e_r), \quad e_r \sim \mathcal{N}_{e_r}\left(0, \sqrt{\frac{2}{\nu_{i;\mathring{t}} - 4}}\right), \ cf.\ (8.23), \\
\tilde{r}_i &= \hat{r}_{i;\mathring{t}} \text{ if the sample } \tilde{r}_i \leq 0, \\
\tilde{\theta}_i &= \bar{\theta}_{i;\mathring{t}} + \sqrt{\tilde{r}_i}\bar{C}_i^{0.5} e, \quad e \sim \mathcal{N}_e(0, I_{\mathring{\theta}_i}).
\end{aligned}$$

   The normal, moment-preserving, approximation of the marginal pdf $f(r_i|d(\mathring{t}))$ *is adopted above.*

4. *Test the component stability using Algorithm 9.1.3 and go to the beginning of* Iterative mode *if* $\tilde{\Theta} \notin \Theta_s^*$, *if the generated multi-variate AR is unstable.*
5. *Set, for* $i = 1, \ldots, \mathring{d}$,

$$\tilde{m}_i = m_i + q\left(\tilde{\Theta}_i\right), \quad \tilde{\theta}_{is} = \hat{\theta}_{is} + \frac{q\left(\tilde{\Theta}_i\right)}{\tilde{m}_i}\left(\hat{\theta}_{is} - \tilde{\theta}_i\right), \quad \tilde{r}_{is} = \hat{r}_{is} + \frac{q\left(\tilde{\Theta}_i\right)}{\tilde{m}_i}\left(\hat{r}_{is} - \tilde{r}_i\right).$$

6. *Set* $m = m + 1$, $\hat{\Theta}_{is} = \tilde{\Theta}_{is}$ *and* $m_i = \tilde{m}_i$, *if* $\tilde{\theta}_s$ *– made of* $\tilde{\theta}_i$ *– is a stable AR model; check it by Algorithm 9.1.3.*
7. *Go to the beginning of* Iterative mode.

Accept $\hat{\Theta}_s$ *as the final estimate if* $m = \mathring{m}$. *Otherwise take the construction as unsuccessful.*

9.1.3 Advices and their influence

The specialization of the model presented in Section 7.1.3, is implied by Agreement 9.0.1. Among others, it implies that recognizable actions enter linearly regression vectors. Otherwise, the specialization to normal mixtures brings nothing new and it is done in connection with the respective designs.

9.1.4 Practical presentation aspects

Multi-step-ahead predictors

Ideally, the estimated mixture should describe well both deterministic and stochastic relationships between observed history and future behavior. It is especially important for the discussed design part. Quality of the estimated mixture can be well judged according to its performance as a *multi-step-ahead* predictor. Its exact practical construction is hard exactly for the same reasons that make hard the mixture estimation. Thus, an approximation is needed. The following one is based on our ability to evaluate the one-step-ahead predictor. Essentially, this predictor is used as random generator of a next data item, which is inserted into the condition of the next predictor etc.

This relatively short-horizon simulation is expected to reveal a significant information about the simulated model due to a large number of such simulation periods. The algorithm is rather simple.

Algorithm 9.1.5 (Simulation-based multi-step predictor)
Initial mode

- *Specify the analyzed normal mixture, typically, estimate it as described in Chapter 8.*

- *Select the prediction horizon $T \geq 1$.*

Sequential mode, *running for t=1,2,...,*

1. *Complement the state vector ϕ_t by d_t.*
2. *Specify recognizable actions $u_{0;t+1}$ and complement the regression vectors $\psi_{\mathring{\Delta}c;t+1}$, $c \in c^*$.*
3. *Generate $c_{t+1} \sim [\alpha_1, \ldots, \alpha_{\mathring{c}}]$.*

Make copies $\tilde{\phi}_{\mathring{\Delta}c;t} = \phi_{\mathring{\Delta}c;t}$, $c \in c^*$.

For $i = \mathring{d}, \ldots, \mathring{\Delta} + 1$

Simulate $d_{i;t+1} \sim \mathcal{N}_{d_{i;t+1}}\left(\theta'_{ic_{t+1}} \tilde{\psi}_{ic_{t+1};t+1}, r_{ic;t+1}\right)$.

Complement $\tilde{\psi}_{(i-1)c_{t+1};t+1}$ *by the simulated output of factors* $d_{i;t+1}$.

end of the cycle over i

Take d_{t+1} = *the initial* $\mathring{d}$*-entries of* $\tilde{\Psi}_{c_{t+1};t+1} \equiv \tilde{\psi}_{0c_{t+1};t+1}$.

For $c = 1, \ldots, \mathring{c}$

Create $\tilde{\phi}_{c;t+1}$.

end of the cycle over c

For $\tau = t+1, \ldots, t+T$

Generate $c_\tau \sim [\alpha_1, \ldots, \alpha_{\mathring{c}}]$.

For $i = \mathring{d}, \ldots, 1$

Simulate $d_{i;\tau} \sim \mathcal{N}_{d_{i;\tau}}\left(\theta'_{ic_\tau} \tilde{\psi}_{ic;\tau}, r_{ic_\tau;\tau}\right)$.

Complement $\tilde{\psi}_{(i-1)c_\tau;\tau}$ *by the simulated output of factors* $d_{i;\tau}$.

end of the cycle over i

Take the simulated d_τ *from initial* $\mathring{d}$ *entries of* $\tilde{\Psi}_{c_\tau;\tau} \equiv \tilde{\psi}_{0c_\tau;\tau}$.

For $c = 1, \ldots, \mathring{c}$

Create $\tilde{\psi}_{\mathring{d}c;\tau+1} = \tilde{\phi}_{c;\tau}$.

end of the cycle over c

end of the cycle over τ

Use one-step-ahead predictor, Proposition 8.1.14, given by the state vectors $\tilde{\phi}_{c;t+T-1}$, *for instance, evaluate moments of* d_{t+T}.

Remark(s) 9.1.4

1. *Except of the initial prediction time, when the recognizable actions are determined externally, the whole data record d_t should be simulated to get the real test of the overall model.*
2. *Component probabilities used for selecting the active mixture are either those estimated or their dynamic approximation, Section 7.1.2. The approximation can be restricted to the first step only or applied over whole prediction horizon. The second variant is conjectured as the better one.*
3. *A sort of ergodic hypothesis is behind the construction of the algorithm: relevance of the collection of simulations is supposed. This aspect should be inspected more closely.*

Support of visible part of normal mixture

A graphic representation of the mixture predicting several data entries, say y, is shown to the operator. It requires evaluation of a multi-variate interval $[\underline{y}, \overline{y}]$ on which the values of the pdf

$$f(y|\mathcal{P}) = \sum_{c \in c^*} \alpha_c \mathcal{N}_y(\mu_c, r_c)$$

are high enough. The component weights α_c, the vector-valued expected values μ_c and covariance matrices $r_c = L_c' D_c L_c$ are fixed in this task by the experience $\mathcal{P}$ in the condition.

We derive simple but useful algorithm for determining the interval $[\underline{y}, \overline{y}]$.

Normality implies that the highest value $(2\pi)^{-0.5\mathring{y}}\bar{\beta}$ of the weighted involved component is

$$(2\pi)^{-0.5\mathring{y}}\bar{\beta} \quad \equiv \quad \max_{c\in c^*, y\in(-\infty,\infty)^{\mathring{y}}} \alpha_c f(y|\mathcal{P},c) = (2\pi)^{-0.5\mathring{y}} \max_{c\in c^*} \beta_c, \quad \beta_c \equiv \frac{\alpha_c}{\sqrt{|r_c|}} = \frac{\alpha_c}{\sqrt{\prod_{i=1}^{\mathring{y}} D_{ic}}}.$$

Let us define the *visibility level* $K \in (0,1)$, typically, $K = 0.01$. Then, the weighted pdf describing the weighted c-th component is to be displayed on the appropriate interval $[\underline{y}, \overline{y}]$ on which $f(y|\mathcal{P},c) \geq K(2\pi)^{-0.5\mathring{y}}\bar{\beta}$. The extreme values $\underline{y}_{ic} \equiv \mu_{ic} - x$, $\overline{y}_{ic} = \mu_{ic} + x$ for the c-th component on the i-th axis, $i = 1,\ldots,\mathring{y}$, are determined by the positive solution of the equation

$$\max_{x_j\in(-\infty,\infty),\, j\neq i} \exp\left\{-0.5\,[x_1,\ldots,x_{i-1},x,x_{i+1},\ldots,x_{\mathring{y}}]\, r_c^{-1}\, [x_1,\ldots,x_{i-1},x,x_{i+1},\ldots,x_{\mathring{y}}]'\right\} =$$

$$= \quad \exp\left\{-0.5\frac{x^2}{D_{ic}}\right\} = K\frac{\bar{\beta}}{\beta_c} \Leftrightarrow x = \sqrt{D_{ic}\zeta} \equiv \sqrt{D_{ic} 2\ln\left(\frac{\beta_c}{K\bar{\beta}}\right)}.$$

The positive solution exists iff $\frac{\beta_c}{K\bar{\beta}} > 1$. The interval searched for has to cover union of intervals $[\underline{y}_{ic}, \overline{y}_{ic}]$ for all components $c \in c^*$ and all axes $i \in \{1,\ldots,\mathring{y}\}$. These considerations lead to the overall algorithm.

Algorithm 9.1.6 (Practical support of normal mixture)

Initial mode

- *Evaluate moments* $\mu_c, r_c \equiv L_c' D_c L_c$ *and weights* α_c *of the normal mixture of the interest.*
- *Specify the visibility level* $K \in (0,1)$*, typically,* $K = 0.01$.
- *Define the initial values of the constructed interval* $\underline{y}_i = +\infty$, $\overline{y}_i = -\infty$, $i = 1,\ldots,\mathring{y}$.
- *Set* $\bar{\beta} = -\infty$.

Construction mode

For $c = 1,\ldots,\mathring{c}$

$$\beta_c = \frac{\alpha_c}{\sqrt{\prod_{i=1}^{\mathring{y}} D_{ic}}}, \quad \bar{\beta} = \max(\bar{\beta}, \beta_c).$$

end of the cycle over c

For $c = 1,\ldots,\mathring{c}$

$$\beta_c \equiv \frac{\beta_c}{\bar{\beta}}, \quad \zeta_c = 2\ln\left(\frac{\beta_c}{K}\right).$$

Do if $\zeta_c > 0$, $\zeta_c = \sqrt{\zeta_c}$

For $i = 1,\ldots,\mathring{y}$

$$x_{ic} = \sqrt{D_{ic}}\zeta_c, \quad \underline{y}_i = \min(\underline{y}_i, \mu_{ic} - x_{ic}), \quad \overline{y}_i = \max(\overline{y}_i, \mu_{ic} + x_{ic})$$

end of the condition $\zeta_c > 0$

end of the cycle over i

end of the cycle over c

Re-scaling

Both learning and design work on scaled data. Scaling is necessary both for numeric reasons and unification of various default values. The final presentation has to be done in original user's units. It concerns predicted values only as we can always normalize data feeded into conditions, including those just contemplated ones. Thus, re-scaling reduces to a simple, often affine, transformation of pdfs, cf. Proposition 2.3.5.

Projections to be presented

Operator has to be presented low dimensional pdfs. Thus, a question arises, which of many possible projections is adequate to the addressed problem. The projected pdf is the ideal joint pdf resulting from the design. The optimized influence of the presentation action is based on the fact that the operator can willingly influence only the distribution of the shown variables having indices p_t. Thus, he can modify

$$f(d_t|d(t-1)) \rightarrow \frac{f(d_t|d(t-1))}{f(d_{p_t;t}|d(t-1))}\, {}^{\lfloor I}f(d_{p_t;t}|d(t-1)) \equiv {}^{\lfloor I}f(d_t|p_t, d(t-1)).$$

This relation models the influence of presentation actions p_t, Section 9.1.3. The strategy generating the actions p_t is optimized to get the modified pdf ${}^{\lfloor I}f(d_t|p_t, d(t-1))$ as close as possible to the user ideal pdf. It leads to tautology: the marginal ${}^{\lfloor I}f(d_{p_t;t}|d(t-1))$ has to be presented, otherwise the optimal nature of the constructed advising is lost.

Point advice

Generally, advices to the operator are presented as marginal pdfs of a mixture resulting from an optimization with the common instruction *seek for a high-probability area.* Sometimes, a point advice is required. If the mixture is uni-modal (if it contains single component), the maximizing argument is such an advice. It is harmonized with the above instruction. Question arises whether the same argument should be used in the multi-modal case. The answer is positive as such a choice can be interpreted as the maximum probable advice that approximately minimizes the distance of the closed loop behavior to the user target pdf also by selecting the user ideal pdf on advices as the optimized design knob, cf. Proposition 7.2.2.

On experimental evaluation of the KL distance

During experimental verifications of the designed advisory system, the natural question arisen how to evaluate experimentally the KL distance of the objective pdf ${}^{\lfloor o}f$ of data, see Chapter 2, to the user ideal pdf ${}^{\lfloor U}f$. The most straightforward approximation of the unknown ${}^{\lfloor o}f$ by a formal sample pdf fails. The sample pdf equals to the average of Dirac delta functions sitting on measured data and cannot be used in expressions of the type $\bullet \ln(\bullet)$. When approximating Dirac functions smoothly by Gaussian pdfs centered on measured data and having variance $R \rightarrow 0$, we also get infinite values. This made us to use the normal approximation but selecting R as the minimizer of the inspected KL distance instead of $R \rightarrow 0$. It seems to be a proper solution of the problem. Let us describe it in a detail.

Let ${}^{\lfloor o}f(d(\mathring{t})) = \prod_{t\in t^*} {}^{\lfloor o}f(d_t|\phi_{t-1})$, where the state vectors ϕ_t include both the state vector describing the objective dependence and the state vector in the user ideal pdf ${}^{\lfloor U}f(d(\mathring{t})) = \prod_{t\in t^*} {}^{\lfloor U}f(d_t|\phi_t)$. The normalized asymptotic KL distance can be written as follows

$$\overline{\lim}_{\mathring{t}\rightarrow\infty} \frac{1}{\mathring{t}} \mathcal{D}\left({}^{\lfloor o}f||\,{}^{\lfloor U}f\right) = \overline{\lim}_{\mathring{t}\rightarrow\infty} \mathcal{E}\underbrace{\left[\int {}^{\lfloor o}f(d_t|\phi_{t-1}) \ln\left(\frac{{}^{\lfloor o}f(d_t|\phi_{t-1})}{{}^{\lfloor U}f(d_t|\phi_{t-1})}\right) dd_t\right]}_{\omega(\phi_{t-1})}.$$

If we manage to evaluate at least approximately a finite sample version of $\omega(\phi_{t-1})$, we can use ordinary approximation

$$\mathcal{E}[\omega(\phi_{t-1})] \approx \frac{1}{\mathring{t}} \sum_{t=1}^{\mathring{t}} \omega(\tilde{\phi}_{t-1}),$$

hoping in ergodicity. The sign $\tilde{}$ denotes measured values in this paragraph.

As said above, we approximate the objective pdf ${}^{\lfloor o}f(d_t|\phi_{t-1})$ at the measured point $\tilde{d}_t, \tilde{\phi}_{t-1}$ by the normal pdf

$${}^{\lfloor o}f(d_t|\tilde{\phi}_{t-1}) \approx \mathcal{N}_{d_t}(\tilde{d}_t, R)$$

with the optional covariance matrix R. For the following normal user ideal pdf

$${}^{\lfloor U}f(d_t|\tilde{\phi}_{t-1}) = \mathcal{N}_{d_t}\left({}^{\lfloor U}M(\tilde{\phi}_{t-1}),\, {}^{\lfloor U}R\right), \text{ Proposition 8.1.10 implies}$$

$$\omega(\tilde{\phi}_{t-1}) = 0.5\left[\ln\left|{}^{\lfloor U}RR^{-1}\right| - \mathring{d} + \mathrm{tr}\left[R\,{}^{\lfloor U}R^{-1}\right] + \left(\tilde{d}_t - {}^{\lfloor U}M(\tilde{\phi}_{t-1})\right)'\,{}^{\lfloor U}R^{-1}\left(\tilde{d}_t - {}^{\lfloor U}M(\tilde{\phi}_{t-1})\right)\right].$$

It is minimized by $R = {}^{\lfloor U}R$. This result and the sample-mean replacement of the expectation give the practical plausible indicator of the design quality

$$\frac{1}{2\mathring{t}} \sum_{t=1}^{\mathring{t}} \left(\tilde{d}_t - {}^{\lfloor U}M(\tilde{\phi}_{t-1})\right)' {}^{\lfloor U}R^{-1} \left(\tilde{d}_t - {}^{\lfloor U}M(\tilde{\phi}_{t-1})\right). \tag{9.25}$$

9.1.5 Approximate fully probabilistic design and quadratic forms

Section 7.1.4 expresses the KL distance of the constructed ideal pdf on the p-data $d(\mathring{t})$ to the user ideal pdf ${}^{\lfloor U}f(d(\mathring{t}))$. It consists of the true user ideal pdf $\prod_{t\in t^*} {}^{\lfloor U}f(d_{o;t}|d_o(t-1))$ and of the factor on the surplus data $d_{p+}(\mathring{t})$ of the p-system. According to the assumption (5.6), it coincides with the pdf $\prod_{t\in t^*} {}^{\lfloor I}f(d_{p+;t}|d(t-1))$ derived from the constructed ideal pdf ${}^{\lfloor I}f(d(\mathring{t}))$. The fully probabilistic design, Proposition 2.4.4, with this special target is described by Proposition 7.1.4. Its normal counterpart brings nothing new. The specific nature of the normal case can be exploited after expressing upper bounds on the KL distance. Before progressing in this respect, the necessary manipulations with expected quadratic forms and the conditional KL distance of normal pdfs are prepared. The formulas are expressed in the way suitable for an efficient evaluation that relies on the factorized form of normal components.

It is worthwhile to stress, that LDL' decomposition is used further on as it fits to the operations needed in the discussed design part of construction of the p-system.

Proposition 9.1.4 (Expected quadratic form for a factorized normal component) *Under Agreement 9.0.1, let us consider a normal parameterized component described by*

$$\begin{aligned}
f(\Delta_t|u_{o;t},\phi_{t-1},\Theta) &\equiv \prod_{i\in i^*} \mathcal{N}_{\Delta_{i;t}}(\theta_i'\psi_{i;t}, r_i) \quad \textit{with} \qquad (9.26)\\
\Theta &\equiv (\Theta_1,\ldots,\Theta_{\mathring{\Delta}}), \quad \Theta_i \equiv [\theta_i, r_i]\\
\psi_{i;t}' &\equiv [\Delta'_{(i+1)\ldots\mathring{\Delta};t}, u_{o;t}', \phi_{t-1}'] \equiv [\Delta_{i+1;t}, \psi_{i+1;t}'], \ i = 0,1,\ldots,\mathring{\Delta}-1, \ \textit{recall}\ \psi_{0;t} \equiv \Psi_t,\\
\psi_{\mathring{\Delta};t}' &\equiv [u_{o;t}', \phi_{t-1}'].
\end{aligned}$$

Let us consider a quadratic form in $\psi_{0;t} \equiv \Psi_t$ with the decomposed kernel $L_0D_0L_0'$.

Then, the expected quadratic form, lifted by a constant k_0, reads

$$\begin{aligned}
&\mathcal{E}[k_0 + \Psi_t' L_0 D_0 L_0' \Psi_t | u_{o;t}, \phi_{t-1}] \equiv \mathcal{E}[k_0 + \psi_{0;t}' L_0 D_0 L_0' \psi_{0;t} | \psi_{\mathring{\Delta};t}] = k_{\mathring{\Delta}} + \psi_{\mathring{\Delta};t}' L_{\mathring{\Delta}} D_{\mathring{\Delta}} L_{\mathring{\Delta}}' \psi_{\mathring{\Delta};t}, \quad \textit{where}\\
&L_{i+1}D_{i+1}L_{i+1}' = {}^{\lfloor \psi}L_i\, {}^{\lfloor \psi}D_i\, {}^{\lfloor \psi}L_i' + \left(\theta_{i+1} + {}^{\lfloor \Delta\psi}L_i\right) {}^{\lfloor \Delta}D_i \left(\theta_{i+1} + {}^{\lfloor \Delta\psi}L_i\right)' \qquad (9.27)\\
&k_{i+1} = k_i + {}^{\lfloor \Delta}D_i r_{i+1}, \quad \textit{for}\ i = 0,\ldots,\mathring{\Delta}-1, \quad \textit{with}\\
&L_i \equiv \begin{bmatrix} 1 & 0 \\ {}^{\lfloor \Delta\psi}L_i & {}^{\lfloor \psi}L_i \end{bmatrix}, \ D_i \equiv \operatorname{diag}\left[{}^{\lfloor \Delta}D_i,\ {}^{\lfloor \psi}D_i\right], \quad \textit{where}\ {}^{\lfloor \Delta}D_i\ \textit{is scalar and}\ \mathring{D}_{i+1} = \mathring{D}_i - 1.
\end{aligned}$$

The updating the LDL' decomposition can be made using directly Algorithm 8.1.2.

Proof: The expectation is taken over entries of Δ_t as the remaining part of the data vector Ψ_t is fixed by the condition $\psi_{\mathring{\Delta};t} \equiv [u_{o;t}', \phi_{t-1}']'$. The chain rule for expectations, Proposition 2.3.6, implies that we can evaluate conditional expectations of individual entries in the vector Δ_t one-by-one, starting from the first one. Taking a generic step and using the identity

$$\mathcal{E}\left[\Delta_{i+1}^2|\psi_{i+1}\right] = r_{i+1} + \{\mathcal{E}[\Delta_{i+1}|\psi_{i+1}]\}^2, \quad \text{we have}$$

$$\begin{aligned}
&\mathcal{E}\left[k_i + \psi_{i;t}' L_i D_i L_i' \psi_{i;t} | \psi_{i+1;t}\right] = \underbrace{k_i + {}^{\lfloor \Delta}D_i r_{i+1}}_{k_{i+1}} + \psi_{i+1;t}' \begin{bmatrix} \theta_{i+1}' \\ I_{\mathring{\psi}_{i+1}} \end{bmatrix}' L_i D_i L_i' \begin{bmatrix} \theta_{i+1}' \\ I_{\mathring{\psi}_{i+1}} \end{bmatrix} \psi_{i+1;t} =\\
&= k_{i+1} + \psi_{i+1;t}' \underbrace{\left[{}^{\lfloor \psi}L_i\, {}^{\lfloor \psi}D_i\, {}^{\lfloor \psi}L_i' + \left(\theta_{i+1} + {}^{\lfloor \Delta\psi}L_i\right) {}^{\lfloor \Delta}D_i \left(\theta_{i+1} + {}^{\lfloor \Delta\psi}L_i\right)'\right]}_{L_{i+1}D_{i+1}L_{i+1}'} \psi_{i+1;t}.
\end{aligned}$$

□

The inclusion of the unit into the regression vector allows us to compute formally without the linear term that is present in a general quadratic form. Numerically, it is dangerous: unit in the state is the entry evolving on stability boundary. Consequently, the evaluation noise causes numerical divergence in dynamic programming. Moreover, the constant lift of the quadratic form is split into two parts and design algorithms need their sum. In software implementation, it is simply avoided by treating the *linear term of quadratic form* independently. For it, the following proposition is useful.

Proposition 9.1.5 (Expected linear form for a factorized normal component) *Under Agreement 9.0.1, let us consider a normal parameterized component described by (9.26).*

Let us consider a linear form $l_0'\psi_{0;t}$ in $\psi_{0;t} \equiv \Psi_t$ with the vector weight l_0.

Then, the expected linear form is

$$\begin{aligned} \mathcal{E}[l_0'\Psi_t|u_{o;t},\phi_{t-1}] &\equiv \mathcal{E}[l_0'\psi_{0;t}|\psi_{\mathring{\Delta};t}] = l'_{\mathring{\Delta}}\psi_{\mathring{\Delta};t}, \quad \text{where for } i = 0,\ldots,\mathring{\Delta}-1 \qquad (9.28)\\ l'_{i+1} &= {}^{\lfloor\Delta}l_i\theta'_{i+1} + {}^{\lfloor\psi}l'_i \quad \text{with, } \; l_i \equiv \left[\; {}^{\lfloor\Delta}l_i \quad {}^{\lfloor\psi}l'_i \;\right]', \; {}^{\lfloor\Delta}l_i \text{ is scalar and } \mathring{l}_{i+1} = \mathring{l}_i - 1. \end{aligned}$$

Proof: The expectation is taken over entries of Δ_t as the remaining part of the data vector Ψ_t is fixed by the condition $\psi_{\mathring{\Delta};t} \equiv [u'_{o;t},\phi'_{t-1}]'$. The chain rule for expectations, Proposition 2.3.6, implies that we can evaluate conditional expectations of individual entries in Δ_t one-by-one, starting from the first one. Taking a generic step, we have

$$\mathcal{E}\left[l_i'\psi_{i;t}|\psi_{i+1;t}\right] = l_i'\begin{bmatrix}\theta'_{i+1}\\ I_{\mathring{\psi}_{i+1}}\end{bmatrix}\psi_{i+1;t} = \underbrace{\left[{}^{\lfloor\Delta}l_i\theta'_{i+1} + {}^{\lfloor\psi}l'_i\right]}_{l'_{i+1}}\psi_{i+1;t}.$$

□

A combination of Propositions 9.1.4, 9.1.5 gives the algorithm for a numerically safe evaluation of the expected value of the general lifted quadratic form.

Algorithm 9.1.7 (Safe evaluation of the expected value of the lifted quadratic form)
Initial mode

- *Specify parameters $\left(\bar\theta'_i, r_i\right) \equiv \left([\theta'_i,\mu_i], r_i\right)$ of a component in the factorized form with the scalar* offset *μ_i at the last position.*
- *Specify the lift $\bar k_0 \equiv k_0$ and the kernel $\bar L_0\bar D\bar L_0'$ of the lifted quadratic form in $k_0 + \psi'_{0;t}\bar L_0\bar D\bar L_0'\psi_{0;t}$ whose expectation conditioned by $\psi_{\mathring{\Delta};t}$ is evaluated. The matrices $\bar L_i, \bar D_i,\; i = 0,\ldots,\mathring{\Delta}-1$, are split*

$$\bar L_i = \begin{bmatrix} L_i & 0\\ l'_i & 1\end{bmatrix},\; \bar D_i = \begin{bmatrix} D_i & \\ & 0\end{bmatrix}.$$

Recursive mode

For $\quad i = 1,\ldots,\mathring{\Delta}$

$L_{i+1}D_{i+1}L'_{i+1} = {}^{\lfloor\psi}L_i\,{}^{\lfloor\psi}D_i\,{}^{\lfloor\psi}L'_i + \left(\theta_{i+1} + {}^{\lfloor\Delta\psi}L_i\right){}^{\lfloor\Delta}D_i\left(\theta_{i+1} + {}^{\lfloor\Delta\psi}L_i\right)'$, *where updating of the LDL' decomposition is performed using Algorithm 8.1.2*

$k_{i+1} = k_i + {}^{\lfloor\Delta}D_i r_{i+1}$, *with*

$L_i \equiv \begin{bmatrix} 1 & 0\\ {}^{\lfloor\Delta\psi}L_i & {}^{\lfloor\psi}L_i\end{bmatrix}$, $D_i \equiv \operatorname{diag}\left[{}^{\lfloor\Delta}D_i,\, {}^{\lfloor\psi}D_i\right]$, *where* ${}^{\lfloor\Delta}D_i$ *is scalar,* $\mathring{D}_{i+1} = \mathring{D}_i - 1$

$l'_{i+1} = {}^{\lfloor\Delta}l_i\bar\theta'_{i+1} + {}^{\lfloor\psi}l'_i$ *with,* $l_i \equiv \left[\;{}^{\lfloor\Delta}l_i \quad {}^{\lfloor\psi}l'_i\;\right]'$, *where* ${}^{\lfloor\Delta}l_i$ *is scalar,* $\mathring{l}_{i+1} = \mathring{l}_i - 1$

$k_{i+1} \equiv k_i + 2l_{\mathring{\psi}_{i+1}(i+1)}$, $\; l_{\mathring{\psi}_{i+1}(i+1)} = 0$ *shifts constant from the linear term to the lift.*

end of the cycle over i

The following auxiliary proposition ie repeatedly used in approximate, fully probabilistic designs.

Proposition 9.1.6 (The conditional KL distance) *Under Agreement 9.0.1, let* $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t})$

$$\text{and } f(\Delta_t|u_{o;t}, d(t-1)) = \prod_{i=1}^{\mathring{\Delta}} \mathcal{N}_{\Delta_{i;t}}(\theta_i'\psi_{i;t}, r_i), \quad {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d(t-1)) = \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\ {}^{\lfloor U}r_i\right).$$

The regression coefficients ${}^{\lfloor U}\theta_i$ *of the user ideal pdf* ${}^{\lfloor U}f(\cdot)$ *are complemented by zeros so that the regression vectors* $\psi_{i;t}$ *for the* i*-th factor and the corresponding factor of the user ideal pdf are common. Recall that*

$$\begin{aligned}
\psi_{i;t} &= [\Delta'_{(i+1)\dots\mathring{\Delta};t}, u'_{o;t}, \phi'_{t-1}]' \equiv [\Delta'_{(i+1)\dots\mathring{\Delta};t}, \psi'_{\mathring{\Delta};t}]' \equiv [\Delta_{(i+1);t}, \psi'_{i+1;t}]',\ i \in \{0,\dots,\mathring{\Delta}-1\}\\
\psi_{\mathring{\Delta};t} &= [u'_{o;t}, \phi'_{t-1}]'.
\end{aligned}$$

Then,

$$\begin{aligned}
\omega(u_{o;t}, \phi_{t-1}) &\equiv \omega(\psi_{\mathring{\Delta};t}) \equiv 2\int f(\Delta_t|u_{o;t}, \phi_{t-1}) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, \phi_{t-1})}{{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, \phi_{t-1})}\right) d\Delta_t = \qquad (9.29)\\
&= k_{\mathring{\Delta}} + \psi'_{\mathring{\Delta};t} L_{\mathring{\Delta}} D_{\mathring{\Delta}} L'_{\mathring{\Delta}} \psi_{\mathring{\Delta};t},\ \text{where the lift } k_{\mathring{\Delta}} \text{ and kernel } L_{\mathring{\Delta}} D_{\mathring{\Delta}} L'_{\mathring{\Delta}} \text{ of}
\end{aligned}$$

the conditional KL distance are found recursively

For $\quad i = 1,\dots,\mathring{\Delta}$

$$\begin{aligned}
&L_i D_i L_i' = {}^{\lfloor\psi}L_{i-1}\,{}^{\lfloor\psi}D_{i-1}\,{}^{\lfloor\psi}L'_{i-1} + \left(\theta_i + {}^{\lfloor\Delta\psi}L_{i-1)}\right) {}^{\lfloor\Delta}D_{i-1} \left(\theta_i + {}^{\lfloor\Delta\psi}L_{i-1}\right)' +\\
&+\chi\left(i \le \mathring{\Delta}_o\right)\left(\theta_i - {}^{\lfloor U}\theta_i\right) {}^{\lfloor U}r_i^{-1} \left(\theta_i - {}^{\lfloor U}\theta_i\right)'\\
&k_i = k_{i-1} + {}^{\lfloor\Delta}D_{i-1} r_i + \chi\left(i \le \mathring{\Delta}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right) + \frac{r_i}{{}^{\lfloor U}r_i}\right],\\
&k_0 = -\mathring{\Delta}_o,\ D_0 = 0_{\mathring{\psi},\mathring{\psi}},\ L_0 = I_{\mathring{\psi}} \qquad (9.30)\\
&L_i \equiv \begin{bmatrix} 1 & 0\\ {}^{\lfloor\Delta\psi}L_i & {}^{\lfloor\psi}L_i \end{bmatrix},\ D_i \equiv \operatorname{diag}\left[{}^{\lfloor\Delta}D_i,\ {}^{\lfloor\psi}D_i\right], \quad \text{where}\\
&{}^{\lfloor\Delta}D_i \text{ is scalar and } \mathring{D}_{i+1} = \mathring{D}_i - 1.
\end{aligned}$$

end $\quad$ *of the cycle over* i

The updating of the LDL' (!) decomposition can be done by a double, if the set indicator $\chi(\cdot) = 1$*, or single, if* $\chi(\cdot) = 0$*, use of Algorithm 8.1.2.*

Proof: Let the innovations be split $\Delta = (\Delta_o, \Delta_{p+})$. Then, $\omega(u_{o;t}, \phi_{t-1}) \equiv$

$$\begin{aligned}
&\equiv 2\int f(\Delta_t|u_{o;t}, \phi_{t-1}) \ln\left(\frac{f(\Delta_t|u_{o;t}, d(t-1))}{f(\Delta_{p+;t}|u_{o;t}, d(t-1))\ {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d(t-1))}\right) d\Delta_t =\\
&= \sum_{i=1}^{\mathring{\Delta}_o} \ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right) + \sum_{i=1}^{\mathring{\Delta}_o} \int f(\Delta_t|u_{o;t}, \phi_{t-1}) \left[-\frac{\left(\Delta_{i;t} - \theta_i'\psi_{i;t}\right)^2}{r_i} + \frac{\left(\Delta_{i;t} - {}^{\lfloor U}\theta_i'\psi_{i;t}\right)^2}{{}^{\lfloor U}r_i}\right] d\Delta_t =\\
&\underbrace{=}_{\substack{\text{Proposition 2.3.6}\\ \mathcal{E}[\bullet^2] = \mathcal{E}^2[\bullet] + \operatorname{cov}[\bullet]}} \sum_{i=1}^{\mathring{\Delta}_o} \ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right) - \mathring{\Delta}_o + \sum_{i=1}^{\mathring{\Delta}_o} \int f(\Delta_t|u_{o;t}, \phi_{t-1}) \frac{(\Delta_{i;t} - {}^{\lfloor U}\theta_i'\psi_{i;t})^2}{{}^{\lfloor U}r_i} d\Delta_t =\\
&\underbrace{=}_{\text{Proposition 2.3.6}} -\mathring{\Delta}_o + \sum_{i=1}^{\mathring{\Delta}_o}\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right) + \frac{r_i}{{}^{\lfloor U}r_i}\right] + \sum_{i=1}^{\mathring{\Delta}_o} \mathcal{E}\left[\psi'_{i;t}\left(\theta_i - {}^{\lfloor U}\theta_i\right) {}^{\lfloor U}r_i^{-1} \left(\theta_i - {}^{\lfloor U}\theta_i\right)' \psi_{i;t} | \psi_{\mathring{\Delta};t}\right].
\end{aligned}$$

Let us define kernel $L_{i-1}D_{i-1}L'_{i-1}$ of the lifted quadratic form $k_{i-1}+\psi'_{i-1;t}L_{i-1}D_{i-1}L'_{i-1}\psi_{i-1;t}$ for which we evaluate $\mathcal{E}[k_{i-1}+\psi'_{i-1;t}L_{i-1}D_{i-1}L'_{i-1}\psi_{i-1;t}|\psi_{\mathring{\Delta};t}]$. Then, according to the equation (9.27) in the proof of Proposition 9.1.4, an intermediate lifted quadratic form arises $\tilde{k}_i+\psi'_{i;t}\tilde{L}_i\tilde{D}_i\tilde{L}'_i\psi_{i;t}$ with

$$\tilde{L}_i\tilde{D}_i\tilde{L}'_i \;=\; {}^{\lfloor\psi}L_{i-1}\,{}^{\lfloor\psi}D_{i-1}\,{}^{\lfloor\psi}L'_{i-1}+\left(\theta_i+{}^{\lfloor\Delta\psi}L_{i-1}\right){}^{\lfloor\Delta}D_{i-1}\left(\theta_i+{}^{\lfloor\Delta\psi}L_{i-1}\right)',\quad \tilde{k}_i=k_{i-1}+{}^{\lfloor\Delta}D_{i-1}r_i.$$

While $i\leq\mathring{\Delta}_o$, the expression $\ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right)+\frac{r_i}{{}^{\lfloor U}r_i}+\psi'_{i;t}\left(\theta_i-{}^{\lfloor U}\theta_i\right){}^{\lfloor U}r_i^{-1}\left(\theta_i-{}^{\lfloor U}\theta_i\right)'\psi_{i;t}$ has to be added to it

$$\begin{aligned}
L_iD_iL'_i &= {}^{\lfloor\psi}L_{i-1}\,{}^{\lfloor\psi}D_{i-1}\,{}^{\lfloor\psi}L'_{i-1}+\left(\theta_i+{}^{\lfloor\Delta\psi}L_{i-1}\right){}^{\lfloor\Delta}D_{i-1}\left(\theta_i+{}^{\lfloor\Delta\psi}L_{i-1}\right)'+\\
&+\ \chi\left(i\leq\mathring{\Delta}_o\right)\frac{\left(\theta_i-{}^{\lfloor U}\theta_i\right)\left(\theta_i-{}^{\lfloor U}\theta_i\right)'}{{}^{\lfloor U}r_i}\\
k_i &= k_{i-1}+{}^{\lfloor\Delta}D_{i-1}r_i+\\
&+\ \chi\left(i\leq\mathring{\Delta}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right)+\frac{r_i}{{}^{\lfloor U}r_i}\right].
\end{aligned}$$

The updating can be made directly by a double or single use of Algorithm 8.1.2. The proper initial values are $k_0=-\mathring{\Delta}_o$ and $D_0=0$; $L_0=I=$ unit matrix. □

While performing the approximate fully probabilistic design, we work with a weighted version of the conditional KL distance. The following slight extension of Proposition 9.1.6 supports the cases for which the function $-\ln(\gamma)$ determining the Bellman function is approximated by a lifted quadratic form.

Proposition 9.1.7 (The weighted conditional KL distance) *Under Agreement 9.0.1, let $\Delta_t=(\Delta_{o;t},\Delta_{p+;t})=$ (o-innovations, innovations in the surplus p-space) and*

$$f(\Delta_t|u_{o;t},d(t-1)) \;=\; \prod_{i=1}^{\mathring{\Delta}}\mathcal{N}_{\Delta_{i;t}}(\theta'_i\psi_{i;t},r_i),\quad {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},d(t-1))=\prod_{i=1}^{\mathring{\Delta}_o}\mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta'_i\psi_{i;t},\,{}^{\lfloor U}r_i\right).$$

The regression coefficients ${}^{\lfloor U}\theta_i$ of the user ideal pdf ${}^{\lfloor U}f(\cdot)$ are complemented by zeros so that regression vectors $\psi_{i;t}$ of the corresponding factors coincide. Recall,

$$\psi_{i;t}=[\Delta'_{(i+1)\ldots\mathring{\Delta};t},u'_{o;t},\phi'_{t-1}]'\equiv[\Delta'_{(i+1)\ldots\mathring{\Delta};t},\psi'_{\mathring{\Delta};t}]'\equiv[\Delta_{(i+1);t},\psi'_{i+1;t}]',i\in\{0,\ldots,\mathring{\Delta}-1\},\ \psi_{\mathring{\Delta};t}=[u'_{o;t},\phi'_{t-1}]'.$$

Let, moreover,

$$\begin{aligned}
\gamma(\phi_t) &\equiv \exp\left[-0.5(k_\gamma+\phi'_tL_\gamma D_\gamma L'_\gamma\phi_t)\right],\quad \textit{where} &(9.31)\\
\phi_t &\equiv [d'_{t\ldots(t-\partial+1)},1]'\Rightarrow\psi_{0;t}\equiv\Psi_t\equiv[\Delta'_t,u'_{o;t},\phi'_{t-1}]',\\
L_\gamma &\equiv \textit{a lower triangular matrix with unit diagonal}\\
D_\gamma &\equiv \textit{a diagonal matrix with non-negative diagonal entries.}
\end{aligned}$$

$$\textit{Then,}\quad \omega_\gamma(u_{o;t},\phi_{t-1})\equiv 2\int f(\Delta_t|u_{o;t},\phi_{t-1})\ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t},u_{o;t},\phi_{t-1})}{\gamma(\phi_t)\,{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},\phi_{t-1})}\right)d\Delta_t= \qquad (9.32)$$

$$= k_{\mathring{\Delta}}+\psi'_{\mathring{\Delta};t}L_{\mathring{\Delta}}D_{\mathring{\Delta}}L'_{\mathring{\Delta}}\psi_{\mathring{\Delta};t},\quad \textit{the lift and kernel of the KL distance are found recursively}$$

For $i=1,\ldots,\mathring{\Delta}$

$$\begin{aligned}
L_iD_iL'_i &= {}^{\lfloor\psi}L_{i-1}\,{}^{\lfloor\psi}D_{i-1}\,{}^{\lfloor\psi}L'_{i-1}+\left(\theta_i+{}^{\lfloor\Delta\psi}L_{i-1}\right){}^{\lfloor\Delta}D_{i-1}\left(\theta_i+{}^{\lfloor\Delta\psi}L_{i-1}\right)'+\\
&+\chi\left(i\leq\mathring{\Delta}_o\right)\frac{\left(\theta_i-{}^{\lfloor U}\theta_i\right)\left(\theta_i-{}^{\lfloor U}\theta_i\right)'}{{}^{\lfloor U}r_i}\\
k_i &= k_{i-1}+{}^{\lfloor\Delta}D_{i-1}r_i+\chi\left(i\leq\mathring{\Delta}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_i}\right)+\frac{r_i}{{}^{\lfloor U}r_i}\right]
\end{aligned}$$

$$
\begin{aligned}
&\textit{end} \quad \textit{of the cycle over } i\\
k_0 &= -\mathring{\Delta}_o + k_\gamma,\ L_0 D_0 L_0' = \mathcal{K} L_\gamma D_\gamma L_\gamma' \mathcal{K}'\\
\mathcal{K}' &\equiv \begin{bmatrix} I_{\mathring{d}(\partial-1)} & 0 & 0\\ 0 & 0_{1,\mathring{d}} & 1 \end{bmatrix} \qquad (9.33)\\
L_i &\equiv \begin{bmatrix} 1 & 0\\ {}^{\lfloor \Delta\psi}L_i & {}^{\lfloor \psi}L_i \end{bmatrix},\ D_i \equiv \mathrm{diag}\left[{}^{\lfloor \Delta}D_i,\ {}^{\lfloor \psi}D_i\right],\quad \textit{where } {}^{\lfloor \Delta}D_i \textit{ is scalar, } \mathring{D}_{i+1} = \mathring{D}_i - 1.
\end{aligned}
$$

The updating of the LDL' (!) decomposition can be made by a double, if the set indicator $\chi(\cdot) = 1$, or single, if $\chi(\cdot) = 0$, use of Algorithm 8.1.2.

Proof: The function $\gamma(\phi_t)$ just adds non-trivial initial conditions. The matrix $\mathcal{K}$ extends the quadratic form in ϕ_t to the quadratic form in $\Psi_t \equiv \psi_{0;t}$. Otherwise, the proof of Proposition 9.1.6 can be copied. $\square$

The role of the matrix $\mathcal{K}$ (9.33) that expresses the vector ϕ_t as a function of $\Psi_t \equiv \psi_{0;t}$ is more visible if we distinguish the entries corresponding to the position of 1 in the phase form of the state vector, Agreement 9.0.1.

$$
\text{Let} \qquad L_\gamma \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_\gamma & 0\\ {}^{\lfloor \phi 1}L_\gamma' & 1 \end{bmatrix},\ D_\gamma = \mathrm{diag}\left[{}^{\lfloor \phi 0}D_\gamma,\ {}^{\lfloor 1}D_\gamma\right],\ {}^{\lfloor 1}D_\gamma \text{ is scalar.} \qquad (9.34)
$$

$$
\text{Then,} \qquad L_\gamma'\mathcal{K}' = \begin{bmatrix} {}^{\lfloor \phi 0}L_\gamma' & 0 & {}^{\lfloor \phi 1}L_\gamma\\ 0 & 0_{1,\mathring{d}} & 1 \end{bmatrix} \text{ and}
$$

$$
\mathcal{K} L_\gamma D_\gamma L_\gamma' \mathcal{K}' = \begin{bmatrix} {}^{\lfloor \phi 0}L_\gamma\, {}^{\lfloor \phi 0}D_\gamma\, {}^{\lfloor \phi 0'}L_\gamma & 0 & {}^{\lfloor \phi 0}D_\gamma\, {}^{\lfloor \phi 1}L_\gamma\\ 0 & 0_{\mathring{d},\mathring{d}} & 0_{\mathring{d},1}\\ {}^{\lfloor \phi 1}L_\gamma'\, {}^{\lfloor \phi 0}L_\gamma & 0_{1,\mathring{d}} & {}^{\lfloor \phi 1}L_\gamma'\, {}^{\lfloor \phi 0}D_\gamma\, {}^{\lfloor \phi 1}L_\gamma + {}^{\lfloor 1}D_\gamma \end{bmatrix}.
$$

For reference purposes, let us describe it algorithmically.

Algorithm 9.1.8 (Extension of kernel $\phi_t' L_\gamma D_\gamma L_\gamma' \phi_t \to \psi_{0;t}' L_0 D_0 L_0' \psi_{0;t}$)

$$
D_0 = \mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor \phi 0}D_\gamma\right], 0_{1,\mathring{d}}, {}^{\lfloor 1}D_\gamma\right],\ L_0 = \begin{bmatrix} {}^{\lfloor \phi 0}L_\gamma & 0 & 0\\ 0 & I_{\mathring{d}} & 0\\ {}^{\lfloor \phi 1}L_\gamma' & 0 & 1 \end{bmatrix},\ {}^{\lfloor 1}D_\gamma \textit{ is scalar.}
$$

The following auxiliary proposition is useful in the industrial and simultaneous designs operating at the factor level.

Proposition 9.1.8 (Normal expectation of quadratic form in exponent) *Let $\psi_{\mathring{\Delta}}' = [u', \phi']$ and $f(u|\phi) = \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{i-\mathring{\Delta}}}(\theta_i'\psi_i, r_i) \equiv \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{\Delta_i}(\theta_i'\psi_i, r_i)$ with known parameters and $\psi_i' = [u_{i+1-\mathring{\Delta}}', \psi_{i+1}']$, $\mathring{\Delta} \le i < \mathring{d}$, $\psi_{\mathring{d}} = \phi$. Let us consider the lifted quadratic positive semi-definite form $k_{\mathring{\Delta}} + \psi_{\mathring{\Delta}}' L_{\mathring{\Delta}} D_{\mathring{\Delta}} L_{\mathring{\Delta}}' \psi_{\mathring{\Delta}}$. The lift $k_{\mathring{\Delta}}$ is non-negative, $L_{\mathring{\Delta}}$ is a lower triangular matrix with unit diagonal and $D_{\mathring{\Delta}}$ is a diagonal matrix with non-negative diagonal entries. Then,*

$$
\begin{aligned}
&\mathcal{E}\left[\exp\left\{-0.5\left(k_{\mathring{\Delta}} + \psi_{\mathring{\Delta}}' L_{\mathring{\Delta}} D_{\mathring{\Delta}} L_{\mathring{\Delta}}' \psi_{\mathring{\Delta}}\right)\right\} | \phi\right] \equiv \int f(u|\phi) \exp\left\{-0.5(k_{\mathring{\Delta}} + \psi_{\mathring{\Delta}}' L_{\mathring{\Delta}} D_{\mathring{\Delta}} L_{\mathring{\Delta}}' \psi_{\mathring{\Delta}})\right\} du =\\
&= \exp\left[-0.5(k_{\mathring{d}} + \phi' L_{\mathring{d}} D_{\mathring{d}} L_{\mathring{d}}' \phi)\right],\ \textit{where the lift and kernel are found recursively}\\
&\textit{For} \quad i = \mathring{\Delta}+1, \ldots, \mathring{d}\\
&\qquad \tilde{L}_i \tilde{D}_i \tilde{L}_i' = L_{i-1} D_{i-1} L_{i-1}' + [-1, \theta_i']' r_i^{-1} [-1, \theta_i'], \qquad (9.35)\\
&\qquad \tilde{L}_i = \begin{bmatrix} 1 & 0\\ {}^{\lfloor 1\psi}\tilde{L}_i & L_i \end{bmatrix},\ \tilde{D}_i = \mathrm{diag}\left[{}^{\lfloor 1}\tilde{D}_i, D_i\right],\ {}^{\lfloor 1}\tilde{D}_i \textit{ is scalar}\\
&\qquad k_i = k_{i-1} + \ln\left(r_i\, {}^{\lfloor 1}\tilde{D}_i\right),\ \mathring{D}_i = \mathring{D}_{i-1} - 1.\\
&\textit{end} \quad \textit{of the cycle over } i
\end{aligned}
$$

The updating of the LDL' (!) decomposition can be done by Algorithm 8.1.2.

Proof: The chain rule for expectation, Proposition 2.3.6, implies the need to evaluate, for $i = \mathring{\Delta}+1,\ldots,\mathring{d}$,

$$\begin{aligned}
&\int (2\pi r_i)^{-0.5} \exp\left\{-0.5\left(k_{i-1} + [u,\psi_i'] \left([-1,\theta_i']' r_i^{-1}[-1,\theta_i'] + L_{i-1}D_{i-1}L_{i-1}'\right)[u,\psi_i']'\right)\right\} du \equiv \\
\equiv\ &\int (2\pi r_i)^{-0.5} \exp\left\{-0.5\left(k_{i-1} + [u,\psi_i']\tilde{L}_i\tilde{D}_i\tilde{L}_i'[u,\psi_i']'\right)\right\} du = \\
=\ &\exp\left[-0.5\left(k_{i-1} + \ln\left(r_i\, {}^{\lfloor 1}\tilde{D}_i\right) + \psi_i' L_i D_i L_i' \psi_i\right)\right].
\end{aligned}$$

It remains to recall that $\psi_{\mathring{d}} \equiv \phi$. □

Within the consider framework, the evaluation of the KL distances reduces manipulations with expected quadratic forms. In the computationally advantageous factorized version, the correct way of computing the second moment for mixture models can easily be overlooked. This and use in special design versions make us to evaluate this moment on the component basis.

Proposition 9.1.9 (Expected quadratic form for normal mixtures with matrix components) *Let L be a lower triangular matrix with unit diagonal and D be a diagonal matrix with non-negative entries. Let the normal mixture with matrix components describe the evolution of the state vector in the phase form. Then, it holds*

$$\begin{aligned}
\mathcal{E}[\phi_t' LDL'\phi_t|\psi_t,\Theta] &= \sum_{c\in c^*}\left[\psi_t' A_c' LDL' A_c \psi_t + \alpha_c \mathrm{tr}\left({}^{\lfloor\Delta}L\, {}^{\lfloor\Delta}D\, {}^{\lfloor\Delta}L' r_c\right)\right] \qquad (9.36)\\
A_c &\equiv \sqrt{\alpha_c}\begin{bmatrix}\theta_c' \\ \Lambda\end{bmatrix},\quad \Lambda \equiv \begin{bmatrix} I_{\mathring{\psi}-\mathring{\Delta}-1} & 0_{\mathring{\psi}-\mathring{\Delta}-1,\mathring{\Delta}+1} & 0 \\ 0_{1,\mathring{\psi}-\mathring{\Delta}-1} & 0_{1,\mathring{\Delta}+1} & 1\end{bmatrix},\ \textit{where}\\
L &= \begin{bmatrix} {}^{\lfloor\Delta}L & 0 \\ {}^{\lfloor\Delta\psi}L & {}^{\lfloor\psi}L\end{bmatrix},\ D = \begin{bmatrix} {}^{\lfloor\Delta}D & 0 \\ 0 & {}^{\lfloor\psi}D\end{bmatrix},\ {}^{\lfloor\Delta}L,\ {}^{\lfloor\Delta}D\ \textit{are}\ (\mathring{\Delta},\mathring{\Delta})\textit{-matrices.}
\end{aligned}$$

Proof: It exploits the phase form of the state, known moments of components and identities: $\mathcal{E}[xx'] = \mathcal{E}[x]\mathcal{E}[x'] + \mathrm{cov}(x)$, $\mathcal{E}[x'Qx] = \mathrm{tr}[Q\mathrm{cov}(x)]$ valid for any vector and any symmetric compatible kernel Q. □

Now we are ready to address the fact that the individual predictors forming the optimized pdf ${}^{\lfloor I}f(d(\mathring{t}))$ are finite mixtures. This makes evaluation of the KL distance $\mathcal{D}\left({}^{\lfloor I}f||\,{}^{\lfloor U}f\right)$ and consequently the application of Proposition 2.4.4 impossible. For this reason, we follow the direction outlined in Chapter 7 and use the Jensen inequality (2.14) for finding an upper bound on the KL distance. It gives us a chance to optimize at least this upper bound. Its structure is demonstrated on the matrix form of the involved normal pdfs.

Proposition 9.1.10 (The J distance of a mixture to ${}^{\lfloor U}f$ for normal pdfs) *Let us consider the joint pdf on observed data $d(\mathring{t}) \in d^*(\mathring{t})$*

$$f(d(\mathring{t})) \equiv \prod_{t\in t^*}\sum_{c\in c^*}\alpha_{c;t}\mathcal{N}_{d_t}(\theta_c'\phi_{t-1}, r_c),\ \textit{where}$$

the matrix parameters θ_c, r_c of normal components as well as their, possibly past-data-dependent, probabilistic weights $\alpha_{c;t}$ are known. Let ${}^{\lfloor U}f(d(\mathring{t})) = \prod_{t\in t^*}\mathcal{N}_{d_t}\left({}^{\lfloor U}\theta'\phi_{t-1},\, {}^{\lfloor U}r\right)$ be another known normal pdf. If need be, the matrix regression coefficients θ_c in all components as well as in ${}^{\lfloor U}\theta$ are complemented by zeros so that the state vector ϕ_{t-1} is common to all of them. Then, the following inequality holds

$$\begin{aligned}
\mathcal{D}\left(f||\,{}^{\lfloor U}f\right) &\le \mathcal{J}\left(f||\,{}^{\lfloor U}f\right) \equiv 0.5\mathcal{E}\left\{\sum_{t\in t^*}\alpha_t'\omega(\phi_{t-1})\right\}\quad \textit{with}\\
\omega(\phi_{t-1}) &\equiv [\omega(1,\phi_{t-1}),\ldots,\omega(\mathring{c},\phi_{t-1})]'\\
\omega(c,\phi_{t-1}) &\equiv 2\int f(d_t|d(t-1),c)\ln\left(\frac{f(d_t|d(t-1),c)}{{}^{\lfloor U}f(d_t|d(t-1))}\right) dd_t = k_c + \phi_{t-1}' L_c D_c L_c' \phi_{t-1},\ \textit{where}\\
k_c &\equiv \ln\left|{}^{\lfloor U}r r_c^{-1}\right| - \mathring{d} + \mathrm{tr}\left[r_c\, {}^{\lfloor U}r^{-1}\right],\quad L_c D_c L_c' \equiv \left(\theta_c - {}^{\lfloor U}\theta\right){}^{\lfloor U}r^{-1}\left(\theta_c - {}^{\lfloor U}\theta\right)'.
\end{aligned}$$

The J distance $\mathcal{J}\left(f||\,^{\lfloor U}f\right)$ *is non-negative and equal to zero iff* $r_c = \,^{\lfloor U}r$ *and* $\theta'_c\phi_{t-1} = \,^{\lfloor U}\theta'\phi_{t-1}\ \forall t \in t^*$ *and* $c \in c^*$ *for which* $\alpha_{c;t} > 0$.

Proof: It is implied by a direct combination of Propositions 7.1.5, 8.1.10. □

9.2 Design of advising strategy

Here, design variants of the academic, industrial and simultaneous advisory systems are discussed in the special case of normal components and the normal user ideal pdf.

9.2.1 Academic design

The recommended pointers to components c_t are the actions of the academic p-system. They are described by the optimized academic strategy $\left\{ ^{\lfloor I}f(c_t|d(t-1))\right\}_{t\in t^*}$. It determines, together with the estimated model of the o-system, the ideal pdfs

$$^{\lfloor I}f(d_t, c_t|d(t-1)) \equiv \prod_{i=1}^{\mathring{d}} \mathcal{N}_{d_{ic_t;t}}\left(\theta'_{ic_t}\psi_{ic_t;t}, r_{ic_t}\right)\,^{\lfloor I}f(c_t|d(t-1)) \text{ with } d_t \equiv \Delta_t. \tag{9.37}$$

For the adopted fully probabilistic design, we have to specify the user ideal pdf. We assume its normality on the o-data $d^*_{o;t}$. This true user ideal pdf is extended on d^*_t in the way discussed in Section 5.1.3. It is extended further on c^*_t by specifying the target pf $^{\lfloor U}f(c_t|d(t-1))$ for the academic advices, i.e. for the recommended pointers c_t.

Always, we evaluate initially dangerous components, see Agreement 5.4.6 and Section 9.1.1, and reduce the support of $^{\lfloor U}f(c_t|d(t-1))$ to non-dangerous components. The optimal design of the academic p-system is solved by this choice if the reduced set contains just a single component. This component is presented to the operator as the designed ideal pdf. Otherwise, the normal version of the optimal advising academic strategy described by Proposition 7.2.1 has to be searched for.

Proposition 9.2.1 (Academic advisory system minimizing γ-bound on the KL distance) *Let us consider the academic design for the o-system described by the mixture with normal components (9.1) having the state ϕ in the phase form, Agreement 9.0.1. Lack of recognizable actions makes the innovation Δ_t and the data record d_t coincide.*

Let $d_t = (d_{o;t}, d_{p+;t})$ =(o-data, surplus p-data) and the user ideal pdf $^{\lfloor U}f(d(\mathring{t}))$ on $d^*(\mathring{t})$ be defined by

$$^{\lfloor U}f(d_t|d(t-1)) \equiv \prod_{i_o\in i^*_o} \mathcal{N}_{d_{i;t}}\left(\,^{\lfloor U}\theta'_i\psi_{i;t},\ ^{\lfloor U}r_i\right)\,^{\lfloor I}f(d_{p+;t}|d(t-1)),\ i^*_o \equiv \{1,\ldots,\mathring{d}_o\} \subset i^* \equiv \{1,\ldots,\mathring{d}\}. \tag{9.38}$$

The parameters $\Theta_{ic} = [\theta_{ic}, r_{ic}]$ of the mixture model as well those determining the normal user ideal pdf $^{\lfloor U}\Theta_i \equiv \left[\,^{\lfloor U}\theta_i,\ ^{\lfloor U}r_i\right]$ are assumed to be known. Regression coefficients θ_{ic}, $^{\lfloor U}\theta_i$ are complemented by zeros so that the corresponding factors of the user ideal and the mixture model have the common regression vectors $\psi_{i;t} \equiv \left[d_{i+1;t}, \psi'_{i+1;t}\right]' = \left[d'_{(i+1)\ldots\mathring{d};t}, \phi'_{t-1}\right]'$, $i < \mathring{d}$, $\psi_{\mathring{d};t} \equiv \phi_{t-1}$.

The recommended pointers c_t are allowed to have non-zero values at most for indices in c^* that point to non-dangerous components, Agreement 5.4.6.

Let us search for the causal advising strategy $\{d^*(t-1) \to c_t \in c^*\}_{t\in t^*}$, that, at least approximately, minimizes the KL distance of $^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ to the user ideal

$$^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) = \prod_{t\in t^*}\,^{\lfloor U}f(d_t|d(t-1))\,^{\lfloor U}f(c_t|d(t-1)).$$

In this description, the definition (9.38) is used and the user ideal pf on c_{t+1} is considered in the form

$$\begin{aligned} ^{\lfloor U}f(c_{t+1}|d(t)) &\propto\ ^{\lfloor U}f(c_{t+1})\exp\left[-0.5\,^{\lfloor U}\omega_\gamma(c_{t+1}, \phi_t)\right] && (9.39)\\ ^{\lfloor U}f(c_{t+1}) &\ \text{is a probabilistic vector} \\ ^{\lfloor U}\omega(c_{t+1}, \phi_t) &\equiv\ ^{\lfloor U}k_{c_{t+1};t} + \phi'_t\,^{\lfloor U}L_{c_{t+1};t}\,^{\lfloor U}D_{c_{t+1};t}\,^{\lfloor U}L'_{c_{t+1};t}\phi_t. && (9.40)\end{aligned}$$

The fixed, data independent, part $^{\lfloor U}f(c_{t+1})$ *serves for the specification of support on non-dangerous components. The lifts* $^{\lfloor U}k_{c;t}$ *and kernels* $^{\lfloor U}L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}$*, determined by the lower triangular matrices with unit diagonal* $^{\lfloor U}L_{c_{t+1};t}$ *and the diagonal matrices* $^{\lfloor U}D_{c_{t+1};t}$*, are assumed to be data independent. Mostly, they are even time-invariant. The lifted quadratic form determines a preferable user strategy of selecting pointers* c_{t+1} *to mixture components.*

The LDL' *decomposition of kernels used below are split*

$$L \equiv \begin{bmatrix} 1 & 0 \\ ^{\lfloor d\psi}L & ^{\lfloor \psi}L \end{bmatrix}, \quad D \equiv \mathrm{diag}\left[{}^{\lfloor d}D,\ {}^{\lfloor \psi}D\right], \quad \textit{where } {}^{\lfloor d}D \textit{ is scalar.}$$

Let us consider the following strategy, $c \in c^*$,

$$\begin{aligned} {}^{\lfloor I}f(c_t|\phi_{t-1}) &\propto {}^{\lfloor U}f(c_t)\exp[-0.5\omega_\gamma(c_t,\phi_{t-1})], \quad \textit{where} \\ \omega_\gamma(c_t,\phi_{t-1}) &\equiv k_{c_t;t-1} + \phi'_{t-1}L_{c_t;t-1}D_{c_t;t-1}L'_{c_t;t-1}\phi_{t-1} \end{aligned} \tag{9.41}$$

The lifts $k_{c_t;t-1}$ *and kernels* $L_{c_t;t-1}D_{c_t;t-1}L'_{c_t;t-1}$, $c \in c^*$, *are evaluated against time course* $t = \mathring{t}, \mathring{t}-1,\ldots,1$, *with the initial values* $k_{c;\mathring{t}} = 0$, $L_{c;\mathring{t}} = I_{\mathring{\phi}} =$ *unit matrix,* $D_{c;\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$.

$$\begin{array}{l}
\textit{For} \quad t = \mathring{t},\ldots,1 \\
\quad \textit{For} \quad c = 1,\ldots,\mathring{c} \\
\qquad\qquad \textit{Create the common initial conditions of recursions over factors} \\
\qquad\qquad k_{0c} \equiv -\mathring{d}_o \equiv k_a \equiv \text{average lift} \\
\qquad\qquad L_{0c} = I_{\mathring{\Psi}},\ D_{0c} = 0_{\mathring{\Psi},\mathring{\Psi}} \equiv L_aD_aL'_a \equiv \text{average kernel} \\
\qquad \textit{For} \quad \tilde{c} = 1,\ldots,\mathring{c} \\
\qquad\qquad \beta_{\tilde{c}} \propto {}^{\lfloor U}f(\tilde{c})\exp(-0.5k_{\tilde{c};t}) \qquad\qquad (9.42) \\
\qquad\qquad L_{0c}D_{0c}L'_{0c} \equiv L_{0c}D_{0c}L'_{0c} + \beta_{\tilde{c}}\tilde{L}_{\tilde{c};t}\tilde{D}_{\tilde{c};t}\tilde{L}'_{\tilde{c};t}. \\
\qquad\qquad \textit{Terms } \tilde{L}_{\tilde{c};t},\ \tilde{D}_{\tilde{c};t} \textit{ are specified using the split} \\
\qquad\qquad L_{\tilde{c};t} \equiv \begin{bmatrix} {}^{\lfloor \psi 0}L_{\tilde{c};t} & 0 \\ {}^{\lfloor \psi 1}L'_{\tilde{c};t} & 1 \end{bmatrix},\ D_{\tilde{c};t} = \mathrm{diag}\left[{}^{\lfloor \psi 0}D_{\tilde{c};t},\ {}^{\lfloor 1}D_{\tilde{c};t}\right],\ {}^{\lfloor 1}D_{\tilde{c};t} \textit{ is scalar.} \\
\qquad\qquad \tilde{L}_{\tilde{c};t} \equiv \begin{bmatrix} {}^{\lfloor \psi 0}L_{\tilde{c};t} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor \psi 1}L'_{\tilde{c};t} & 0 & 1 \end{bmatrix},\ \tilde{D}_{\tilde{c};t} = \mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor \psi 0}D_{\tilde{c};t}\right], 0_{1,\mathring{d}}, {}^{\lfloor 1}D_{\tilde{c};t}\right]. \\
\qquad \textit{end} \quad \textit{of the cycle over } \tilde{c} \\
\qquad \textit{For} \quad i = 1,\ldots,\mathring{d} \\
\qquad\qquad L_{ic}D_{ic}L'_{ic} = {}^{\lfloor \psi}L_{(i-1)c}\,{}^{\lfloor \psi}D_{(i-1)c}\,{}^{\lfloor \psi}L'_{(i-1)c} + \\
\qquad\qquad + \left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right){}^{\lfloor d}D_{(i-1)c}\left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right)' + \\
\qquad\qquad + \chi\left(i \le \mathring{d}_o\right)\left(\theta_{ic} - {}^{\lfloor U}\theta_i\right){}^{\lfloor U}r_i^{-1}\left(\theta_{ic} - {}^{\lfloor U}\theta_i\right)' \\
\qquad\qquad k_{ic} = k_{(i-1)c} + {}^{\lfloor d}D_{(i-1)c}r_{ic} + \chi\left(i \le \mathring{d}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right) + \frac{r_{ic}}{{}^{\lfloor U}r_i}\right]. \\
\qquad \textit{end} \quad \textit{of the cycle over } i \\
\qquad\qquad L_{c;t-1}D_{c;t-1}L'_{c;t-1} \equiv L_{\mathring{d}c}D_{\mathring{d}c}L'_{\mathring{d}c} + {}^{\lfloor U}L_{c;t-1}\,{}^{\lfloor U}D_{c;t-1}\,{}^{\lfloor U}L'_{c;t-1} \\
\qquad\qquad k_{c;t-1} \equiv k_{\mathring{d}c} + {}^{\lfloor U}k_{c;t-1} \\
\quad \textit{end} \quad \textit{of the cycle over } c \\
\textit{end} \quad \textit{of the cycle over } t
\end{array}$$

Then, this strategy minimizes the KL distance loss for the design horizon equal one. For longer horizons, it minimizes the upper bound of the γ*-type, see Proposition 7.1.8, on the KL distance as it* *replaces the correct Bellman function*

$$-\ln\left\{\sum_{c_{t+1}\in c^*} {}^{\lfloor U}f(c_{t+1})\exp[-0.5\omega_\gamma(c_{t+1},\phi_t)]\right\} \text{ by the larger value} \tag{9.43}$$
$$\frac{1}{2}\sum_{c_{t+1}\in c^*}\beta_{c_{t+1}}\phi_t' L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}\phi_t + \text{data and strategy independent constant.}$$

Proof: Let us assume that the Bellman function is $-\ln(\gamma(\phi_t))$, with

$$\begin{aligned}\gamma(\phi_t) &= \exp\left[-0.5\sum_{c_{t+1}\in c^*}\beta_{c_{t+1}}\phi_t' L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}\phi_t\right], \text{ where}\\ \beta_{c_{t+1}} &\propto {}^{\lfloor U}f(c_{t+1})\exp[-0.5k_{c_{t+1};t}].\end{aligned}$$

The kernels $L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}$ and the lifts $k_{c_{t+1};t}$ are conjectured to be independent of data.

This conjecture is true for $t=\mathring{t}$ with $k_{c_{\mathring{t}+1};\mathring{t}}=0,\ L_{c_{\mathring{t}+1};\mathring{t}}=I_{\mathring{\phi}},\ D_{c_{\mathring{t}+1};\mathring{t}}=0_{\mathring{\phi},\mathring{\phi}}$. We prove that this form is preserved during the backward induction for $t<\mathring{t}$, if the approximation (9.43) is used. It gives us also the algorithmic solution of the inspected problem.

We re-write the assumed form of $\gamma(\phi_t)$ into the quadratic form in $\Psi_t\equiv\psi_{0;t}$, taking into account the phase form of the state $\phi_t=\mathcal{K}'\Psi_t$, see (9.33). This implies

$$\begin{aligned}\gamma(\phi_t) &= \exp\left[-0.5\psi_{0;t}'L_0D_0L_0'\psi_{0;t}\right] \text{ with}\\ L_0D_0L_0' &\equiv \sum_{c_{t+1}\in c^*}\beta_{c_{t+1}}\mathcal{K}L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}\mathcal{K}' \ \textit{average kernel.}\end{aligned}$$

Algorithm 9.1.8 implies the explicit expression of the kernel

$$\begin{aligned}L_0D_0L_0' &\equiv \sum_{c_{t+1}\in c^*}\beta_{c_{t+1}}\tilde{L}_{c_{t+1};t}\tilde{D}_{c_{t+1};t}\tilde{L}'_{c_{t+1};t}\\ \tilde{L}_{c;t} &\equiv \begin{bmatrix} {}^{\lfloor\psi 0}L_{c;t} & 0 & 0\\ 0 & I_{\mathring{d}} & 0\\ {}^{\lfloor\psi 1}L'_{c;t} & 0 & 1\end{bmatrix},\quad \tilde{D}_{c;t}=\mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor\psi 0}D_{c;t}\right],0_{1,\mathring{d}},\ {}^{\lfloor 1}D_{c;t}\right].\end{aligned}$$

The t-th step of dynamic programming, with $\mathring{\Delta}\equiv\mathring{d}$, becomes

$$\min_{\left\{{}^{\lfloor I}f(c_t|d(t-1))\right\}}\ \sum_{c_t\in c^*}\ {}^{\lfloor I}f(c_t|d(t-1))\left[0.5\left(\tilde{\omega}_\gamma(c_t,\phi_{t-1})+\underbrace{{}^{\lfloor U}\omega(c_t,\phi_{t-1})}_{\text{see (9.40)}}\right)+\ln\left(\frac{{}^{\lfloor I}f(c_t|d(t-1))}{{}^{\lfloor U}f(c_t)}\right)\right]$$
$$\tilde{\omega}_\gamma(c_t,\phi_{t-1}) \equiv 2\int f(d_t|\phi_{t-1},c_t)\ln\left(\frac{f(d_{o;t}|d_{p+;t},\phi_{t-1},c_t)}{\gamma(\phi_t)\ {}^{\lfloor U}f(d_{o;t}|\phi_{t-1})}\right)dd_t.$$

The definition $\omega_\gamma(c_t,\phi_{t-1})\equiv\tilde{\omega}_\gamma(c_t,\phi_{t-1})+{}^{\lfloor U}\omega(c_t,\phi_{t-1})$ and use of the Proposition 9.1.7 imply

$$\begin{aligned}\omega_\gamma(c_t,\phi_{t-1}) &= k_{\mathring{d}c_t;t-1}+{}^{\lfloor U}k_{c_t;t-1}+\phi_{t-1}'L_{c_t;t-1}D_{c_t;t-1}L'_{c_t;t-1}\phi_{t-1},\\ L_{c;t-1}D_{c;t-1}L'_{c;t-1} &\equiv L_{\mathring{d}c}D_{\mathring{d}c}L'_{\mathring{d}c}+{}^{\lfloor U}L_{c;t-1}\,{}^{\lfloor U}D_{c;t-1}\,{}^{\lfloor U}L'_{c;t-1},\ \text{ where}\\ L_{ic}D_{ic}L'_{ic} &= {}^{\lfloor\psi}L_{(i-1)c}\,{}^{\lfloor\psi}D_{(i-1)c}\,{}^{\lfloor\psi}L'_{(i-1)c}+\left(\theta_{ic}+{}^{\lfloor d\psi}L_{(i-1)c}\right){}^{\lfloor d}D_{(i-1)c}\left(\theta_{ic}+{}^{\lfloor d\psi}L_{(i-1)c}\right)'+\\ &+\ \chi\left(i\le\mathring{d}_o\right)\left(\theta_{ic}-{}^{\lfloor U}\theta_i\right){}^{\lfloor U}r_i^{-1}\left(\theta_{ic}-{}^{\lfloor U}\theta_i\right)',\\ k_{ic} &= k_{(i-1)c}+{}^{\lfloor d}D_{(i-1)c}r_{ic}+\chi\left(i\le\mathring{d}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right)+\frac{r_{ic}}{{}^{\lfloor U}r_i}\right],\ i=1,\ldots,\mathring{d},\end{aligned}$$

with the *common initial conditions*

$$L_{0c} = L_0,\ D_{0c}=D_0,\ \ k_{0c}=-\mathring{d}_o+k_0.$$

The minimizing argument is $f(c_t|\phi_{t-1}) = {}^{\lfloor U}f(c_t)\exp[-0.5\omega_\gamma(c_t,\phi_{t-1})]/\tilde{\gamma}(\phi_{t-1})$, where $\tilde{\gamma}(\phi_{t-1})$ is normalizing factor, i.e.

$$\tilde{\gamma}(\phi_{t-1}) = q\sum_{c_t\in c^*}\beta_{c_t}\exp\left[-0.5\phi_{t-1}L_{c_t;t-1}D_{c_t;t-1}L'_{c_t;t-1}\phi_{t-1}\right] \text{ with } q\equiv\sum_{c_t\in c^*}{}^{\lfloor U}f(c_t)\exp[-0.5k_{c_t;t-1}].$$

The form of the reached minimum $-\ln(\tilde{\gamma}(\phi_{t-1}))$ differs from that assumed and cannot be practically used further on. The inequality between weighted arithmetic and geometric means implies that

$$\tilde{\gamma}(\phi_{t-1}) \geq q\exp\left[-0.5\sum_{c_t\in c^*}\beta_{c_t}\phi'_{t-1}L_{c;t-1}D_{c;t-1}L'_{c;t-1}\phi_{t-1}\right]\equiv\gamma(\phi_{t-1}), \text{ where}$$

the right-hand side has the same form as that assumed for $\gamma(\phi_t)$. The Bellman function $-\ln(\tilde{\gamma}(\phi))$ is decreasing function of the values $\tilde{\gamma}(\phi_t)$. Consequently, the adopted approximation (9.43), that replaces the function $\tilde{\gamma}(\phi_t)$ by the function $\gamma(\phi_t)$, bounds the minimized functional from above while preserving the advantageous analytical form. $\square$

The proved proposition combined with the certainty-equivalent strategy gives the following algorithm.

Algorithm 9.2.1 (Fixed academic advisory system minimizing γ-bound on the KL distance)

Initial (off line) mode

- *Estimate normal mixture model of the o-system with the state ϕ_t, Chapter 8 and use the point estimates of parameters in definition of the individual components $\prod_{i=1}^{\mathring{d}}\mathcal{N}_{d_{ic}}(\theta'_{ic}\psi_{ic}, r_{ic})$, $c\in c^*$.*
- *Evaluate the steady state behavior of individual components, Section 9.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if $\mathring{c}=0$.*
- *Take the non-dangerous component as the ideal pdf offered to the user and go to* Sequential mode *if $\mathring{c}=1$.*
- *Specify the true user ideal pdf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{d}_o}\mathcal{N}_{d_{i;t}}\left({}^{\lfloor U}\theta'_i\psi_{i;t},\, {}^{\lfloor U}r_i\right).$$

- *Specify* data and time invariant *part ${}^{\lfloor U}f(c_t)$ of the user ideal on the recommended pointers. It is zero on dangerous components.*
- *Specify* data and time invariant lifts ${}^{\lfloor U}k_{c_t}$ *and* kernels *of the user KL distance ${}^{\lfloor U}L_{c_t}$, ${}^{\lfloor U}D_{c_t}$. It completes definition of the user ideal pf ${}^{\lfloor U}f(c_t|d(t-1))$, see (9.39).*
- *Select the length of the design horizon $\mathring{t}\geq 1$.*
- *Initialize the iterative mode by setting $L_{c;\mathring{t}} = I_{\mathring{\phi}}$, $D_{c;\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$, $k_{c;\mathring{t}}=0$, $c\in c^*$.*

Iterative (off line) mode

- *Correct the quadratic forms, for $t=\mathring{t},\mathring{t}-1,\ldots,1$, according to the formulas (9.42).*
- *Take the terminal characteristics as those describing the approximate optimal* steady-state strategy, *Chapter 3,*

$$k_c\equiv k_{c;1},\ L_c\equiv L_{c;1},\ D_c\equiv D_{c;1},\ c\in c^*.$$

Sequential (on line) mode, *running for t=1,2,. . . ,*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*

2. *Evaluate the values of the pf describing the optimal academic advising strategy*

$$^{\lfloor I}f(c_{t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}) \exp\left[-0.5\left(k_{c_{t+1}} + \phi_t' L_{c_{t+1}} D_{c_{t+1}} L_{c_{t+1}}' \phi_t\right)\right].$$

3. *Present to the user projections of the ideal pdf*

$$^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1}\in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) \prod_{i=1}^{\mathring{d}} \mathcal{N}_{d_{ic_{t+1};t+1}}\left(\theta_{ic_{t+1}}' \psi_{ic_{t+1};t}, r_{ic_{t+1}}\right).$$

4. *Go to the beginning of* Sequential mode.

Remark(s) 9.2.1

1. *The solution is prepared for the numerically safe factorized implementation.*

2. *The chosen form $^{\lfloor U}f(c_t|d(t-1))$ that coincides with that of the optimal advising strategy allows us to stay within the computational advantageous class while being able to*
   - *exclude a priori bad (dangerous) components, Agreement 5.4.6,*
   - *respect the "natural" requirement that advices should not be changed too quickly,*
   - *employ the IST strategy, Algorithm 7.2.4, in an adaptive implementation of the above strategy.*

3. *The redundant specification of probabilities $^{\lfloor U}f(c)$ and lifts k_c simplifies interpretation: $^{\lfloor U}f(c)$ cares about the support of the desired recommended pointers. The lifts k_c distinguish data-invariant absolute values of individual probabilities. The user-specified KL kernel given by $^{\lfloor U}L_c$, $^{\lfloor U}D_c$ prescribes data-dependent preferences among the recommended pointers.*

4. *All data invariant parts in the conditional KL distances have to be concentrated into the lift including that occurring in the quadratic form written in terms of the state vectors with unit as its entry. Ideally, the updating variant described by Algorithm 9.1.7 should be used.*

5. *The* grouped variant of advising, *Proposition 7.2.4, can be designed and used. Essentially, the initial conditions k_{0c}, L_{0c}, D_{0c} are constructed as described above only each n-th step. In intermediate recursions, $k_{0c} = k_{c;t+1}$, $L_{0c} = L_{c;t+1}$, $D_{0c} = D_{c;t+1}$.*

6. *The last two remarks apply for all advising algorithms discussed in this chapter.*

The on line use of the fixed advisory system is extremely simple. It requires just evaluation of values of the normal-mixture predictor. Naturally, the quality of its advices heavily depends on the quality of the used mixture model. This makes us to write down an adaptive version of the above algorithm. The increased complexity can be well counteracted by using the IST patch, Algorithm 7.2.4. The lifts $k_{c;t}$ and the kernels $L_{c;t}D_{c;t}L_{c;t}'$ of quadratic forms obtained at time t seem to be "natural" parameter ϑ needed: they are used as initial conditions for the design made at time $t+1$. For the IST strategy, the receding horizon $T = 1$ is expected to be mostly sufficient.

Let us write down the adaptive algorithm with receding-horizon, certainty-equivalent strategy complemented by the IST patch.

Algorithm 9.2.2 (Adaptive academic advising minimizing γ-bound on the KL distance)
Initial (off line) mode

- *Estimate normal mixture model of the o-system with the state ϕ_t, Chapter 8.*
- *Evaluate the steady state behavior of individual components, Section 9.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*

- *Return to the learning phase if all components are dangerous, i.e. if* $\mathring{c} = 0$.
- *Take the non-dangerous component as the ideal pdf offered to the user and skip* Iterative mode *if* $\mathring{c} = 1$.
- *Specify the true user ideal pdf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{d}_o} \mathcal{N}_{d_{i;t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\ {}^{\lfloor U}r_i\right).$$

- *Specify* data and time invariant *part* $^{\lfloor U}f(c_t)$ *of the user ideal on the recommended pointers. It is zero on dangerous components.*
- *Specify* data invariant lifts $^{\lfloor U}k_{c_t;t-1}$ *and* kernels *of the user KL distance* $^{\lfloor U}L_{c_t;t-1}$, $^{\lfloor U}D_{c_t;t-1}$. *It completes definition of the user ideal pf*

$$^{\lfloor U}f(c_t|d(t-1)) \propto {}^{\lfloor U}f(c_t)\exp\left[-0.5\left(k_{c_t;t-1} + \phi_{t-1}'L_{c_t;t-1}D_{c_t;t-1}L_{c_t;t-1}'\phi_{t-1}\right)\right], \text{ see (9.39).}$$

- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update estimates of the model parameters, Section 8.5.*
3. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $L_{c;\tau} = I_{\mathring{\phi}}$, $D_{c;\tau} = 0_{\mathring{\phi},\mathring{\phi}}$, $k_{c;\tau} = 0$. *The initialization of* L, D, k *is skipped for* $t > 1$ *if the IST strategy is used.*

Iterative mode

Correct the following quadratic forms using the current point estimates of the factor parameters

For $\tau = t+T, \ldots, t+1$

For $c = 1, \ldots, \mathring{c}$

Create the common initial conditions of recursions over factors

$k_{0c} \equiv -\mathring{d}_o \equiv k_a \equiv$ average lift

$L_{0c} = I_{\mathring{\Psi}}$, $D_{0c} = 0_{\mathring{\Psi},\mathring{\Psi}} \equiv L_a D_a L_a' \equiv$ average kernel

For $\tilde{c} = 1, \ldots, \mathring{c}$

$\beta_{\tilde{c}} \propto {}^{\lfloor U}f(\tilde{c})\exp(-0.5k_{\tilde{c};\tau})$

$L_{0c}D_{0c}L_{0c}' \equiv L_{0c}D_{0c}L_{0c}' + \beta_{\tilde{c}}\tilde{L}_{\tilde{c};\tau}\tilde{D}_{\tilde{c};\tau}\tilde{L}_{\tilde{c};\tau}'$.

Terms $\tilde{L}_{\tilde{c};\tau}$, $\tilde{D}_{\tilde{c};\tau}$ *are specified using the split*

$$L_{\tilde{c};\tau} \equiv \begin{bmatrix} ^{\lfloor \psi 0}L_{\tilde{c};\tau} & 0 \\ ^{\lfloor \psi 1}L_{\tilde{c};\tau}' & 1 \end{bmatrix}, \quad D_{\tilde{c};\tau} = \mathrm{diag}\left[{}^{\lfloor \psi 0}D_{\tilde{c};\tau},\ {}^{\lfloor 1}D_{\tilde{c};\tau}\right], \quad {}^{\lfloor 1}D_{\tilde{c};\tau} \text{ is scalar.}$$

$$\tilde{L}_{\tilde{c};\tau} \equiv \begin{bmatrix} ^{\lfloor \psi 0}L_{\tilde{c};\tau} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ ^{\lfloor \psi 1}L_{\tilde{c};\tau}' & 0 & 1 \end{bmatrix}, \quad \tilde{D}_{\tilde{c};\tau} = \mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor \psi 0}D_{\tilde{c};\tau}\right], 0_{1,\mathring{d}},\ {}^{\lfloor 1}D_{\tilde{c};\tau}\right].$$

end of the cycle over $\tilde{c}$

For $i = 1, \ldots, \mathring{d}$

$$L_{ic}D_{ic}L_{ic}' = {}^{\lfloor \psi}L_{(i-1)c}\,{}^{\lfloor \psi}D_{(i-1)c}\,{}^{\lfloor \psi}L_{(i-1)c}' +$$
$$+\left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right){}^{\lfloor d}D_{(i-1)c}\left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right)' +$$
$$+\chi\left(i \leq \mathring{d}_o\right)\left(\theta_{ic} - {}^{\lfloor U}\theta_i\right){}^{\lfloor U}r_i^{-1}\left(\theta_{ic} - {}^{\lfloor U}\theta_i\right)'$$

$$k_{ic} = k_{(i-1)c} + {}^{\lfloor d}D_{(i-1)c} r_{ic} + \chi\left(i \le \mathring{d}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right) + \frac{r_{ic}}{{}^{\lfloor U}r_i}\right].$$

end of the cycle over i

$$L_{c;\tau-1} D_{c;\tau-1} L'_{c;\tau-1} \equiv L_{\mathring{d}c} D_{\mathring{d}c} L'_{\mathring{d}c} + {}^{\lfloor U}L_{c;\tau-1}\, {}^{\lfloor U}D_{c;\tau-1}\, {}^{\lfloor U}L'_{c;\tau-1}$$

$$k_{c;\tau-1} \equiv k_{\mathring{d}c} + {}^{\lfloor U}k_{c;\tau-1}$$

end of the cycle over c

end of the cycle over τ

4. *Evaluate the pf describing the optimal academic advising strategy*

$$^{\lfloor I}f(c_{t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}) \exp\left[-0.5(k_{c_{t+1};t} + \phi'_t L_{c_{t+1};t} D_{c_{t+1};t} L'_{c_{t+1};t} \phi_t)\right].$$

5. *Present to the user projections of the ideal pdf*

$$^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1} \in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) \prod_{i \in i^*} \mathcal{N}_{d_{ic_{t+1};t+1}}\left(\theta'_{ic_{t+1}} \psi_{ic_{t+1};t}, r_{ic_{t+1}}\right).$$

6. *Go to the beginning of* Sequential mode.

For completeness, let us write down a version of this algorithm that selects the most probable recommended pointer as its action in its application step. Within the iterative evaluations, the maximizer $^{\lfloor I}c_{\tau+1}$ of $^{\lfloor I}f(c_{\tau+1}|\phi_\tau)$ is selected under simplifying assumption that $\phi_\tau = \phi_t$.

This assumption allows us to preserve computational feasibility. It brings no additional approximation when using the IST strategy with the horizon $T = 1$.

Algorithm 9.2.3 (Adaptive choice of the most probable academic advices)
Initial (off line) mode

- *Estimate normal mixture model of the o-system with the state* ϕ_t, *Chapter 8.*
- *Evaluate the steady state behavior of individual components, Section 9.1.1.*
- *Exclude dangerous components, Agreement 5.4.6.*
- *Return to the learning phase if all components are dangerous, i.e. if* $\mathring{c} = 0$.
- *Take the non-dangerous component as the ideal pdf offered to the user and skip* Iterative mode *if* $\mathring{c} = 1$.
- *Specify the true user ideal pdf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{d}_o} \mathcal{N}_{d_{i;t}}\left({}^{\lfloor U}\theta'_i \psi_{i;t}, {}^{\lfloor U}r_i\right).$$

- *Select the length of the receding horizon* $T \ge 1$.
- *Select randomly the pointer* $^{\lfloor I}c_{T+1} \in c^*$.

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update estimates of model parameters, Section 8.5.*

3. *Initialize the iterative mode by setting* $\tau = t+T$ *and* $L_{c;\tau} = I_{\mathring{\phi}}$, $D_{c;\tau} = 0_{\mathring{\phi},\mathring{\phi}}$, $k_{c;\tau} = 0$. *The initialization of* L, D, k *is skipped for* $t > 1$ *if the IST strategy is used.*

Iterative mode

Correct the following quadratic forms using the current point estimates of the factor parameters.

For $\tau = t+T, \ldots, t+1$

For $c = 1, \ldots, \mathring{c}$

Create the common initial conditions of recursions over factors

$$L_{0c}D_{0c}L'_{0c} \equiv \tilde{L}_{\,^{\lfloor I}c_{\tau+1};\tau}\tilde{D}_{\,^{\lfloor I}c_{\tau};\tau}\tilde{L}'_{\,^{\lfloor I}c_{\tau+1};\tau}$$

$$k_{0c} \equiv -\mathring{d}_o + k_{\,^{\lfloor I}c_{\tau+1};\tau}, \quad \textit{where}$$

$\tilde{L}_{\,^{\lfloor I}c_{\tau+1};\tau}$, $\tilde{D}_{\,^{\lfloor I}c_{\tau+1};\tau}$ *are specified using the split*

$$L_{\,^{\lfloor I}c_{\tau+1};\tau} \equiv \begin{bmatrix} ^{\lfloor \psi 0}L_{\,^{\lfloor I}c_{\tau+1};\tau} & 0 \\ ^{\lfloor \psi 1}L'_{\,^{\lfloor I}c_{\tau+1};\tau} & 1 \end{bmatrix}$$

$$D_{\,^{\lfloor I}c_{\tau+1};\tau} = \mathrm{diag}\left[^{\lfloor \psi 0}D_{\,^{\lfloor I}c_{\tau+1};\tau},\ ^{\lfloor 1}D_{\,^{\lfloor I}c_{\tau+1};\tau}\right],\ ^{\lfloor 1}D_{\,^{\lfloor I}c_{\tau+1};\tau} \textit{ is scalar.}$$

$$\tilde{L}_{\,^{\lfloor I}c_{\tau+1};\tau} \equiv \begin{bmatrix} ^{\lfloor \psi 0}L_{\,^{\lfloor I}c_{\tau+1};\tau} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ ^{\lfloor \psi 1}L'_{\,^{\lfloor I}c_{\tau+1};\tau} & 0 & 1 \end{bmatrix}$$

$$\tilde{D}_{\,^{\lfloor I}c_{\tau+1};\tau} = \mathrm{diag}\left[\mathrm{diag}\left[^{\lfloor \psi 0}D_{\,^{\lfloor I}c_{\tau+1};\tau}\right], 0_{1,\mathring{d}},\ ^{\lfloor 1}D_{\,^{\lfloor I}c_{\tau+1};\tau}\right].$$

For $i = 1, \ldots, \mathring{d}$

$$L_{ic}D_{ic}L'_{ic} = \,^{\lfloor \psi}L_{(i-1)c}\,^{\lfloor \psi}D_{(i-1)c}\,^{\lfloor \psi}L'_{(i-1)c} +$$
$$+\left(\theta_{ic} + \,^{\lfloor d\psi}L_{(i-1)c}\right)\,^{\lfloor d}D_{(i-1)c}\left(\theta_{ic} + \,^{\lfloor d\psi}L_{(i-1)c}\right)' +$$
$$+\chi\left(i \le \mathring{d}_o\right)\left(\theta_{ic} - \,^{\lfloor U}\theta_i\right)\,^{\lfloor U}r_i^{-1}\left(\theta_{ic} - \,^{\lfloor U}\theta_i\right)'$$
$$k_{ic} = k_{(i-1)c} + \,^{\lfloor d}D_{(i-1)c}r_{ic} + \chi\left(i \le \mathring{d}_o\right)\left[\ln\left(\frac{^{\lfloor U}r_i}{r_{ic}}\right) + \frac{r_{ic}}{^{\lfloor U}r_i}\right].$$

end of the cycle over i

$$L_{c;\tau-1}D_{c;\tau-1}L'_{c;\tau-1} \equiv L_{\mathring{d}c}D_{\mathring{d}c}L'_{\mathring{d}c} + \,^{\lfloor U}L_{c;\tau-1}\,^{\lfloor U}D_{c;\tau-1}\,^{\lfloor U}L'_{c;\tau-1}$$
$$k_{c;\tau-1} \equiv k_{\mathring{d}c} + \,^{\lfloor U}k_{c;\tau-1}$$

end of the cycle over c

Select $^{\lfloor I}c_\tau \equiv \mathrm{Arg}\min_{c \in c^*}\left(k_{c;\tau} + \phi'_t L_{c;\tau}D_{c;\tau}L'_{c;\tau}\phi_t\right)$ *(the simplifying assumption used!).*

end of the cycle over τ

4. *Present to the user projections of the ideal pdf*

$$^{\lfloor I}f(d_{t+1}|\phi_t) = \prod_{i=1}^{\mathring{d}} \mathcal{N}_{d_{i\,^{\lfloor I}c_{t+1};t+1}}\left(\theta'_{i\,^{\lfloor I}c_{t+1}}\psi_{i\,^{\lfloor I}c_{t+1};t+1}, r_{i\,^{\lfloor I}c_{t+1}}\right).$$

5. *Go to the beginning of* Sequential mode.

Remark(s) 9.2.2

1. *In a former formulation tried, the probabilities* α *determining component weights, Agreement 5.2.1, were taken as academic actions. The corresponding design was much more complex. Experiments indicated, however, reasonable properties of such a strategy. Thus, it might be useful to return back to this direction.*

2. *It is worth of stressing that the selected strategy tries to minimize directly the KL distance and not only its Jensen upper bound as it was needed for the previous variant.*

3. *The specification of the user KL distance determining* ${}^{\lfloor U}f(c_{t+1}|\phi_t)$ *influences substantially properties of the designed strategy. Mostly, we rely on discussion presented in Section 7.2.2. Specifically, grouped versions of algorithms are implemented, see Proposition 7.2.4 and Remark 5.*

9.2.2 Industrial design

The industrial design has to be used whenever component weights have objective meaning and cannot be influenced by the operator. Here, we address it by using ideas of Proposition 7.2.6 proved in Section 7.2.3. It should be stressed that the more effective and simpler simultaneous design should be used, see Section 9.2.3, whenever the component weights are under the operator control.

The normal components with explicitly marked recognizable actions

$$\left\{ f(\Delta_t|u_{o;t},d(t-1),c)f(u_{o;t}|d(t-1),c) \equiv \prod_{i=1}^{\mathring{\Delta}} \mathcal{N}_{\Delta_{i;t}}\left(\theta_{ic}'\psi_{ic;t},r_{ic}\right) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left(\theta_{ic}'\psi_{ic;t},r_{ic}\right) \right\}_{t\in t^*,c\in c^*} \tag{9.44}$$

and their weights $\{\alpha_c\}_{c\in c^*}$ are assumed to be known (well estimated). Here, we use the convention that the data record is ordered as in previous sections $d_t = (\Delta_{o;t}, \Delta_{p+;t}, u_{o;t})$.

The following extension, see Section 5.1.3, of the true user ideal pdf is considered

$$\begin{aligned}
{}^{\lfloor U}f(d(\mathring{t})) &\equiv \prod_{t\in t^*} {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},d_o(t-1))\, {}^{\lfloor I}f(\Delta_{p+;t}|u_{o;t},d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1)) \\
{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},d_o(t-1)) &\equiv \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\, {}^{\lfloor U}r_i\right), \text{ note that } \Delta_{io;t} = \Delta_{i;t} \text{ for } i \le \mathring{\Delta}_o, \qquad (9.45) \\
{}^{\lfloor U}f(u_{o;t}|d_o(t-1)) &\equiv \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\, {}^{\lfloor U}r_i\right).
\end{aligned}$$

The upper bound on the KL distance, serving as the loss whose expectation is optimized, reads

$$\begin{aligned}
Z(d(\mathring{t})) &\equiv \sum_{t\in t^*} \int {}^{\lfloor I}f(u_{o;t}|d(t-1)) \left[\ln\left(\frac{{}^{\lfloor I}f(u_{o;t}|d(t-1))}{{}^{\lfloor U}f(u_{o;t}|d(t-1))} \right) + \omega(u_{o;t},d(t-1)) \right] du_{o;t} \\
\omega(u_{o;t},d(t-1)) &\equiv \sum_{c\in c^*} f(c|u_{o;t},d(t-1))\omega(c,u_{o;t},d(t-1)) \\
\omega(c,u_{o;t},d(t-1)) &\equiv \ln\left(\frac{f(c|u_{o;t},d(t-1))}{\alpha_c} \right) + \qquad (9.46) \\
&+ \int f(\Delta_t|u_{o;t},d(t-1),c) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t},u_{o;t},d(t-1),c)}{{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},d_o(t-1))} \right) d\Delta_t.
\end{aligned}$$

Its optimization calls for the evaluation of the weights

$$f(c|u_{o;t},d(t-1)) \equiv \frac{\alpha_c f(u_{o;t}|d(t-1),c)}{\sum_{c\in c^*} \alpha_c f(u_{o;t}|d(t-1),c)}. \tag{9.47}$$

For the considered normal mixture, they have to be approximated. It seems to be reasonable to approximate them by α_c within the design horizon and to use the values (9.47) in the final step only. The following proposition applies Proposition 7.2.6 in conjunction with this idea.

Proposition 9.2.2 (Industrial design minimizing the bound (7.22) & $f(c|u_{o;t}d(t-1)) \approx \alpha_c$) *Let the joint pdf*

$${}^{\lfloor I}f(\Delta(\mathring{t}),u_o(\mathring{t})) \equiv \prod_{t\in t^*} \frac{\sum_{c\in c^*} \alpha_c f(\Delta_t|u_{o;t},d(t-1),c) f(u_{o;t}|d(t-1),c)}{\sum_{c\in c^*} \alpha_c f(u_{o;t}|d(t-1),c)}\, {}^{\lfloor I}f(u_{o;t}|d(t-1)),$$

with components (9.44), be determined by the optional industrial advising strategy described by pdfs $\left\{ {}^{\lfloor I}f(u_{o;t}|d(t-1)) \right\}_{t\in t^*}$.

Let us search for the strategy approximately minimizing the expected value of the loss $Z(d(\mathring{t}))$ *(9.46). It is the upper bound on the KL distance* $\mathcal{D}\left({}^{\lfloor I}f || {}^{\lfloor U}f \right)$ *to the user ideal pdf (9.45). Moreover, let us assume that* $f(c|u_{o;t}, d(t-1)) \approx \alpha_c$, *cf. (9.47), for* $t > 1$.

In description of the strategy, we use the definition (9.33)

$$\mathcal{K}' \equiv \begin{bmatrix} I_{\mathring{d}(\partial-1)} & 0 & 0 \\ 0 & 0_{1,\mathring{d}} & 1 \end{bmatrix}, \quad L_t = \begin{bmatrix} {}^{\lfloor \phi 0}L_t & 0 \\ {}^{\lfloor \phi 1}L_t & 1 \end{bmatrix}, \; D_t = \mathrm{diag}\left[{}^{\lfloor \phi 0}D_t, \, {}^{\lfloor 1}D_t \right], \; {}^{\lfloor 1}D_t \text{ is scalar.}$$

We also use the following split of matrices defining LDL' *decompositions of involved kernels*

$$L_{ic} \equiv \begin{bmatrix} 1 & 0 \\ {}^{\lfloor \Delta\psi}L_{ic} & {}^{\lfloor \psi}L_{ic} \end{bmatrix}, \; D_{ic} \equiv \mathrm{diag}\left[{}^{\lfloor \Delta}D_{ic}, \, {}^{\lfloor \psi}D_{ic} \right], \quad \text{where } {}^{\lfloor \Delta}D_{ic} \text{ is scalar and } \mathring{D}_{(i+1)c} = \mathring{D}_{ic} - 1.$$

Then, with the introduced notation, the specialization of the approximately optimal strategy described in Proposition 7.2.6 becomes

$$ {}^{\lfloor I}f(u_{o;t}|d(t-1)) = \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}} \left({}^{\lfloor I}\theta'_{i;t-1}\psi_{i;t}, \, {}^{\lfloor I}r_{i;t-1} \right). \tag{9.48}$$

The used coefficients ${}^{\lfloor I}\theta_{i;t-1}$ *and variance* ${}^{\lfloor I}r_{i;t-1}$ *are generated by the following algorithm.*

$$\text{Set } L_{\mathring{t}} = I_{\mathring{\phi}}, \; D_{\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}} \tag{9.49}$$

For $t = \mathring{t}, \ldots, 1$

$$L_{\mathring{\Delta}} = I_{\mathring{\psi}}, D_{\mathring{\Delta}} = 0_{\mathring{\psi},\mathring{\psi}}$$

For $c = 1, \ldots, \mathring{c}$

$$L_{0c} \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_t & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor \phi 1}L'_t & 0 & 1 \end{bmatrix}, \quad D_{0c} \equiv \mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor \phi 0}D_t \right], 0_{1,\mathring{d}}, \, {}^{\lfloor 1}D_t \right], \quad {}^{\lfloor 1}D_t \text{ is scalar.}$$

For $i = 1, \ldots, \mathring{\Delta}$

$$\begin{aligned} L_{ic}D_{ic}L'_{ic} = {} & {}^{\lfloor \psi}L_{(i-1)c} \, {}^{\lfloor \psi}D_{(i-1)c} \, {}^{\lfloor \psi}L'_{(i-1)c} + \\ & + \left(\theta_{ic} + {}^{\lfloor \Delta\psi}L_{(i-1)c} \right) {}^{\lfloor \Delta}D_{(i-1)c} \left(\theta_{ic} + {}^{\lfloor \Delta\psi}L_{(i-1)c} \right)' + \\ & + \chi\left(i \le \mathring{\Delta}_o \right) \left(\theta_{ic} - {}^{\lfloor U}\theta_i \right) {}^{\lfloor U}r_i^{-1} \left(\theta_{ic} - {}^{\lfloor U}\theta_i \right)'. \end{aligned}$$

end of the cycle over i

$$L_{\mathring{\Delta}}D_{\mathring{\Delta}}L'_{\mathring{\Delta}} = L_{\mathring{\Delta}}D_{\mathring{\Delta}}L'_{\mathring{\Delta}} + \alpha_c L_{\mathring{\Delta}c}D_{\mathring{\Delta}c}L'_{\mathring{\Delta}c}$$

end of the cycle over c

For $i = \mathring{\Delta} + 1, \ldots, \mathring{d}$

$$\tilde{L}_i\tilde{D}_i\tilde{L}'_i = L_{i-1}D_{i-1}L'_{i-1} + \left[-1, \, {}^{\lfloor U}\theta'_i \right]' \, {}^{\lfloor U}r_i^{-1} \left[-1, \, {}^{\lfloor U}\theta'_i \right]$$

$$\tilde{L}_i \equiv \begin{bmatrix} 1 & 0 \\ -\,{}^{\lfloor I}\theta_{i;t-1} & L_i \end{bmatrix}, \quad \tilde{D}_i \equiv \mathrm{diag}\left[{}^{\lfloor I}r_{i;t-1}^{-1}, \mathrm{diag}[D_i] \right], \quad {}^{\lfloor I}r_{i;t-1} \text{ is scalar.}$$

end of the cycle over i

$$L_{t-1} \equiv L_{\mathring{d}}, \; D_{t-1} \equiv D_{\mathring{d}}.$$

end of the cycle over t

The ideal pdf presented to the user is ${}^{\lfloor I}f(d_{o;t}|d(t-1)) \equiv$

$$\equiv \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{(i-\mathring{\Delta})o;t}} \left({}^{\lfloor I}\theta'_{ic;t-1}\psi_{i;t}; \, {}^{\lfloor I}r_{ic;t-1} \right) \sum_{c\in c^*} f(c|u_{o;t}, d(t-1)) \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}(\theta'_{ic}\psi_{i;t}; r_{ic}) \quad \text{with}$$

$$f(c|u_{o;t}, d(t-1)) \propto \alpha_c \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}} (\theta'_{ic}\psi_{i;t}, r_{ic}) . \tag{9.50}$$

Proof: The proof is based on Proposition 9.2.2. Let us assume that the Bellman function has the form $-\ln(\gamma(d(t))) = 0.5\phi'_t L_t D_t L'_t \phi_t$ with data independent kernel $L_t D_t L'_t$. It is true for $t = \mathring{t}$.

Specialization of (7.39) should confirm this assumption and provide the corresponding algorithm.

We assume that $f(c|u_{o;t}, d(t-1)) \approx \alpha_c$ so that the first term in the definition (7.39) of $\omega(c, u_{o;t}, d(t-1))$ is zero and

$$\begin{aligned} 2\omega_\gamma(u_{o;t}, d(t-1)) &\equiv \sum_{c\in c^*} \alpha_c \int f(\Delta_t|u_{o;t}, d(t-1), c) \ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c)}{\gamma(d(t))\, {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))}\right) d\Delta_t = \\ &= \sum_{c\in c^*} \alpha_c \left[k_{\mathring{\Delta}c} + \psi'_{\mathring{\Delta};t} L_{\mathring{\Delta}c} D_{\mathring{\Delta}c} L'_{\mathring{\Delta}c} \psi_{\mathring{\Delta};t}\right] \equiv k_{\mathring{\Delta}} + \psi'_{\mathring{\Delta};t} L_{\mathring{\Delta}} D_{\mathring{\Delta}} L'_{\mathring{\Delta}} \psi_{\mathring{\Delta};t}. \end{aligned}$$

The individual lifts $k_{\mathring{\Delta}_c}$ and kernels $L_{\mathring{\Delta}_c}, D_{\mathring{\Delta}_c}$ in the above sum are evaluated using Proposition 9.1.7 that provides the values of the weighted conditional KL distance. The respective recursions start from the common initial condition $k_{0c} = k_\gamma = 0$, $L_{0c}D_{0c}L'_{0c} = \mathcal{K}L_tD_tL'_t\mathcal{K}'$. Results are summed into a single shifted quadratic form. The resulting strategy is proportional to the product ${}^{\lfloor U}f(u_{o;t}|d(t-1))\exp[-0.5\omega(u_{o;t}, \phi_{t-1})]$ Thus, the quadratic form is increased by the quadratic form in the exponent of the normal user ideal pdf on recognizable actions. The integration over $u_{o;t}$ can be performed recursively according to Proposition 9.1.8. It shows that $\gamma(d(t-1))$ preserves the assumed form as the average lift cancels in normalization and can be fixed at zero. Consequently, individual lifts need not be evaluated at all. It completes the full design step. Before summarizing results we recall that

$$\mathcal{K}' \equiv \begin{bmatrix} I_{\mathring{d}(\partial-1)} & 0 & 0 \\ 0 & 0_{1,\mathring{d}} & 1 \end{bmatrix}, \ L_t = \begin{bmatrix} {}^{\lfloor\phi 0}L_t & 0 \\ {}^{\lfloor\phi 1}L_t & 1 \end{bmatrix}, \ D_t = \mathrm{diag}\left[{}^{\lfloor\phi 0}D_t,\ {}^{\lfloor 1}D_t\right], \ {}^{\lfloor 1}D_t \text{ is scalar}$$

$$L_{ic} \equiv \begin{bmatrix} 1 & 0 \\ {}^{\lfloor\Delta\psi}L_{ic} & {}^{\lfloor\psi}L_{ic} \end{bmatrix}, \ D_{ic} \equiv \mathrm{diag}\left[{}^{\lfloor\Delta}D_{ic},\ {}^{\lfloor\psi}D_{ic}\right], \ \text{where } {}^{\lfloor\Delta}D_{ic} \text{ is scalar and } \mathring{D}_{(i+1)c} = \mathring{D}_{ic} - 1.$$

The computation can be organized as described in (9.49). □

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm that adds just the estimation "environment" to the algorithm described.

Algorithm 9.2.4 (Industrial design minimizing (7.22) with $f(c|u_{o;t}, d(t-1)) \approx \alpha_c$)
Initial (off line) mode

- *Estimate the mixture model of the o-system with the state ϕ_t, Chapter 8.*
- *Specify the true user ideal pdf (9.45) on the response of the o-system.*
- *Select the length of the receding horizon $T \geq 1$.*

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update estimates of the model parameters, Section 8.5, if you deal with the adaptive advisory system.*
3. *Initialize the iterative mode by setting $\tau = t + T$ and $L_\tau D_\tau L_\tau = 0$. Omit the initialization of L, D if $t > 0$ and the IST strategy is used.*

   Iterative mode

   - *Apply the algorithm given in Proposition 9.2.2 while replacing t by τ and stopping at $\tau = t+1$.*
4. *Present to the user projections of the ideal pdf (9.50) with pdfs $f(\Delta_{o;t}|u_{o;t+1}, \phi_t, c)$, $f(u_{o;t+1}|\phi_t, c)$ derived from the c-th learn mixture component.*

5. *Go to the beginning of* Sequential mode.

Remark(s) 9.2.3

A grouped version of the industrial design is implemented, see Proposition 7.2.4 and Remark 5.

Problem 9.2.1 (Improvements of industrial design) *The specific normal form of the adopted models could and should be used for constructing improved approximations both of the upper bound on the KL distance and on $f(c|u_{o;t}, d(t-1))$. Preliminary inspections have shown that differing influence of the adopted strategy on different components could be respected in a better way.*

At the end of this section, we attach a version of industrial design with a quadratic criterion. It has a tight connection with the multiple-model direction studied currently in control theory [90]. As such, it is of an independent interest. This makes us to present it. The optimization is made via dynamic programming written for a quadratic additive loss, Proposition 2.4.2.

Proposition 9.2.3 (Dynamic programming for a quadratic loss & known normal mixtures) *Let us consider the data-driven design and search for the optimal admissible strategy $\left\{d^*(t-1) \to u^*_{o;t}\right\}_{t\in t^*}$ acting on a non-decreasing sequence of experience $d(t)$. Let the system be described by the known normal mixture.*

Then, the optimal strategy minimizing expected value of the quadratic loss

$$\sum_{t\in t^*} \left(\phi_t' \, {}^{\lfloor\phi}L \, {}^{\lfloor\phi}D \, {}^{\lfloor\phi}L'\phi_t + u'_{o;t} \, {}^{\lfloor u}L \, {}^{\lfloor u}D \, {}^{\lfloor u}L' u_{o;t}\right), \tag{9.51}$$

determined by the constant penalization kernels ${}^{\lfloor\phi}L \, {}^{\lfloor\phi}D \, {}^{\lfloor\phi}L'$ and ${}^{\lfloor u}L \, {}^{\lfloor u}D \, {}^{\lfloor u}L'$ decomposed into the LDL' form, is generated by the following algorithm.

Algorithm 9.2.5 (Linear quadratic design for mixture models)
Initial mode

- *Construct the matrices A_c, $c \in c^*$ according to (9.36).*
- *Choose penalization kernels ${}^{\lfloor\phi}L \, {}^{\lfloor\phi}D \, {}^{\lfloor\phi}L'$ and ${}^{\lfloor u}L \, {}^{\lfloor u}D \, {}^{\lfloor u}L'$.*
- *Set $L = I_{\mathring{\phi}}$, $D = 0_{\mathring{\phi},\mathring{\phi}}$.*

Iterative mode

1. *Transform D, L to $\bar{D}$, $\bar{L}$ of the same type so that*

$$\bar{L}\bar{D}\bar{L}' = LDL' + {}^{\lfloor\phi}L \, {}^{\lfloor\phi}D \, {}^{\lfloor\phi}L'. \tag{9.52}$$

2. *Compute $\bar{A}_c = \bar{L}A_c$, $c \in c^*$, while exploiting the special form of the shifting matrix Λ, see (9.36).*
3. *Transform $\bar{A}_c$, $c \in c^*$, and $\bar{D}$, to $\tilde{L}, \tilde{D}$ so that the following quadratic form is preserved*

$$\sum_{c\in c^*} \bar{A}_c'\bar{D}\bar{A}_c + \left[{}^{\lfloor u}L', 0\right]' \, {}^{\lfloor u}D \left[{}^{\lfloor u}L', 0\right] = \tilde{L}\tilde{D}\tilde{L}'. \tag{9.53}$$

4. *Split $\tilde{L}$, $\tilde{D}$*

$$\tilde{L} \equiv \begin{bmatrix} {}^{\lfloor u}\tilde{L} & 0 \\ {}^{\lfloor u\phi}\tilde{L} & {}^{\lfloor\phi}\tilde{L} \end{bmatrix}, \ \tilde{D} = \mathrm{diag}\left[{}^{\lfloor u}\tilde{D}, \, {}^{\lfloor\phi}\tilde{D}\right], \quad \textit{where } {}^{\lfloor u}\tilde{L}, \ {}^{\lfloor u}\tilde{D} \textit{ are of } (\mathring{u},\mathring{u})\textit{-type.} \tag{9.54}$$

5. *Set $L = {}^{\lfloor\phi}\tilde{L}$ and $D = {}^{\lfloor\phi}\tilde{D}$.*
6. *Go to the beginning of* Iterative mode *until convergence is observed.*

Sequential (on line) mode, *running for t=1,2,...,*
Generate the optimal recognizable actions by the linear feedback

$$u_{o;t} = -\left({}^{\lfloor u}\tilde{L}'\right)^{-1} {}^{\lfloor u\phi}\tilde{L}'\phi_{t-1}. \tag{9.55}$$

Proof: For the considered finite horizon $\mathring{t}$ and additive loss, the optimal strategy can be constructed value-wise. The optimal recognizable inputs are minimizing arguments in the functional Bellman equation (2.23)

$$\mathcal{V}(d(t-1)) = \min_{u_{o;t}\in u^*_{o;t}} \mathcal{E}\left[\phi_t' \,{}^{\lfloor\phi}L\,{}^{\lfloor\phi}D\,{}^{\lfloor\phi}L'\phi_t + u_{o;t}'\,{}^{\lfloor u}L\,{}^{\lfloor u}D\,{}^{\lfloor u}L'u_{o;t} + \mathcal{V}(d(t))|u_{o;t}, d(t-1)\right],\ t\in t^*,$$

starting with $\mathcal{V}(d(\mathring{t}))=0$. In the considered case of the normal mixture with the common state in the phase form, the Bellman function depends on the state only. It is conjectured to be of the lifted quadratic form

$$\mathcal{V}(d(t)) = \mathcal{V}(\phi_t) = k_t + \phi_t' L_t D_t L_t' \phi_t \tag{9.56}$$

with the non-negative scalar lift k_t and the decomposed kernel $L_tD_tL_t'$ independent of data.

For $\mathring{t}$, the conjecture is valid with $L_{\mathring{t}} = I_{\mathring{\phi}}$, $D_{\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$ and $k_{\mathring{t}} = 0$. According to the inductive assumption for a generic t, the right-hand of the Bellman equation is

$$\begin{aligned}
&\min_{u_{o;t}} \mathcal{E}\left[\phi_t'\left({}^{\lfloor\phi}L\,{}^{\lfloor\phi}D\,{}^{\lfloor\phi}L' + L_tD_tL_t'\right)\phi_t + u_{o;t}'\,{}^{\lfloor u}L\,{}^{\lfloor u}D\,{}^{\lfloor u}L'u_{o;t}|\psi_t\right] + k_t \underbrace{=}_{(9.52)} \qquad (9.57)\\
=\ & \min_{u_{o;t}} \mathcal{E}\left[\phi_t'\,{}^{\lfloor\phi}\bar{L}\,{}^{\lfloor\phi}\bar{D}\,{}^{\lfloor\phi}\bar{L}'\phi_t + u_{o;t}'\,{}^{\lfloor u}L\,{}^{\lfloor u}D\,{}^{\lfloor u}L'u_{o;t}|\psi_t\right] + k_t \underbrace{=}_{(9.36)}\\
=\ & \min_{u_{o;t}} \psi_t'\left(\sum_{c\in c^*} A_c'\,{}^{\lfloor\phi}\bar{L}\,{}^{\lfloor\phi}\bar{D}\,{}^{\lfloor\phi}\bar{L}'A_c\right)\psi_t + u_{o;t}'\,{}^{\lfloor u}L\,{}^{\lfloor u}D\,{}^{\lfloor u}L'u_{o;t} +\\
+\ & \underbrace{\mathrm{tr}\left[[I_{\mathring{\Delta}},0]\bar{L}\bar{D}\bar{L}'[I_{\mathring{\Delta}},0]'\sum_{c\in c^*}\alpha_c r_c\right] + k_t}_{k_{t-1}} \underbrace{=}_{(9.53)}\\
=\ & \min_{u_{o;t}} \psi_t'\tilde{L}\tilde{D}\tilde{L}'\psi_t + k_{t-1} \underbrace{=}_{(9.54)} \phi_{t-1}'\,{}^{\lfloor\phi}\tilde{L}\,{}^{\lfloor\phi}\tilde{D}\,{}^{\lfloor\phi}\tilde{L}'\phi_{t-1} + k_{t-1} \text{ with the minimizing argument}\\
u_{o;t}' =\ & -\phi_{t-1}'\,{}^{\lfloor u\phi}\tilde{L}\,{}^{\lfloor u}\tilde{L}^{-1}.
\end{aligned}$$

These evaluations prove both the claimed form and provide the algorithm giving the optimal feedback. □

Remark(s) 9.2.4

1. *The algorithm is presented directly in the way suitable for numerically efficient, factorized solution based on Algorithm* dydr *8.1.3.*
2. *For simplicity, the regulation problem is considered. An extension to the combined regulation and tracking problems is straightforward and standard one.*
3. *The optimization is made for the known mixture model. As usual in linear-quadratic design problems, the normality assumption serves us for learning. For the design, just the knowledge of the second moment (9.36) is needed.*
4. *The key message is that the system matrix corresponding to the mixture is not weighted mean of system matrices of individual components. The combination has to be done in the mean square sense as reflected in Algorithm 9.2.5.*
5. *Knowledge on linear-quadratic control theory can be simply exploited. For instance, if there is a controller that makes the expected loss – divided by $\mathring{t}$ – finite for $\mathring{t}\to\infty$, the designed strategy makes this loss finite, too. Also, a sufficient rank of penalizing matrices and existence of a stabilizing controller imply that the designed one is stabilizing, etc.*

6. *The optimization is equivalent to a solution of the Riccati equation [26]. Thus, the presented evaluations can viewed as its extension to mixture models.*

7. *The IST strategy, Section 4.2.1, should be used in the adaptive context.*

8. *Considering the derived algorithm with empty $u_{o;t}$, we evaluate second non-central moment assigned to the whole mixture. This evaluation can serve us for practical test of the* stability of the mixture *as a whole.*

Problem 9.2.2 (Extension of linear-quadratic optimization art to mixtures) *It is obvious that the well developed linear-quadratic optimization art can be extended to mixture models. It is worth to do it in detail.*

9.2.3 Simultaneous academic and industrial design

Whenever possible, the simultaneous academic and industrial design provides the best problem formulation and solution. The considered actions of the academic part of the simultaneous p-system $c_t \in c^* \equiv \{1, \ldots, \mathring{c}\}$ are to be generated by a causal strategy $d^*(t-1) \rightarrow c^*$. The industrial part generates the recommended recognizable actions $d^*(t-1) \rightarrow u^*_{o;t}$. The strategy determines the potential ideal pdfs, among which the best one is searched for,

$$
\begin{aligned}
{}^{\lfloor I}f(d_t, c_t|d(t-1)) &= f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t)\times \qquad (9.58)\\
&\times\ f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1)) \text{ using the user ideal pdf}\\
{}^{\lfloor U}f(d_t, c_t|d(t-1)) &= {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\times\\
&\times\ f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)) \propto\\
&\propto \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\, {}^{\lfloor U}r_i\right) \prod_{i=\mathring{\Delta}_o+1}^{\mathring{\Delta}} \mathcal{N}_{\Delta_{i;t}}\left(\theta'_{ic_t}\psi_{ic_t;t}, r_{ic_t}\right) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\, {}^{\lfloor U}r_i\right)\times\\
&\times\ {}^{\lfloor U}f(c_t)\exp\left[-0.5\left({}^{\lfloor U}k_{c_t;t-1} + \psi_t'\, {}^{\lfloor U}L_{c_t;t-1}\, {}^{\lfloor U}D_{c_t;t-1}\, {}^{\lfloor U}L'_{c_t;t-1}\psi_t\right)\right], \text{ where}
\end{aligned}
$$

$f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t) \equiv \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{ic_t;t}}\left(\theta'_{ic_t}\psi_{ic_t;t}, r_{ic_t}\right)$ is the pdf derived from the c_t-th learnt component describing the o-innovations;

$f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t) \equiv \prod_{i=\mathring{\Delta}_o+1}^{\mathring{\Delta}} \mathcal{N}_{\Delta_{i;t}}\left(\theta'_{ic_t}\psi_{ic_t;t}, r_{ic_t}\right)$ is the pdf derived from the c_t-th learnt component describing the surplus innovations of the p-system;

$\left\{{}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) \equiv {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1))\right\}_{t\in t^*}$ is the optimized simultaneous strategy;

${}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1)) = \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\, {}^{\lfloor U}r_i\right)$ is the true user ideal pdf on the o-innovations, see Section 5.1.3;

${}^{\lfloor U}f(u_{o;t}|d_o(t-1)) = \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\, {}^{\lfloor U}r_i\right)$ is the true user ideal pdf on the recognizable actions $u_{o;t}$, see Section 5.1.3;

${}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)) \propto {}^{\lfloor U}f(c_t)\exp\left[-0.5\left({}^{\lfloor U}k_{c_t;t-1} + \psi_t'\, {}^{\lfloor U}L_{c_t;t-1}\, {}^{\lfloor U}D_{c_{t-1};t}\, {}^{\lfloor U}L'_{c_t;t-1}\psi_t\right)\right]$ is the pf representing the optional knob of the p-system that can respect special requirements like stability of advices. It is determined by the time invariant pf ${}^{\lfloor U}f(c_t)$ and by the lifted quadratic forms in exponents with kernels determined by the lower triangular matrices ${}^{\lfloor U}L_{c_t;t-1}$ with unit diagonal and by the diagonal matrices ${}^{\lfloor U}D_{c_t;t-1}$.

Note that the quadratic forms in regression vector ψ_t are considered, i.e. the desirable c_t can be made dependent on $u_{o;t}$. The function ${}^{\lfloor U}\omega(c_t, \psi_t) \equiv {}^{\lfloor U}k_{c_t;t-1} + \psi_t'\, {}^{\lfloor U}L_{c_t;t-1}\, {}^{\lfloor U}D_{c_t;t-1}\, {}^{\lfloor U}L'_{c_t;t-1}\psi_t$ can be interpreted as a version of the user KL distance.

Proposition 9.2.4 (Simultaneous advisory system minimizing the KL distance) *Let us consider the simultaneous academic and industrial design for the o-system described by the normal mixture with the state ϕ in the phase form. The data record d_t contains both innovations $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t})$ = (innovations in d_o^*, innovations in d_{p+}^*) and the <u>unrestricted</u> recognizable actions $u_{o;t}$. The o-data $d_{o;t}$ consist of $(\Delta_{o;t}, u_{o;t})$ = (o-innovations, recognizable actions). The data record is ordered as follows $d = (\Delta_o, \Delta_{p+}, u_o)$.*

The assumed influence of advices and the user ideal pdf are described by (9.58). The parameters $\Theta_{ic} = [\theta_{ic}, r_{ic}]$ of the mixture model as well those determining the true user ideal pdf

$$ {}^{\lfloor U}\Theta_i \equiv \left[{}^{\lfloor U}\theta_i,\ {}^{\lfloor U}r_i \right],\ i \in i^* \equiv \left\{ 1, \ldots, \mathring{\Delta}_o \right\} \cup \left\{ \mathring{\Delta} + 1, \ldots, \mathring{d} \right\} $$

are known and fixed. The regression coefficients are complemented by zeros so that, for each index i in the set i^*, the corresponding factors in the user ideal pdf and mixture model have the common regression vectors $\psi_{i;t} \equiv [d_{i+1;t}, \psi'_{i+1;t}]' = [d'_{(i+1)\_\mathring{d};t}, \phi'_{t-1}]'$, $i < \mathring{d}$, $\psi_{\mathring{d};t} \equiv \phi_{t-1}$.

Let us search for the advising strategy $\left\{ {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) \right\}_{t \in t^*}$ selecting both the recommended pointers c_t and the recognizable actions $u_{o;t}$ that, at least approximately, minimizes the KL distance of

$$ \begin{aligned} {}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t})) &= \prod_{t \in t^*} f(d_t, c_t|d(t-1)) \quad \textit{to the user ideal pdf} \\ {}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) &= \prod_{t \in t^*} {}^{\lfloor U}f(d_t, c_t|d(t-1)) \ \textit{with its factors given by (9.58).} \end{aligned} $$

The definition of the KL distance, chain rule and marginalization imply that the minimized KL distance can be interpreted as the expected value of the loss function,

$$ \begin{aligned} &\sum_{t \in t^*, c_t \in c^*} \int f(\Delta_t|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1)) \times \qquad (9.59) \\ \times \quad &\ln \left(\frac{f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1))}{{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1))\, {}^{\lfloor U}f(c_t|u_{o;t}, d_o(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1))} \right) dd_t. \end{aligned} $$

Let us consider the following strategy

$$ \begin{aligned} {}^{\lfloor I}f(c_t, u_{o;t}|\phi_{t-1}) &\propto {}^{\lfloor U}f(c_t) \exp\left[-0.5\omega_\gamma(c_t, \phi_{t-1})\right] {}^{\lfloor I}f(u_{o;t}|\phi_{t-1}, c_t), \quad \textit{where} \qquad (9.60) \\ \omega_\gamma(c_t, \phi_{t-1}) &\equiv k_{c_t;t-1} + \phi'_{t-1} L_{c_t;t-1} D_{c_t;t-1} L'_{c_t;t-1} \phi_{t-1} \\ {}^{\lfloor I}f(u_{o;t}|\phi_{t-1}, c_t) &\equiv \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta})};t} \left({}^{\lfloor I}\theta'_{ic_t;t-1}\psi_{i;t},\ {}^{\lfloor I}r_{ic_t;t-1} \right). \end{aligned} $$

The involved lifts $k_{c_t;t-1}$ and kernels $L_{c_t;t-1} D_{c_t;t-1} L'_{c_t;t-1}$ determining $\omega_\gamma(c_t, \phi_{t-1})$ as well as the parameters ${}^{\lfloor I}\Theta_{ic_t;t-1} \equiv \left[{}^{\lfloor I}\theta'_{ic_t;t-1},\ {}^{\lfloor I}r_{ic_t;t-1} \right]$, defining pdfs ${}^{\lfloor I}f(u_{o;t}|\phi_{t-1}, c_t)$, are generated recursively. In the description of recursions, the following split of LDL' kernels is used

$$ L \equiv \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L & {}^{\lfloor \psi}L \end{bmatrix}, \quad D \equiv \operatorname{diag}\left[{}^{\lfloor d}D,\ {}^{\lfloor \psi}D \right], \quad \textit{where } {}^{\lfloor d}D \textit{ is scalar.} $$

An average quadratic form *$\phi'_{t-1} L_{a;t-1} D_{a;t-1} L'_{a;t-1} \phi_{t-1}$ is used in the specification of the initial conditions of recursions defining the strategy. Its following split is used*

$$ L_{a;t} \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_{a;t} & 0 \\ {}^{\lfloor \phi 1}L'_{a;t} & 1 \end{bmatrix},\ D_{a;t} = \operatorname{diag}\left[{}^{\lfloor \phi 0}D_{a;t},\ {}^{\lfloor 1}D_{a;t} \right],\ {}^{\lfloor 1}D_{a;t} \textit{ is scalar.} $$

The recursions are described by the following formulas.

$$ \textit{Set } L_{a;\mathring{t}} = I_{\mathring{\phi}} = \mathring{\phi}\textit{-unit matrix},\ D_{a;\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}} \equiv (\mathring{\phi}, \mathring{\phi})\textit{-zero matrix.} $$

For $t = \mathring{t}, \ldots, 1$

$q = 0$

For $c = 1, \ldots, \mathring{c}$

$$L_{0c} \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_{a;t} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor \phi 1}L'_{a;t} & 0 & 1 \end{bmatrix}, \; D_{0c} = \operatorname{diag}\left[\operatorname{diag}\left[{}^{\lfloor \phi 0}D_{a;t}\right], 0_{1,\mathring{d}}, {}^{\lfloor 1}D_{a;t}\right],$$

$k_{0c} \equiv -\mathring{d}_o$

For $i = 1, \ldots, \mathring{\Delta}$

$$L_{ic}D_{ic}L'_{ic} = {}^{\lfloor \psi}L_{(i-1)c}\,{}^{\lfloor \psi}D_{(i-1)c}\,{}^{\lfloor \psi}L'_{(i-1)c} +$$

$$+\left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right) {}^{\lfloor d}D_{(i-1)c} \left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right)' +$$

$$+\chi\left(i \le \mathring{\Delta}_o\right) \left(\theta_{ic} - {}^{\lfloor U}\theta_i\right) {}^{\lfloor U}r_i^{-1} \left(\theta_{ic} - {}^{\lfloor U}\theta_i\right)',$$

$$k_{ic} \equiv k_{(i-1)c} + {}^{\lfloor d}D_{(i-1)c}r_{ic} + \chi\left(i \le \mathring{\Delta}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right) + \frac{r_{ic}}{{}^{\lfloor U}r_i}\right], \; \mathring{D}_{ic} = \mathring{D}_{(i-1)c} - 1$$

end *of the cycle over* i

$$L_{\mathring{\Delta}c}D_{\mathring{\Delta}c}L'_{\mathring{\Delta}c} \equiv L_{\mathring{\Delta}c}D_{\mathring{\Delta}c}L'_{\mathring{\Delta}c} + {}^{\lfloor U}L_{c;t-1}\,{}^{\lfloor U}D_{c;t-1}\,{}^{\lfloor U}L'_{c;t-1}$$

$k_{\mathring{\Delta}c} \equiv k_{\mathring{\Delta}c} + {}^{\lfloor U}k_{c;t-1}$

end *of the cycle over* c

$L_{a;t-1} = I_{\mathring{\phi}}, \; D_{a;t-1} = 0_{\mathring{\phi},\mathring{\phi}}$

For $c = 1, \ldots, \mathring{c}$

For $i = \mathring{\Delta} + 1, \ldots, \mathring{d}$

$$\tilde{L}_{ic}\tilde{D}_{ic}\tilde{L}'_{ic} = L_{(i-1)c}D_{(i-1)c}L'_{(i-1)c} + \left[-1, {}^{\lfloor U}\theta'_i\right]' {}^{\lfloor U}r_i^{-1}\left[-1, {}^{\lfloor U}\theta'_i\right]$$

$$\tilde{L}_{ic} = \begin{bmatrix} 1 & 0 \\ -{}^{\lfloor I}\theta_{ic;t-1} & L_{ic} \end{bmatrix}$$

$$\tilde{D}_{ic} = \operatorname{diag}\left[{}^{\lfloor I}r^{-1}_{ic;t-1}, D_{ic}\right], \; {}^{\lfloor I}r_{ic;t-1} \textit{ is scalar},$$

$$k_{ic} = k_{(i-1)c} + \ln\left({}^{\lfloor U}r_i\,{}^{\lfloor I}r^{-1}_{ic;t-1}\right)$$

end *of the cycle over* i

$L_{c;t-1} \equiv L_{\mathring{d}c}, \; D_{c;t-1} \equiv D_{\mathring{d}c}, \; k_{c;t-1} \equiv k_{\mathring{d}c}$

$\beta_c \equiv {}^{\lfloor U}f(c)\exp(-0.5k_{\mathring{d}c})$

$q = q + \beta_c$

end *of the cycle over* c

For $c = 1, \ldots, \mathring{c}$

$$\beta_c = \frac{\beta_c}{q}$$

$$L_{a;t-1}D_{a;t-1}L'_{a;t-1} = L_{a;t-1}D_{a;t-1}L'_{a;t-1} + \beta_c L_{c;t-1}D_{c;t-1}L'_{c;t-1}.$$

end *of the cycle over* c

end *of the cycle over* t

The updating of the LDL′ (!) decomposition can be done by Algorithm 8.1.2.

Then, this strategy minimizes the KL distance loss for the horizon equal one. For other horizons, it minimizes the upper on the expected loss as it replaces the correct Bellman function, cf. (9.60),

$$-\ln\left\{\sum_{c_t \in c^*} {}^{\lfloor U}f(c_t)\exp[-0.5\omega_\gamma(c_t, \phi_{t-1})]\right\} \textit{ by the larger value} \tag{9.61}$$

$$0.5\phi'_{t-1}\left(\sum_{c_t\in c^*}\beta_{c_t}L_{c_t;t-1}D_{c_c;t-1}L'_{c_t;t-1}\right)\phi_{t-1}.$$

Proof: Let us assume that the Bellman function is $-\ln(\gamma(\phi_t))$, with

$$\begin{aligned}
\gamma(\phi_t) &= \exp\left[-0.5\sum_{c_{t+1}\in c^*}\beta_{c_{t+1}}\underline{\omega}_\gamma(c_{t+1},\phi_t)\right]\equiv\exp\left[-0.5\omega_\gamma(\phi_t)\right], \text{ where}\\
\beta_{c_{t+1}} &\propto {}^{\lfloor U}f(c_{t+1})\exp[-0.5k_{c_{t+1};t}]\\
\underline{\omega}_\gamma(c_{t+1},\phi_t) &= \phi'_tL_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}\phi_t, \text{ i.e. } \omega_\gamma(\phi_t)=\phi'_tL_{a;t}D_{a;t}L'_{a;t}\phi_t \text{ with}\\
L_{a;t}D_{a;t}L'_{a;t} &\equiv \sum_{c_{t+1}\in c^*}\beta_{c_{t+1}}L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}.
\end{aligned}$$

The kernels $L_{c_{t+1};t}D_{c_{t+1};t}L'_{c_{t+1};t}$ and the scalar lifts $k_{c_{t+1};t}$ are conjectured to be independent of data.

This conjecture is true for $t=\mathring{t}$ with $k_{c_{\mathring{t}+1};\mathring{t}}=0,\ L_{c_{\mathring{t}+1};\mathring{t}}=I_{\mathring{\phi}},\ D_{c_{\mathring{t}+1};\mathring{t}}=0_{\mathring{\phi},\mathring{\phi}}$. We prove that this form is preserved during the backward induction for $t<\mathring{t}$ if the approximation (9.61) is used. It gives us also the algorithmic solution of the inspected problem.

Under the inductive assumption, the weighted conditional KL distance is

$$2\tilde{\omega}_\gamma(c_t,\psi_t) \equiv 2\int f(\Delta_t|\psi_t,c_t)\ln\left[\frac{f(\Delta_{o;t}|\Delta_{p+;t},\psi_t,c_t)}{\gamma(\phi_t)\,{}^{\lfloor U}f(\Delta_{o;t}|\psi_t)}\right]d\Delta_t = k_{\mathring{\Delta}c_t}+\psi'_tL_{\mathring{\Delta}c_t}D_{\mathring{\Delta}c_t}L'_{\mathring{\Delta}c_t}\psi_t, \text{ where}$$

according to Proposition 9.1.7 the lifts and the kernels are found recursively for $i=1,\ldots,\mathring{\Delta}$

$$\begin{aligned}
L_{ic}D_{ic}L'_{ic} &= {}^{\lfloor\psi}L_{(i-1)c}\,{}^{\lfloor\psi}D_{(i-1)c}\,{}^{\lfloor\psi}L'_{(i-1)c}+\left(\theta_{ic}+{}^{\lfloor\Delta\psi}L_{(i-1)c}\right){}^{\lfloor\Delta}D_{(i-1)c}\left(\theta_{ic}+{}^{\lfloor\Delta\psi}L_{(i-1)c}\right)'+\\
&+\ \chi\left(i\le\mathring{\Delta}_o\right)\left(\theta_{ic}-{}^{\lfloor U}\theta_i\right){}^{\lfloor U}r_i^{-1}\left(\theta_{ic}-{}^{\lfloor U}\theta_i\right)'\\
k_{ic} &= k_{(i-1)c}+{}^{\lfloor\Delta}D_{(i-1)c}r_{ic}+\chi\left(i\le\mathring{\Delta}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right)+\frac{r_{ic}}{{}^{\lfloor U}r_i}\right]\\
&\ \text{starting from the initial conditions}\\
k_{0c} &= -\mathring{\Delta}_o,\quad L_{0c}D_{0c}L'_{0c}=\mathcal{K}L_{a;t}D_{a;t}L'_{a;t}\mathcal{K}'\\
\mathcal{K}' &\equiv \begin{bmatrix} I_{\mathring{d}(\partial-1)} & 0 & 0\\ 0 & 0_{1,\mathring{d}} & 1\end{bmatrix}\\
L_{ic} &\equiv \begin{bmatrix} 1 & 0\\ {}^{\lfloor\Delta\psi}L_{ic} & {}^{\lfloor\psi}L_{ic}\end{bmatrix},\ D_{ic}\equiv\operatorname{diag}\left[{}^{\lfloor\Delta}D_{ic},\,{}^{\lfloor\psi}D_{ic}\right], \text{ where}\\
&\ {}^{\lfloor\Delta}D_{ic} \text{ is scalar and } \mathring{D}_{(i+1)c}=\mathring{D}_{ic}-1.
\end{aligned}$$

The optimal ${}^{\lfloor I}f(c_t,u_{o;t}|d(t-1))$ is minimizer of

$$\begin{aligned}
&\sum_{c_t\in c^*}\int {}^{\lfloor I}f(c_t,u_{o;t}|d(t-1))\left[\omega_\gamma(c_t,\psi_t)+\ln\left(\frac{{}^{\lfloor I}f(c_t,u_{o;t}|d(t-1))}{{}^{\lfloor U}f(c_t|u_{o;t},d_o(t-1))\,{}^{\lfloor U}f(u_{o;t}|\phi_{t-1})}\right)\right]du_{o;t}, \text{ where}\\
&\omega_\gamma(c_t,\psi_t)\equiv\tilde{\omega}_\gamma(c_t,\psi_t)+{}^{\lfloor U}\omega(c_t,\psi_t).
\end{aligned}$$

This combination adds to the kernel $L_{\mathring{\Delta}c}D_{\mathring{\Delta}c}L'_{\mathring{\Delta}c}$ the kernel ${}^{\lfloor U}L_{c;t-1}\,{}^{\lfloor U}D_{c;t-1}\,{}^{\lfloor U}L'_{c;t-1}$ and the lift ${}^{\lfloor U}k_{c;t-1}$ to $k_{\mathring{\Delta}c}$. Then, the minimizer is

$$\begin{aligned}
{}^{\lfloor I}f(c_t,u_{o;t}|\phi_{t-1}) &= \frac{{}^{\lfloor U}f(c_t)\,{}^{\lfloor U}f(u_{o;t}|\phi_{t-1})\exp[-0.5\omega_\gamma(c_t,\psi_t)]}{\tilde{\gamma}(\phi_{t-1})}\\
\tilde{\gamma}(\phi_{t-1}) &\equiv \sum_{c_t\in c^*}{}^{\lfloor U}f(c_t)\int {}^{\lfloor U}f(u_{o;t}|\phi_{t-1})\exp[-0.5\omega_\gamma(c_t,\psi_t)]\,du_{o;t}.
\end{aligned}$$

We modify this expression modified so that the marginal pf ${}^{\lfloor I}f(c_t|\phi_{t-1})$ become visible. For it, we use the explicit form of the user ideal pdf on recognizable actions when expressing the product

$$\exp[-0.5\omega_\gamma(c_t,\psi_t)]\,{}^{\lfloor U}f(u_{o;t}|\phi_{t-1}) = \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \left(2\pi\,{}^{\lfloor U}r_i\right)^{-0.5} \times$$

$$\times\exp\left\{-\frac{1}{2}\left(k_{\mathring{\Delta}c_t} + \psi_t' L_{\mathring{\Delta}c_t} D_{\mathring{\Delta}c_t} L_{\mathring{\Delta}c_t}'\psi_t + \sum_{i=\mathring{\Delta}+1}^{\mathring{d}} [u_{o(i-\mathring{\Delta});t},\psi_{i;t}'] \frac{\left[-1,\,{}^{\lfloor U}\theta_i'\right]'\left[-1,\,{}^{\lfloor U}\theta_i'\right]}{{}^{\lfloor U}r_i}[u_{o(i-\mathring{\Delta});t},\psi_{i;t}']'\right)\right\}.$$

Completion of the squares in individual entries of $u_{o(i-\mathring{\Delta});t}$ defines sequence of kernels for $i=\mathring{\Delta}+1,\ldots,\mathring{d}$, $c\in c^*$,

$$\tilde{L}_{ic}\tilde{D}_{ic}\tilde{L}_{ic}' \equiv L_{(i-1)c}D_{(i-1)c}L_{(i-1)c}' + \frac{\left[-1,\,{}^{\lfloor U}\theta_i'\right]'\left[-1,\,{}^{\lfloor U}\theta_i'\right]}{{}^{\lfloor U}r_i}$$

$$\tilde{L}_{ic} = \begin{bmatrix} 1 & 0 \\ -\,{}^{\lfloor I}\theta_{ic;t-1} & L_{ic}\end{bmatrix},\ \tilde{D}_{ic}\equiv \operatorname{diag}\left[{}^{\lfloor I}r_{ic;t-1}^{-1}, \operatorname{diag}[D_{ic}]\right],\ {}^{\lfloor I}r_{ic;t-1}^{-1} \text{ is scalar.}$$

Thus, $${}^{\lfloor I}f(c_t,u_{o;t}|\phi_{t-1}) = \frac{{}^{\lfloor U}f(c_t)}{\tilde{\gamma}(\phi_{t-1})}\times$$

$$\times\exp\left\{-0.5\left[k_{c_t;t-1} + \phi_{t-1}' L_{c_t;t-1}D_{c_t;t-1}L_{c_t;t-1}'\phi_{t-1}\right]\right\}\prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor I}\theta_{ic;t-1}'\psi_{i;t},\ {}^{\lfloor I}r_{ic;t-1}\right)$$

and the integration over u_{oi}, performed while normalizing, increases lifts to

$$k_{c_t;t-1} \equiv k_{\mathring{\Delta}c_t} + \sum_{i=\mathring{\Delta}+1}^{\mathring{d}} \ln\left(\frac{{}^{\lfloor U}r_i}{{}^{\lfloor I}r_{ic;t-1}}\right) \text{ and } L_{c_t;t-1}\equiv L_{\mathring{d}c_t},\quad D_{c_t;t-1}\equiv D_{\mathring{d}c_t}.$$

$$\begin{aligned}
\tilde{\gamma}(\phi_{t-1}) &\equiv \sum_{c_t\in c^*} {}^{\lfloor U}f(c_t)\exp\left\{-0.5\left[k_{c_t;t-1}+\phi_{t-1}'L_{c_t;t-1}D_{c_t;t-1}L_{c_t;t-1}'\phi_{t-1}\right]\right\} = \\
&= q\sum_{c_t\in c^*}\beta_{c_t}\exp\left[-0.5\phi_{t-1}'L_{c_t;t-1}D_{c_t;t-1}L_{c_t;t-1}'\phi_{t-1}\right] = q\sum_{c_t\in c^*}\beta_{c_t}\exp\left[-0.5\underline{\omega}(c_t;\phi_{t-1})\right] \quad \text{with} \\
\beta_{c_t} &= \frac{{}^{\lfloor U}f(c_t)\exp\left[-0.5k_{c_t;t-1}\right]}{q},\quad q\equiv\sum_{c_t\in c^*}{}^{\lfloor U}f(c_t)\exp\left[-0.5k_{c_t;t-1}\right].
\end{aligned}$$

The obtained minimum $-\ln(\tilde{\gamma}(\phi_{t-1}))$ is approximated from above by replacing weighted arithmetic mean by weighted geometric mean. It reproduces

$$\begin{aligned}
\gamma(\phi_{t-1}) &= \exp\left[-0.5\phi_{t-1}'L_{a;t-1}D_{a;t-1}L_{a;t-1}'\phi_{t-1}\right] \quad \text{with} \\
L_{a;t-1}D_{a;t-1}L_{a;t-1}' &\equiv \sum_{c_t\in c^*}\beta_{c_t}L_{c_t;t-1}D_{c_t;t-1}L_{c_t;t-1}'.
\end{aligned}$$

□

The presented proposition, combined with the certainty-equivalent strategy, justifies the following algorithm for the design of the fixed simultaneous advisory system.

Algorithm 9.2.6 (Fixed simultaneous advising minimizing γ-bound on the KL distance)
Initial (off line) mode

- *Estimate normal mixture model of the o-system with the state ϕ_t, Chapter 8.*
- *Specify the true user ideal pdf on the response of the o-system*

$${}^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{\Delta}_o}\mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\ {}^{\lfloor U}r_i\right)\prod_{i=\mathring{\Delta}+1}^{\mathring{d}}\mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor U}\theta_i'\psi_{i;t},\ {}^{\lfloor U}r_i\right).$$

- *Specify* data and time invariant *part* ${}^{\lfloor U}f(c_t)$ *of the user ideal on the recommended pointers. It is zero on dangerous components.*
- *Specify* data and time invariant lifts ${}^{\lfloor U}k_{c_t}$ *and* data and time invariant kernels *of the user KL distance* ${}^{\lfloor U}L_{c_t}$, ${}^{\lfloor U}D_{c_t}$. *It completes definition of the user ideal pf* $f(c_t|d(t-1))$, *see (9.58). Notice that the quadratic form in the regression vector (not in state) is to be considered.*
- *Select the length of the design horizon* $\mathring{t} \geq 1$.
- *Initialize the iterative mode by setting* $L_{a;\mathring{t}} = I_{\mathring{\phi}}$, $D_{a;\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$.

Iterative (off line) mode

1. *Correct the conditional KL distances, for* $t = \mathring{t}, \ldots, 1$, *as given in (9.60).*
2. *Denote the final characteristics of the approximate optimal* steady-state strategy, *cf. Chapter 3,*

$$k_c \equiv k_{c;1},\ L_c \equiv L_{c;1},\ D_c \equiv D_{c;1},\ \ \theta_{ic} \equiv {}^{\lfloor I}\theta_{ic;1},\ r_{ic} \equiv {}^{\lfloor I}r_{ic;1},\ i = \mathring{\Delta}+1, \ldots, \mathring{d},\ c \in c^*.$$

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Evaluate the ideal pf on pointers*

$$ {}^{\lfloor I}f(c_{t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}) \exp\left[-0.5\left(k_{c_{t+1}} + \phi'_{t-1} L_{c_{t+1}} D_{c_{t+1}} L'_{c_{t+1}} \phi_{t-1}\right)\right].$$

3. *Present to the user selected projections of the ideal pdf*

$$ {}^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1}\in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) \prod_{i=1}^{\mathring{d}} \mathcal{N}_{d_{ic_{t+1};t+1}}(\theta'_{ic_{t+1}}\psi_{ic_{t+1};t}, r_{ic_{t+1}}).$$

4. *Go to the beginning of* Sequential mode.

Proposition 9.2.4, combined with the receding-horizon, certainty-equivalent strategy, justifies the following adaptive design algorithm.

Algorithm 9.2.7 (Adaptive simultaneous advising based on γ-bound of the KL distance)
Initial (off line) mode

- *Estimate normal mixture model of the o-system with the state* ϕ_t, *Chapter 8.*
- *Specify the true user ideal pdf on the response of the o-system*

$$ {}^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta'_i\psi_{i;t},\ {}^{\lfloor U}r_i\right) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{i-\mathring{\Delta};t}}\left({}^{\lfloor U}\theta'_i\psi_{i;t},\ {}^{\lfloor U}r_i\right).$$

- *Specify* data invariant *part* ${}^{\lfloor U}f(c_t)$ *of the user ideal on the recommended pointers.*
- *Specify* data invariant *lifts* ${}^{\lfloor U}k_{c;t}$ *and the kernels* ${}^{\lfloor U}L_{c;t}$, ${}^{\lfloor U}D_{c;t}$ *that define the user ideal pf on recommended pointers*

$$f(c_t|u_{o;t}, d(t-1)) \propto {}^{\lfloor U}f(c_t) \exp\left[-0.5\left({}^{\lfloor U}k_{c_t;t-1} + \psi'_t\, {}^{\lfloor U}L_{c_t;t-1}\, {}^{\lfloor U}D_{c_t;t-1}\, {}^{\lfloor U}L'_{c_t;t-1}\psi_t\right)\right].$$

- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update estimates of the model parameters, Section 8.5.*
3. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $L_{a;\tau} = I_{\mathring{\phi}},\ D_{a;\tau} = 0_{\mathring{\phi},\mathring{\phi}}$.

$$L_{a;\tau} \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_{a;\tau} & 0 \\ {}^{\lfloor \phi 1}L'_{a;\tau} & 1 \end{bmatrix},\ D_{a;\tau} \equiv \left[\mathrm{diag}\left[{}^{\lfloor \phi 0}D_{a;\tau}\right],\ {}^{\lfloor 1}D_{a;\tau}\right],\ {}^{\lfloor 1}D_{a;\tau} \text{ is scalar.}$$

Omit the initialization of $L_{a;\tau}$, $D_{a;\tau}$ *if* $t > 1$ *and the IST strategy is adopted.*

Iterative mode

Correct the lifted quadratic forms defining the KL distances using the current point estimates of parameters

$$\text{Set } L_{a;t+T} = I_{\mathring{\phi}} = \mathring{\phi}\text{-unit matrix},\ D_{a;t+T} = 0_{\mathring{\phi},\mathring{\phi}} \equiv (\mathring{\phi},\mathring{\phi})\text{-zero matrix.}$$

For $\tau = t+T, \ldots, t+1$

$$q = 0$$

For $c = 1, \ldots, \mathring{c}$

$$L_{0c} \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_{a;\tau} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor \phi 1}L'_{a;\tau} & 0 & 1 \end{bmatrix},\ D_{0c} = \mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor \phi 0}D_{a;\tau}\right], 0_{1,\mathring{d}},\ {}^{\lfloor 1}D_{a;\tau}\right],$$

$$k_{0c} \equiv -\mathring{d}_o$$

For $i = 1, \ldots, \mathring{\Delta}$

$$L_{ic}D_{ic}L'_{ic} = {}^{\lfloor \psi}L_{(i-1)c}\,{}^{\lfloor \psi}D_{(i-1)c}\,{}^{\lfloor \psi}L'_{(i-1)c} +$$

$$+\left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right)\,{}^{\lfloor d}D_{(i-1)c}\left(\theta_{ic} + {}^{\lfloor d\psi}L_{(i-1)c}\right)' +$$

$$+\chi\left(i \le \mathring{\Delta}_o\right)\left(\theta_{ic} - {}^{\lfloor U}\theta_i\right)\,{}^{\lfloor U}r_i^{-1}\left(\theta_{ic} - {}^{\lfloor U}\theta_i\right)',$$

$$k_{ic} \equiv k_{(i-1)c} + {}^{\lfloor d}D_{(i-1)c}r_{ic} + \chi\left(i \le \mathring{\Delta}_o\right)\left[\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right) + \frac{r_{ic}}{{}^{\lfloor U}r_i}\right],\ \mathring{D}_{ic} = \mathring{D}_{(i-1)c} - 1$$

end of the cycle over i

$$L_{\mathring{\Delta}c}D_{\mathring{\Delta}c}L'_{\mathring{\Delta}c} \equiv L_{\mathring{\Delta}c}D_{\mathring{\Delta}c}L'_{\mathring{\Delta}c} + {}^{\lfloor U}L_{c;t-1}\,{}^{\lfloor U}D_{c;\tau-1}\,{}^{\lfloor U}L'_{c;\tau-1}$$

$$k_{\mathring{\Delta}c} \equiv k_{\mathring{\Delta}c} + {}^{\lfloor U}k_{c;\tau-1}$$

end of the cycle over c

$$L_{a;\tau-1} = I_{\mathring{\phi}},\ D_{a;\tau-1} = 0_{\mathring{\phi},\mathring{\phi}}$$

For $c = 1, \ldots, \mathring{c}$

For $i = \mathring{\Delta}+1, \ldots, \mathring{d}$

$$\tilde{L}_{ic}\tilde{D}_{ic}\tilde{L}'_{ic} = L_{(i-1)c}D_{(i-1)c}L'_{(i-1)c} + \left[-1,\ {}^{\lfloor U}\theta'_i\right]'\,{}^{\lfloor U}r_i^{-1}\left[-1,\ {}^{\lfloor U}\theta'_i\right]$$

$$\tilde{L}_{ic} = \begin{bmatrix} 1 & 0 \\ -\,{}^{\lfloor I}\theta_{ic;\tau-1} & L_{ic} \end{bmatrix}$$

$$\tilde{D}_{ic} = \mathrm{diag}\left[{}^{\lfloor I}r^{-1}_{ic;\tau-1}, D_{ic}\right],\ {}^{\lfloor I}r_{ic;\tau-1} \text{ is scalar,}$$

$$k_{ic} = k_{(i-1)c} + \ln\left({}^{\lfloor U}r_i\,{}^{\lfloor I}r^{-1}_{ic;\tau-1}\right)$$

end of the cycle over i

$$L_{c;\tau-1} \equiv L_{\mathring{d}c},\ D_{c;\tau-1} \equiv D_{\mathring{d}c},\ k_{c;\tau-1} \equiv k_{\mathring{d}c}$$

$$\beta_c \equiv {}^{\lfloor U}f(c)\exp(-0.5k_{\mathring{d}c})$$

$$q = q + \beta_c$$

end of the cycle over c

For $c = 1, \ldots, \mathring{c}$

$$\beta_c = \frac{\beta_c}{q}$$

$$L_{a;\tau-1} D_{a;\tau-1} L'_{a;\tau-1} = L_{a;\tau-1} D_{a;\tau-1} L'_{a;\tau-1} + \beta_c L_{c;\tau-1} D_{c;\tau-1} L'_{c;\tau-1}.$$

end of the cycle over c

end of the cycle over τ

The updating of the LDL′ (!) decomposition can be done by Algorithm 8.1.2.

4. *Evaluate the ideal pf on pointers*

$$^{\lfloor I}f(c_{t+1}|\phi_t) \quad \propto \quad ^{\lfloor U}f(c_{t+1}) \exp[-0.5(k_{c_{t+1};t} + \phi'_t L_{c_{t+1};t} D_{c_{t+1};t} L'_{c_{t+1};t} \phi_t)].$$

5. *Present to the user projections of the ideal pdf*

$$\begin{aligned} ^{\lfloor I}f(d_{t+1}|\phi_t) \quad &= \quad \sum_{c_{t+1} \in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) \prod_{i=1}^{\mathring{\Delta}} \mathcal{N}_{d_{ic_{t+1};t+1}}(\theta'_{ic_{t+1}} \psi_{ic_{t+1};t}, r_{ic_{t+1}}) \times \\ &\times \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta})c_{t+1};t+1}} \left({}^{\lfloor I}\theta'_{ic_{t+1};t} \psi_{ic_{t+1};t}, {}^{\lfloor I}r_{ic_{t+1};t} \right). \end{aligned}$$

6. *Go to the beginning of* Sequential mode.

Remark(s) 9.2.5

1. *It is worth to stress that the user ideal pf on* c_t *is conditioned on* ψ_t. *Thus, it can modify dependence of* c_t *and* $u_{o;t}$. *In algorithmic terms, the constructed coefficients* $^{\lfloor I}\theta_{ic;t}$ *are modified by the optional* $^{\lfloor U}L_c$, *which is* $(\mathring{\psi}, \mathring{\psi})$*-dimensional matrix.*

2. *Other design variants, like selection of the most probable advices or the grouped version, can be constructed directly according to Chapter 7, using the above evaluations.*

9.3 Interaction with operator

Here, we elaborate design of strategies generating presentation and signaling actions, using the model presented in Section 7.1.3, in the case of normal mixtures and normal user target pdf.

9.3.1 Assigning priorities

We present a normal version of the choice of presentation priorities. It is based on Proposition 7.3.2, on restriction that $\mathring{p}_t = 1$ and simplifying assumption that the probabilities of pointers $c \in c^*$ to components used in the design are close to the estimated weights α_c

$$f(c|d_{p_t;t}, d(t-1)) \approx \alpha_c. \tag{9.62}$$

We also assume that the optimized component weights are approximated by data and time invariant values

$$^{\lfloor I}f(c|d(t-1)) \approx {}^{\lfloor I}\alpha_c. \tag{9.63}$$

Proposition 9.3.1 (Presentation minimizing the bound (7.45) with $\mathring{p}_t = 1$ & (9.62), (9.63))
Let us consider that the academic, industrial or simultaneous design has provided the optimal mixture

$$^{\lfloor I}f(d_t|d(t-1)) = \sum_{c_t \in c^*} {}^{\lfloor I}f(d_t|d(t-1), c_t)\, {}^{\lfloor I}f(c_t|d(t-1)).$$

Let the signaling strategy makes the operator fully alert, i.e. signaling actions $s(\mathring{t}) \equiv 1$. Let us assume that $\mathring{p}_t = 1$ and (9.62), (9.63) hold in addition to assumptions of Proposition 7.3.2. Let us specify the user <u>*data invariant*</u> *ideal pf ${}^{\lfloor U}f(p_t)$ on the set of possible presentation actions $p^* \equiv \{1, \ldots, \mathring{d}_o\}$.*

Let us define presentation strategy that assigns the higher priority to the entries of $d_{p_t;t}$ the higher are the values of the following pf

$$f(p_t|\phi_{t-1}) \propto {}^{\lfloor U}f(p_t) \exp\left[-0.5\left(k_{p_t} + \phi'_{t-1} L_{p_t;t} D_{p_t;t} L'_{p_t;t} \phi'_{t-1}\right)\right]. \tag{9.64}$$

The involved lifts and kernels are generated by the following algorithm.

$$\text{Set } L_{\mathring{t}} = I_{\mathring{\phi}},\ D_{\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}} \tag{9.65}$$

For $t = \mathring{t}, \ldots, 1$

$$q = 0$$

For $p_t = 1, \ldots, \mathring{d}_o$

Permute entry $d_{p_t;t}$

on the tail position in the used ideal pdf and quadratic form with kernel L_t, D_t

$${}^{\lfloor U}f(d_t|d(t-1)) \underbrace{\rightarrow}_{Proposition\ 9.1.1} \prod_{i \in i^*} \mathcal{N}_{d_{i;t}}\left({}^{\lfloor U}\theta'_{ip_t}\psi_{i;t},\ {}^{\lfloor U}r_{ip_t}\right)$$

$$L_t,\ D_t \underbrace{\rightarrow}_{Proposition\ 8.1.4} L_{p_t;t},\ D_{p_t;t}$$

$$\text{Set } L_{p_t} = I_{\mathring{\phi}+1},\ D_{p_t} = 0_{\mathring{\phi}+1,\mathring{\phi}+1},\ k_{p_t} = 0$$

For $c = 1, \ldots, \mathring{c}$

Permute entry $d_{p_t;t}$ *on the tail position in the estimated and ideal pdfs*

$${}^{\lfloor I}f(d_t|d(t-1), c) \underbrace{\rightarrow}_{Proposition\ 9.1.1} \prod_{i \in i^*} \mathcal{N}_{d_{i;t}}\left({}^{\lfloor I}\theta'_{icp_t}\psi_{i;t},\ {}^{\lfloor I}r_{icp_t}\right),$$

$$f(d_t|d(t-1), c) \underbrace{\rightarrow}_{Proposition\ 9.1.1} \prod_{i \in i^*} \mathcal{N}_{d_{i;t}}\left(\theta'_{icp_t}\psi_{i;t}, r_{icp_t}\right),$$

$$k_{0cp_t} = 0,\ L_{0cp_t} \equiv \begin{bmatrix} {}^{\lfloor \phi 0}L_{p_t;t} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor \phi 1}L'_{p_t;t} & 0 & 1 \end{bmatrix},$$

$$D_{0cp_t} \equiv \operatorname{diag}\left[\operatorname{diag}\left[{}^{\lfloor \phi 0}D_{p_t;t}\right], 0_{1,\mathring{d}}, {}^{\lfloor 1}D_{p_t;t}\right],\ \ {}^{\lfloor 1}D_{p_t;t} \text{ is scalar.}$$

For $i = 1, \ldots, \mathring{d} - 1$

$$L_{icp_t} D_{icp_t} L'_{icp_t} = {}^{\lfloor \psi}L_{(i-1)cp_t}\, {}^{\lfloor \psi}D_{(i-1)cp_t}\, {}^{\lfloor \psi}L'_{(i-1)cp_t} +$$

$$+ \left(\theta_{icp_t} + {}^{\lfloor d\psi}L_{(i-1)cp_t}\right) {}^{\lfloor d}D_{(i-1)cp_t} \left(\theta_{icp_t} + {}^{\lfloor d\psi}L_{(i-1)cp_t}\right)',$$

$$k_{icp_t} = k_{(i-1)cp_t} + {}^{\lfloor d}D_{(i-1)cp_t} r_{icp_t} \text{ with}$$

$$L_{(i-1)cp_t} = \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L_{(i-1)cp_t} & {}^{\lfloor \psi}L_{(i-1)cp_t} \end{bmatrix}$$

$$D_{(i-1)cp_t} = \operatorname{diag}\left[{}^{\lfloor d}D_{(i-1)cp_t}, {}^{\lfloor \psi}D_{(i-1)cp_t}\right]$$

${}^{\lfloor d}D_{(i-1)cp_t}$ *is scalar.*

end of the cycle over i

$$L_{p_t}D_{p_t}L'_{p_t} = L_{p_t}D_{p_t}L'_{p_t} + \alpha_c L_{(\mathring{d}-1)cp_t}D_{(\mathring{d}-1)cp_t}L'_{(\mathring{d}-1)cp_t},$$
$$k_{p_t} = k_{p_t} + \alpha_c k_{(\mathring{d}-1)cp_t}.$$

end of the cycle over c

$$L_{p_t} = {}^{\lfloor\psi}L_{p_t},\ D_{p_t} = {}^{\lfloor\psi}D_{p_t}, l_{p_t} = {}^{\lfloor d\psi}L_{p_t},\ \delta_{p_t} = {}^{\lfloor d}D_{p_t}.$$

For $c = 1, \ldots, \mathring{c}$

$$L_{p_t}D_{p_t}L'_{p_t} = L_{p_t}D_{p_t}L'_{p_t} + {}^{\lfloor I}\alpha_c \left\{ \left({}^{\lfloor I}\theta_{\mathring{d}cp_t} + l_{p_t}\right)\delta_{p_t}\left({}^{\lfloor I}\theta_{\mathring{d}cp_t} + l_{p_t}\right)' + \right.$$
$$\left. + \left({}^{\lfloor I}\theta_{\mathring{d}cp_t} - {}^{\lfloor U}\theta_{\mathring{d}p_t}\right) {}^{\lfloor U}r^{-1}_{\mathring{d}p_t} \left({}^{\lfloor I}\theta_{\mathring{d}cp_t} - {}^{\lfloor U}\theta_{\mathring{d}p_t}\right)'\right\},$$
$$k_{p_t} = k_{p_t} + {}^{\lfloor I}\alpha_c \left\{ {}^{\lfloor d}D_{cp_t}\ {}^{\lfloor I}r_{\mathring{d}cp_t} + \ln\left(\frac{{}^{\lfloor U}r_{\mathring{d}p_t}}{{}^{\lfloor I}r_{\mathring{d}cp_t}}\right) + \frac{{}^{\lfloor I}r_{\mathring{d}cp_t}}{{}^{\lfloor U}r_{\mathring{d}p_t}}\right\}.$$

end of the cycle over c

$$\beta_{p_t} = {}^{\lfloor U}f(p_t)\exp(-0.5k_{p_t}),\ q = q + \beta_{p_t}$$

end of the cycle over p_t

$$L_{t-1} = I_{\mathring{\phi}},\ D_{t-1} = 0_{\mathring{\phi},\mathring{\phi}}$$

For $p_t = 1, \ldots, \mathring{d}_o$

$$\beta_{p_t} = \frac{\beta_{p_t}}{q}$$
$$L_{t-1}D_{t-1}L'_{t-1} = L_{t-1}D_{t-1}L'_{t-1} + \beta_{p_t}L_{p_t}D_{p_t}L'_{p_t}.$$

end of the cycle over p_t

end of the cycle over t

This presentation strategy minimizes the upper bound on the KL distance, implied by the inequality described in Proposition 7.1.7 for horizon equal 1 and the γ*-bound, see Proposition 7.1.8, on it when* $\mathring{t} > 1$.

Proof: Let us assume that

$$-\ln(\gamma(d(t)) = 0.5\sum_{p=1}^{\mathring{d}_o}\beta_p\phi'_t L_{p;t}D_{p;t}L'_{p;t}\phi_t = 0.5\phi'_t L_t D_t L'_t\phi_t.$$

It is true for $t = \mathring{t}$ with $L_{\mathring{t}} = I_{\mathring{\phi}}$, $D_{\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$. As usual, the backward induction serves us for verifying this assumption and construction of the resulting algorithms.

For a generic t, the inductive assumption and the assumption (9.62) imply that the function, cf. (7.48),

$$\omega(c, d_{p_t;t}, d(t-1)) \quad = \quad \int f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c)\ln\left(\frac{f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1), c)}{\gamma(d(t))\,{}^{\lfloor U}f(d_{\bar{p}_t;t}|d_{p_t;t}, d(t-1))}\right) dd_{\bar{p}_t;t}$$

has to be evaluated. Note, that the first term in its definition is zero due to the assumption (9.62). In order to make Proposition 9.1.7 applicable (it describes evaluation of the weighted conditional KL distance) we permute the considered entry $d_{p_t;t}$ on the last position in d_t. It is achieved by a sequential use of Proposition 9.1.1. At the same time, the corresponding permutations of the kernel determining $\gamma(d(t))$ have to be applied, see Proposition 8.1.4. These permutation have to be also made both for the normal components ${}^{\lfloor I}f(d_t|d(t-1), c)$ and the user ideal pdf ${}^{\lfloor U}f(d_t|d(t-1))$.

After all these permutations, the application of Proposition 9.1.7 gives recursions, applied for the dimension $\mathring{d}-1$ for computing $\omega(c, d_{p_t;t}, d(t-1)) = k_{c,p_t} + [d_{p_t;t}, \phi'_{t-1}]L_{cp_t;t-1}D_{cp_t;t-1}L'_{cp_t;t-1}[d_{p_t;t}, \phi'_{t-1}]'$.

The assumption (9.62) also implies that the second term in the definition of $\omega(p_t, d(t-1))$, see (7.48), becomes

$${}^{\lfloor I}\mathcal{E}\left\{\sum_{c\in c^*}\alpha_c\left[k_{c,p_t} + [d_{p_t;t}, \phi'_{t-1}]L_{cp_t;t-1}D_{cp_t;t-1}L'_{cp_t;t-1}[d_{p_t;t}, \phi'_{t-1}]'\right]|p_t, \phi_{t-1}, c_t\right\} \equiv$$

$$\equiv \quad k_{c_t,p_t;t-1} + \phi'_{t-1} L_{c_t p_t;t-1} D_{c_t p_t;t-1} L'_{c_t p_t;t-1} \phi_{t-1},$$

where ${}^{\lfloor I}\mathcal{E}\{\cdot\}$ stresses that the expectation is taken with respect to the optimal mixture ${}^{\lfloor I}f(d_{p_t;t}|d(t-1), c_t)$ obtained in previous design.

The result is obtained in two steps. First, the computationally demanding mixing of factorized lifts k_{c,p_t} and kernels $L_{cp_t;t-1} D_{cp_t;t-1} L'_{cp_t;t-1}$ is made. Then, the integration over $d_{p_t;t}$ is made according to Proposition 9.1.4. In the same cycle over $c \in c^*$, the conditional KL distance is evaluated

$$\int {}^{\lfloor I}f(d_{p_t;t}|d(t-1), c_t) \ln\left(\frac{{}^{\lfloor I}f(d_{p_t;t}|d_{p+;t}, d(t-1), c_t)}{{}^{\lfloor U}f(d_{p_t;t}|d(t-1), c_t)}\right) dd_{p_t;t} =$$
$$= \quad {}^{\lfloor I}k_{c_t p_t;t-1} + \phi'_{t-1}\, {}^{\lfloor I}L_{c_t p_t;t-1}\, {}^{\lfloor I}D_{c_t p_t;t-1}\, {}^{\lfloor I}L'_{c_t p_t;t-1}\phi_{t-1}$$

using Proposition 9.1.6. Adopting the assumption (9.63), we mix intermediate results, see (7.48), and get

$$\omega_\gamma(p_t, d(t-1)) = \sum_{c_t \in c^*} {}^{\lfloor I}\alpha_{c_t} \left[{}^{\lfloor I}k_{c_t p_t;t-1} + k_{c_t p_t;t-1} + \right.$$
$$+ \quad \left. \phi'_{t-1}\left(L_{c_t p_t;t-1} D_{c_t p_t;t-1} L'_{c_t p_t;t-1} + {}^{\lfloor I}L_{c_t p_t;t-1}\, {}^{\lfloor I}D_{c_t p_t;t-1}\, {}^{\lfloor I}L'_{c_t p_t;t-1}\right)\phi_{t-1}\right] \equiv$$
$$\equiv \quad {}^{\lfloor I}k_{p_t;t-1} + \phi'_{t-1}\, {}^{\lfloor I}L_{p_t;t-1}\, {}^{\lfloor I}D_{p_t;t-1}\, {}^{\lfloor I}L'_{p_t;t-1}\phi_{t-1}.$$

It defines the pf on the presentation priorities. Its normalizing factor used in the next optimization steps is approximated by replacing weighted arithmetic mean by its smaller geometric counterpart. It guarantees that an upper bound is minimized and the assumed form of γ is reproduced with

$$\omega(d(t-1)) = \phi'_{t-1}\left[\sum_{p_t \in p^*} \beta_{p_t}\, {}^{\lfloor I}L_{p_t;t-1}\, {}^{\lfloor I}D_{p_t;t-1}\, {}^{\lfloor I}L'_{p_t;t-1}\right]\phi_{t-1} \quad \text{with } \beta_{p_t} \equiv {}^{\lfloor U}f(p_t)\exp\left(-0.5 k_{p_t;t-1}\right).$$

□

Algorithm 9.3.1 (Presentation minimizing the bound (7.45) with $\mathring{p}_t = 1$ & (9.62), (9.63))
Initial (off line) mode

- *Estimate the normal mixture model of the o-system with the state ϕ_t, Chapter 8.*
- *Specify the normal user ideal pdf on d_t, c_t*

$$\begin{aligned}
{}^{\lfloor U}f(d_t, c_t|d(t-1)) = \; & {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1)) \\
& f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1)) \propto \\
\propto \; & \prod_{i=1}^{\mathring{\Delta}_o} \mathcal{N}_{\Delta_{i;t}}\left({}^{\lfloor U}\theta'_i\psi_{i;t},\, {}^{\lfloor U}r_i\right) \prod_{i=\mathring{\Delta}_o+1}^{\mathring{\Delta}} \mathcal{N}_{\Delta_{i;t}}\left(\theta'_i\psi_{i;t}, r_i\right) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor U}\theta'_i\psi_{i;t},\, {}^{\lfloor U}r_i\right) \\
& {}^{\lfloor U}f(c_t)\exp\left[-0.5\left({}^{\lfloor U}k_{c_t;t-1} + \psi'_t\, {}^{\lfloor U}L_{c_t;t-1}\, {}^{\lfloor U}D_{c_t;t-1}\, {}^{\lfloor U}L'_{c_t;t-1}\psi_t\right)\right].
\end{aligned}$$

- *Select the user ideal pf ${}^{\lfloor U}f(p_t|d(t-1))$ on the scalar priority actions $p_t \in p^* \equiv \{1, \ldots, \mathring{d}\}$.*
- *Select the length of the receding horizon $T \geq 1$.*

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Update estimates of the model parameters, Section 8.5, if you deal with the adaptive advisory system.*

3. *Design academic, industrial or simultaneous strategy generating the ideal pdf*

$$
\begin{aligned}
{}^{\lfloor I}f(d_t|\phi_{t-1}) &= \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1})\, {}^{\lfloor I}f(d_t|\phi_{t-1},c_t) = \\
&= \sum_{c_t\in c^*} \prod_{i=1}^{\mathring{\Delta}} \mathcal{N}_{d_{i;t}}(\theta'_{ic}\psi_{i;t}, r_{ic}) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \mathcal{N}_{u_{o(i-\mathring{\Delta});t}}\left({}^{\lfloor I}\theta'_{ic;t-1}\psi_{i;t},\, {}^{\lfloor I}r_{ic;t-1}\right) {}^{\lfloor I}f(c_t|\phi_{t-1}), \\
{}^{\lfloor I}f(c_{t+1}|\phi_t) &\propto {}^{\lfloor U}f(c_{t+1}) \exp\left[-0.5\left(k_{c_{t+1}} + \phi'_{t-1}L_{c_{t+1}}D_{c_{t+1}}L'_{c_{t+1}}\phi_{t-1}\right)\right].
\end{aligned}
$$

4. *Initialize the iterative mode by setting* $\tau = t+T$ *and* $L_\tau = I_{\mathring{\phi}},\ D_\tau = 0_{\mathring{\phi},\mathring{\phi}}$.
   The initialization of $L_\tau,\ D_\tau$ *is skipped if* $t > 1$ *and the IST strategy is used.*
5. *Apply the iterative part of the algorithm described in Proposition 9.3.1 on the range* $[t+1, t+T]$. *It gives the pf* $f(p_{t+1}|d(t))$.
6. *Order the values* ${}^{\lfloor I}f(p_{t+1}|\phi_t) \propto {}^{\lfloor U}f(p_{t+1}|\phi_t)\exp[-\omega_\gamma(p_{t+1},\phi_t)]$.
7. *Present projections of the ideal pdf* ${}^{\lfloor I}f(d_{\hat{p}_{t+1};t+1}|\phi_t)$ *to the user. The indices* $\hat{p}_{t+1}$ *are chosen so that they have the highest values of* ${}^{\lfloor I}f(p_{t+1}|\phi_t)$.
8. *Go to the beginning of* Sequential mode.

Problem 9.3.1 (Improvements of presentation strategies) *The presented solution is far from being satisfactory. In spite of significant approximations made, the result is cumbersome and highly computationally demanding. Probably, a direct use of these approximations before estimating the KL distance could simplify the solution.*

Alternatively, academic, industrial or simultaneous designs with $d_{o;t} = d_{p;t}$ *with several different p-th in* $\{1,\ldots,\mathring{d}_o\}$ *could be performed and those options resulting to the smallest KL distances presented. Logic behind is obvious: the operator can optimize only those variables about which he is aware. This justifies the reduction of his observable data space to the presented data space.*

Of course, other directions have to be thought of, too.

9.3.2 Stimulating the operator

The signaling strategy makes the operator alert. It asks him to follow the advised actions, when the ideal pdf, resulting from academic, industrial or simultaneous design, gives a significantly smaller KL distance to the user ideal pdf than the KL distance of the estimated model to this user ideal pdf.

The model relating the signaling action s_t to the guided-system response is

$$
\begin{aligned}
{}^{\lfloor I}f(d_t,s_t|d(t-1)) &\equiv {}^{\lfloor I}f(d_t|s_t,d(t-1))\, {}^{\lfloor I}f(s_t|d(t-1)),\ s_t\in s^*\equiv\{0,1\} \text{ with} \qquad (9.66)\\
{}^{\lfloor I}f(d_t|s_t=0,d(t-1)) &\equiv f(d_t|d(t-1)) \equiv \sum_{c\in c^*}\alpha_c \prod_{i=1}^{\mathring{d}} \mathcal{N}_{d_{i;t}}(\theta'_{ic}\psi_{i;t}, r_{ic}) \equiv \text{learnt mixture}\\
{}^{\lfloor I}f(d_t|s_t=1,d(t-1)) &\equiv {}^{\lfloor I}f(d_t|d(t-1)) \equiv \\
&\equiv \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1}) \prod_{i=1}^{\mathring{d}} \mathcal{N}_{d_{i;t}}\left({}^{\lfloor I}\theta'_{ic}\psi_{i;t},\, {}^{\lfloor I}r_{ic}\right) \equiv \text{designed mixture and}\\
{}^{\lfloor I}f(c_t|\phi_{t-1}) &\propto {}^{\lfloor U}f(c_t)\exp\left\{-0.5\left[k_{c_t;t-1} + \phi'_{t-1}L_{c_t;t-1}D_{c_t;t-1}L'_{c_t;t-1}\phi_{t-1}\right]\right\}.
\end{aligned}
$$

Note that ${}^{\lfloor I}\theta'_{ic} = \theta'_{ic}$, ${}^{\lfloor I}r_{ic} = r_{ic}$ for $i \le \mathring{\Delta}$ as the ideal pdfs use the estimated factors predicting innovations without any change.

The model (9.66) is a special version of the model (9.37) used in the academic design. Thus, the design of the signaling strategy reduces to it. It is reasonable to assume that periods of operator activity or non-activity have to be relatively long. Thus, the grouped version of the design, Proposition 7.2.4 is to be used.

Proposition 9.3.2 (Grouped signaling minimizing γ-bound on Jensen distance) *Let us consider that the academic, industrial or simultaneous design of the advisory system has provided the ideal pdf* $^{\lfloor I}f(d_t|\phi_{t-1})$ *that is built into the used model (9.66).*

Then, the following signaling strategy minimizes the γ-type bound on the KL distance under the grouping restriction $f(s_\tau|d(tn)) = f(s_{tn+1}|d(tn))$ *for* $\tau \in \{tn+1,\ldots,t(n+1)\}$.

$$f(s_{nt+1}|\phi_{nt}) \propto {}^{\lfloor U}f(s_{nt+1}|\phi_{nt}) \exp\left[-0.5\left(k_{s_{nt+1};nt+1} + \phi'_{nt} L_{s_{nt+1};nt+1} D_{s_{nt+1};nt+1} L'_{s_{nt+1};nt+1}\phi_{nt}\right)\right].$$

It is defined for $s_{nt+1} \in \{0,1\}$ *and determined as follows starting with* $L_{a;\mathring{t}} = I_{\mathring{\phi}},\ D_{a;\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$ (9.67)

For $t = \mathring{t},\ldots,\mathring{t} - i\times n+1,\ldots,1$

$q = 0$

For $s = 0,\ldots,1$

$L_{s;t-n} = I_{\mathring{\phi}},\ D_{s;t-n} = 0_{\mathring{\phi},\mathring{\phi}},\ k_s = 0$

For $c = 1,\ldots,\mathring{c}$

$L_{00sc} \equiv L_{a;t},\ D_{00sc} \equiv D_{a;t},\ k_{00sc} = 0.$

For $\tilde{n} = 1,\ldots,n$

$$L_{0\tilde{n}sc} = \begin{bmatrix} {}^{\lfloor\phi 0}L_{\mathring{d}(\tilde{n}-1)sc} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor\phi 1}L'_{\mathring{d}(\tilde{n}-1)sc} & 0 & 1 \end{bmatrix}$$

$$D_{00sc} = \mathrm{diag}\left[\mathrm{diag}\left[{}^{\lfloor\phi 0}D_{\mathring{d}(\tilde{n}-1)sc}\right], 0_{1,\mathring{d}}, {}^{\lfloor 1}D_{\mathring{d}(\tilde{n}-1)sc}\right].$$

$k_{0\tilde{n}sc} = k_{\mathring{d}(\tilde{n}-1)sc}$. *It uses the split*

$$L_{0(\tilde{n}-1)sc} \equiv \begin{bmatrix} {}^{\lfloor\phi 0}L_{\mathring{d}(\tilde{n}-1)sc} & 0 \\ {}^{\lfloor\phi 1}L'_{\mathring{d}(\tilde{n}-1)sc} & 1 \end{bmatrix}$$

$$D_{0(\tilde{n}-1)sc} \equiv \mathrm{diag}\left[{}^{\lfloor\phi 0}D_{\mathring{d}(\tilde{n}-1)sc}, {}^{\lfloor 1}D_{\mathring{d}(\tilde{n}-1)sc}\right]$$

${}^{\lfloor 1}D_{\mathring{d}(\tilde{n}-1)sc}$ *is scalar and*

For $i = 1,\ldots,\mathring{d}$

$$L_{(i-1)\tilde{n}sc}\begin{bmatrix} 1 & 0 \\ {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc} & {}^{\lfloor\psi}L_{(i-1)\tilde{n}sc} \end{bmatrix}$$

$$D_{(i-1)\tilde{n}sc} \equiv \mathrm{diag}\left[{}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}, {}^{\lfloor\psi}D_{(i-1)\tilde{n}sc}\right]$$

${}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}$ *is scalar.*

$$\begin{aligned}
L_{i\tilde{n}sc}D_{i\tilde{n}sc}L'_{i\tilde{n}sc} &= {}^{\lfloor\psi}L_{(i-1)\tilde{n}sc}\,{}^{\lfloor\psi}D_{(i-1)\tilde{n}sc}\,{}^{\lfloor\psi}L'_{(i-1)\tilde{n}sc} + \\
&+\chi\left(i \le \mathring{\Delta}_o\right)\left[\left(\theta_{ic} + {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right){}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}\left(\theta_{ic} + {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right)' + \right. \\
&\left. + \left(\theta_{ic} - {}^{\lfloor U}\theta_{ic}\right){}^{\lfloor U}r_i^{-1}\left(\theta_{ic} - {}^{\lfloor U}\theta_{ic}\right)'\right] + \\
&+\chi\left(\mathring{\Delta}_o < i \le \mathring{\Delta}\right)\left(\theta_{isc} + {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right){}^{\lfloor\Delta}D_{(i-1)sc}\left(\theta_{isc} + {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right)' + \\
&+\chi\left(\mathring{\Delta} < i\right)\left[\left(\theta_{isc} + {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right){}^{\lfloor\Delta}D_{(i-1)sc}\left(\theta_{isc} + {}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right)' + \right. \\
&\left. + \left(\theta_{isc} - {}^{\lfloor U}\theta_{ic}\right){}^{\lfloor U}r_i^{-1}\left(\theta_{isc} - {}^{\lfloor U}\theta_{ic}\right)'\right]
\end{aligned}$$

$$\begin{aligned}
k_{i\tilde{n}sc} &= k_{i\tilde{n}sc} + \chi\left(i \le \mathring{\Delta}_o\right)\left(k_{(i-1)\tilde{n}sc} + {}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}r_{ic}\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right) + \frac{r_{ic}}{{}^{\lfloor U}r_i}\right) + \\
&+\chi\left(\mathring{\Delta}_o < i \le \mathring{\Delta}\right){}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}r_{isc} + \chi\left(\mathring{\Delta} < i\right)\left({}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}r_{isc} + \ln\left(\frac{{}^{\lfloor U}r_i}{r_{isc}}\right) + \frac{r_{isc}}{{}^{\lfloor U}r_i}\right)
\end{aligned}$$

end of the cycle over i

end of the cycle over $\tilde{n}$

end of the cycle over c

For $c = 1, \ldots, \mathring{c}$

$$L_{s;t-n}D_{s;t-n}L'_{s;t-n} = L_{s;t-n}D_{s;t-n}L'_{s;t-n} + \chi(s=0)\left\{\frac{\alpha_c}{n}L_{\mathring{d}nsc}D_{\mathring{d}nsc}L'_{\mathring{d}nsc}\right\} +$$
$$+\chi(s=1)\left\{ {}^{\lfloor U}f(c)\left(\frac{1}{n}L_{\mathring{d}nsc}D_{\mathring{d}nsc}L'_{\mathring{d}nsc} + {}^{\lfloor I}L_c\,{}^{\lfloor I}D_c\,{}^{\lfloor I}L'_c\right)\right\}$$
$$k_{s;t-n} = k_{s;t-n} + \chi(s=0)\frac{\alpha_c}{n}k_{\mathring{d}nsc} + \chi(s=1)\left\{ {}^{\lfloor U}f(c)\left(\frac{1}{n}k_{\mathring{d}n1c} + {}^{\lfloor I}k_c\right)\right\}$$

end of the cycle over c

$$\beta_s \equiv {}^{\lfloor U}f(s)\exp\left(-0.5k_{s;t-n}\right),\ q = q + \beta_s$$

end of the cycle over s

$$L_{a;t-n}D_{a;t-n}L'_{a;t-n} \equiv \frac{\beta_0}{q}L_{0;t-n}D_{0;t-n}L'_{0;t-n} + \frac{\beta_1}{q}L_{1;t-n}D_{1;t-n}L'_{1;t-n}.$$

end of the cycle over t

Proof: Let us assume that

$$\begin{aligned}
-\ln(\gamma(d(n(t+1)))) &= 0.5\sum_{s_{n(t+1)}\in s^*}\beta_{s_{n(t+1)};t}\phi'_{n(t+1)}L_{s_{n(t+1)};n(t+1)}D_{s_{n(t+1)};n(t+1)}L'_{s_{n(t+1)};n(t+1)}\phi_{n(t+1)} = \\
&= 0.5\phi'_{n(t+1)}L_{a;n(t+1)}D_{a;n(t+1)}L'_{a;n(t+1)}\phi_{n(t+1)}, \\
\beta_{s_{n(t+1)};n(t+1)} &\propto {}^{\lfloor U}f(s_{n(t+1)})\exp[-0.5k_{s_{n(t+1)};n(t+1)}].
\end{aligned}$$

The lifts $k_{s_t;t}$ as well as the kernels are assumed to be independent data. It is true for $n(t+1) = \mathring{t}$ with $k_{s_{\mathring{t}};\mathring{t}} = 0,\ L_{a;\mathring{t}} = I_{\mathring{\phi}},\ D_{a;\mathring{t}} = 0_{\mathring{\phi},\mathring{\phi}}$. We use the backward induction to verify this assumption and to derive the design algorithm.

Let us adopt the inductive assumption, approximate the KL distance from above using Jensen inequality (2.14) and evaluate, cf. Proposition 7.2.4, $\omega_\gamma(s_{nt+1} = s, c, d(nt)) \equiv$

$$\equiv \frac{1}{n}\sum_{\tau=nt+1}^{n(t+1)}\int f(d_\tau|d(nt), s_{nt+1}, c)\ln\left(\frac{f(d_{o;\tau}|d_{p+;\tau}, d(nt), s_{nt+1}, c)}{\gamma(d(n(t+1)))\,{}^{\lfloor U}f(d_{o;\tau}|d_o(\tau-1))}\right)dd_\tau \underbrace{=}_{Proposition\ 9.1.6}$$

$= k_{n\mathring{d}sc} + \phi'_{nt}L_{n\mathring{d}sc}D_{n\mathring{d}sc}L'_{n\mathring{d}sc}\phi'_{nt}$ with kernels and lifts found recursively

$L_{00sc} \equiv L_{a;nt},\ D_{00sc} \equiv D_{a;nt}, k_{00sc} = 0$

For $\tilde{n} = 1, \ldots, n$

$$L_{0\tilde{n}sc} = \begin{bmatrix} {}^{\lfloor \phi 0}L_{\mathring{d}(\tilde{n}-1)sc} & 0 & 0 \\ 0 & I_{\mathring{d}} & 0 \\ {}^{\lfloor \phi 1}L'_{\mathring{d}(\tilde{n}-1)sc} & 0 & 1 \end{bmatrix},\quad D_{00sc} = \operatorname{diag}\left[\operatorname{diag}\left[{}^{\lfloor \phi 0}D_{\mathring{d}(\tilde{n}-1)sc}\right], 0_{1,\mathring{d}}, {}^{\lfloor 1}D_{\mathring{d}(\tilde{n}-1)sc}\right]$$

$k_{0\tilde{n}sc} = k_{\mathring{d}(\tilde{n}-1)sc}$

For $i = 1, \ldots, \mathring{d}$

$$L_{i\tilde{n}sc}D_{i\tilde{n}sc}L'_{i\tilde{n}sc} = {}^{\lfloor \psi}L_{(i-1)\tilde{n}sc}\,{}^{\lfloor \psi}D_{(i-1)\tilde{n}sc}\,{}^{\lfloor \psi}L'_{(i-1)\tilde{n}sc} +$$
$$+\chi\left(i \le \mathring{\Delta}_o\right)\left[\left(\theta_{ic} + {}^{\lfloor \Delta\psi}L_{(i-1)\tilde{n}sc}\right){}^{\lfloor \Delta}D_{(i-1)\tilde{n}sc}\left(\theta_{ic} + {}^{\lfloor \Delta\psi}L_{(i-1)\tilde{n}sc}\right)' +\right.$$
$$\left. + \left(\theta_{ic} - {}^{\lfloor U}\theta_{ic}\right){}^{\lfloor U}r_i^{-1}\left(\theta_{ic} - {}^{\lfloor U}\theta_{ic}\right)'\right] +$$
$$+\chi\left(\mathring{\Delta}_o < i \le \mathring{\Delta}\right)\left(\theta_{isc} + {}^{\lfloor \Delta\psi}L_{(i-1)\tilde{n}sc}\right){}^{\lfloor \Delta}D_{(i-1)sc}\left(\theta_{isc} + {}^{\lfloor \Delta\psi}L_{(i-1)\tilde{n}sc}\right)' +$$

$$+\chi\left(\mathring{\Delta}<i\right)\left[\left(\theta_{isc}+{}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right){}^{\lfloor\Delta}D_{(i-1)sc}\left(\theta_{isc}+{}^{\lfloor\Delta\psi}L_{(i-1)\tilde{n}sc}\right)'+\right.$$
$$\left.+\left(\theta_{isc}-{}^{\lfloor U}\theta_{ic}\right){}^{\lfloor U}r_i^{-1}\left(\theta_{isc}-{}^{\lfloor U}\theta_{ic}\right)'\right]$$
$$k_{i\tilde{n}sc}=k_{i\tilde{n}sc}+\chi\left(i\leq\mathring{\Delta}_o\right)\left(k_{(i-1)\tilde{n}sc}+{}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}r_{ic}\ln\left(\frac{{}^{\lfloor U}r_i}{r_{ic}}\right)+\frac{r_{ic}}{{}^{\lfloor U}r_i}\right)+$$
$$+\chi\left(\mathring{\Delta}_o<i\leq\mathring{\Delta}\right){}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}r_{isc}+\chi\left(\mathring{\Delta}<i\right)\left({}^{\lfloor\Delta}D_{(i-1)\tilde{n}sc}r_{isc}+\ln\left(\frac{{}^{\lfloor U}r_i}{r_{isc}}\right)+\frac{r_{isc}}{{}^{\lfloor U}r_i}\right)$$

end of the cycle over i

end of the cycle over $\tilde{n}$

Note that splitting to subsets respects both the distinction of the o- and p-data and the fact that parameters of factors predicting innovations are independent of the signaling action s_t. For specification of the optimal strategy, it remains to mix contributions of distances of individual components. We take into account the form of the component weights for the estimated and designed models. For it we set, $L_0=L_1=I_{\mathring{\phi}}$, $D_0=D_1=0_{\mathring{\phi},\mathring{\phi}}$, $k_0=k_1=0$.

For $c=1,\ldots,\mathring{c}$

$$L_0D_0L_0'=L_0D_0L_0'+\frac{\alpha_c}{n}L_{\mathring{d}n0c}D_{\mathring{d}n0c}L_{\mathring{d}n0c}',\ k_0=k_0+\frac{\alpha_c}{n}k_{\mathring{d}n0c}$$
$$L_1D_1L_1'=L_1D_1L_1'+{}^{\lfloor U}f(c)\left(\frac{1}{n}L_{\mathring{d}n1c}D_{\mathring{d}n1c}L_{\mathring{d}n1c}'+{}^{\lfloor I}L_c\,{}^{\lfloor I}D_c\,{}^{\lfloor I}L_c'\right)$$
$$k_1=k_1+{}^{\lfloor U}f(c)\left(\frac{1}{n}k_{\mathring{d}n1c}+{}^{\lfloor I}k_c\right).$$

end of the cycle over c

The standard minimization implies that the optimal strategy is

$${}^{\lfloor I}f(s_{nt+1}|d(nt))\propto{}^{\lfloor U}f(s_t)\exp\left[-0.5\left(k_{s_{nt+1}}+\phi_{nt}'L_{s_{nt+1}}D_{s_{nt+1}}L_{s_{nt+1}}'\phi_{nt}'\right)\right].$$

Its normalizing factor $\gamma(\phi_{nt})$ defines exact Bellman function to be transferred to the next optimization step. Use of the inequality between the arithmetic and geometric means preserves the assumed form of $-\ln(\gamma(d(nt))$ while guaranteeing that the upper bound is minimized. □

Remark(s) 9.3.1

1. *Probability of ${}^{\lfloor I}f(s_{tn+1}=0|\phi_{tn})$ is to be mapped on traffic lights. Psychologically, it seems to be reasonable to use a nonlinear mapping given by a pair of thresholds $0<w<W<1$*

$$\text{Call for action if } {}^{\lfloor I}f(s_{tn+1}=0|\phi_{tn})\leq w;\quad \text{allow none if } {}^{\lfloor I}f(s_{tn+1}=0|\phi_{tn})\geq W>w. \tag{9.68}$$

2. *The common factors of both models can be exploited when grouping rate $n=1$. Otherwise, this property brings no advantage.*

3. *Attempts to use ω-bound were done but with no significant influence on the algorithm. It worth of returning to them later on.*

The result is rather complex. Thus, a fixed version of signaling will be probably used. Let us write down the corresponding fixed signaling algorithm.

Algorithm 9.3.2 (Fixed signaling based on Proposition 9.3.2)
Initial (off line) mode

- *Estimate normal mixture model of the o-system with the state* ϕ_t*, Chapter 8.*
- *Specify the true user ideal pdf on* $d_{o;t}, c_t$*.*
- *Specify the user "alarm" pf* ${}^{\lfloor U}f(s_t = 0)$ *.*
- *Perform a fixed academic, industrial or simultaneous design.*
- *Compute steady state lifts* k_s *and kernels* $L_s, D_s,\ s \in \{0,1\}$ *according to Proposition 9.3.2.*
- *Specify thresholds* w, W *determining the mapping (9.68).*

Sequential (on line) mode, *running for t=1,2,...,*

1. *Acquire* d_t *and evaluate the state* ϕ_t*.*
2. *Evaluate signaling probabilities* $f(s_{t+1}|\phi_t) \propto {}^{\lfloor U}f(s_{t+1}) \exp\left[-0.5\left(k_{s_{t+1}} + \phi_t' L_{s_{t+1}} D_{s_{t+1}} L_{s_{t+1}}' \phi_t\right)\right]$.
3. *Convert the value* $f(s_{t+1} = 0|\phi_t)$ *into color using the non-linearity (9.68). The call action is the stronger the closer is* $f(s_{t+1} = 0|\phi_t)$ *to zero.*
4. *Go to the beginning of* Sequential mode.

Problem 9.3.2 (Completion and development of advising design) *The presented algorithms cover all formulated tasks. The overall state is, however, far from being satisfactory. The following problems have to be at least addressed*

- *completion and comparison of all variants,*
- *design of new variants,*
- *simplification,*
- *guide of choosing of tuning knobs,*
- *inclusion of stopping rules,*
- *analysis of theoretical properties known from control area (like controllability),*
- *robustness analysis, ...*

Chapter 10

Learning with Markov chain factors and components

Markov chain factors allow us to incorporate logical variables and to work with mixed real and discrete components. A detailed description of all learning aspects related to them forms the content of Chapter.

We deal with Markov chain factors predicting scalar discrete-valued $d_t \in d^* \equiv \{1, \ldots, \mathring{d}\}$. The Markov chain factor is described by the pf

$$f(d_t|d(t-1), \Theta) \equiv f(d_t|\psi_t, \Theta) = \prod_{d \in d^*} \prod_{\psi \in \psi^*} \left[{}^{\lfloor d|\psi}\Theta \right]^{\delta_{[d,\psi']',\Psi_t}}, \text{ where} \tag{10.1}$$

$\psi_t \in \psi^* \equiv \{1, \ldots, \mathring{\psi} < \infty\}$ is a finite-dimensional regression vector with a finite number of different values that can be constructed in a recursive way from the observed data;

$\Psi_t \equiv [d_t, \psi_t']'$ is a finite-dimensional finite-valued data vector;

$\Theta \in \Theta^* \equiv \left\{ {}^{\lfloor d|\psi}\Theta \geq 0, \ \sum_{d \in d^*} {}^{\lfloor d|\psi}\Theta = 1, \ \forall \psi \in \psi^* \right\}$ are unknown transition probabilities;

$\delta_{\cdot,\cdot}$ is the Kronecker symbol that equals 1 for identical arguments and it is zero otherwise.

10.1 Common tools

10.1.1 Dirichlet pdf as a conjugate prior

Markov chain factors belong to the exponential family, see Section 3.2, so that they possess conjugate prior. The following correspondence to (3.5) holds

$$\begin{aligned} f(d_t|d(t-1), \Theta) &= \exp\left[\sum_{[d,\psi']' = \Psi \in \Psi^*} \delta_{[d,\psi']',\Psi_t} \ln\left({}^{\lfloor d|\psi}\Theta \right) \right] \equiv A(\Theta) \exp\left[\langle B(\Psi_t), C(\Theta) \rangle \right] \text{ with} \\ A(\Theta) &\equiv 1, \ B_{\tilde{\Psi}}(\Psi) \equiv \delta_{\tilde{\Psi},\Psi}, \ C(\Theta_{\tilde{\Psi}}) = \ln\left({}^{\lfloor \tilde{d}|\tilde{\psi}}\Theta \right), \text{ with indices } \tilde{\Psi} \equiv \left[\tilde{d}, \tilde{\psi}' \right]' \in \Psi^* \\ \langle B, C \rangle &\equiv \sum_{\tilde{\Psi} \in \Psi^*} B_{\tilde{\Psi}}(\Psi) C_{\tilde{\Psi}}(\Theta). \end{aligned}$$

This correspondence determines the conjugate prior, see (3.12), in the form known as *Dirichlet pdf*

$$Di_{\Theta}(V) \equiv f(\Theta|V) = \frac{\prod_{\Psi = [d,\psi']' \in \Psi^*} \left[{}^{\lfloor d|\psi}\Theta \right]^{{}^{\lfloor d|\psi}V - 1} \chi_{\Theta^*}(\Theta)}{\mathcal{I}(V)}, \text{ where } \chi_{\Theta^*}(\Theta) \text{ is indicator of } \Theta^* \tag{10.2}$$

The value of the normalizing integral $\mathcal{I}(V)$ and restriction on the statistic $V = [{}^{\lfloor d|\psi}V]_{d \in d^*, \psi \in \psi^*}$ that guarantees finiteness of $\mathcal{I}(V)$ are described below together with other properties of this important pdf.

Within this chapter the indices at Θ and statistic V corresponding to the treated data vector $\Psi = [d', \psi']'$ are placed as the left upper index ${}^{\lfloor d|\psi}$. It makes iterative formulas with a lot of other indices a bit more readable.

Proposition 10.1.1 (Basic properties and moments of ***Di*** **pdf)** *The normalization integral of the Dirichlet pdf $Di_\Theta(V)$ is*

$$\begin{aligned} \mathcal{I}(V) &= \prod_{\psi\in\psi^*} \mathcal{B}\left(V_{\cdot|\psi}\right) \equiv \prod_{\psi\in\psi^*} \frac{\prod_{d\in d^*}\Gamma\left({}^{\lfloor d|\psi}V\right)}{\Gamma\left(\sum_{d\in d^*} {}^{\lfloor d|\psi}V\right)} \\ \Gamma(x) &= \int_0^\infty z^{x-1}\exp(-z)\,dz < \infty \text{ for } x > 0. \end{aligned} \tag{10.3}$$

Thus, the normalization factor is finite iff ${}^{\lfloor d|\psi}V > 0$, $\forall \Psi \equiv [d, \psi']' \in \Psi^*$ that is guaranteed for all posterior statistics ${}^{\lfloor d|\psi}V_t$ if ${}^{\lfloor d|\psi}V_0 > 0$.

Dirichlet pdf has the following marginal pdfs and moments

$$\begin{aligned}
& Di_{\,{}^{\lfloor d|\psi}\Theta}(V) \equiv f\left({}^{\lfloor d|\psi}\Theta|V\right) = \frac{\left[{}^{\lfloor d|\psi}\Theta\right]^{{}^{\lfloor d|\psi}V-1}\left[1-{}^{\lfloor d|\psi}\Theta\right]^{{}^{\lfloor\psi}\nu-{}^{\lfloor d|\psi}V-1}\chi_{[0,1]}\left({}^{\lfloor d|\psi}\Theta\right)}{\mathcal{I}\left({}^{\lfloor d|\psi}V\right)} \quad \textit{with} && (10.4)\\
& {}^{\lfloor\psi}\nu \equiv \sum_{d\in d^*} {}^{\lfloor d|\psi}V,\ \ \mathcal{I}\left({}^{\lfloor d|\psi}V\right) \equiv \frac{\Gamma\left({}^{\lfloor d|\psi}V\right)\Gamma\left({}^{\lfloor\psi}\nu - {}^{\lfloor d|\psi}V\right)}{\Gamma\left({}^{\lfloor\psi}\nu\right)} \\
& \mathcal{E}\left[{}^{\lfloor d|\psi}\Theta|V,\Psi\right] = \frac{{}^{\lfloor d|\psi}V}{{}^{\lfloor\psi}\nu} \\
& \mathcal{E}\left[{}^{\lfloor d|\psi}\Theta\ {}^{\lfloor\tilde{d}|\tilde{\psi}}\Theta|V,\Psi,\tilde{\Psi}\right] = (1-\delta_{\Psi,\tilde{\Psi}})\mathcal{E}\left[{}^{\lfloor d|\psi}\Theta|V,\Psi\right]\mathcal{E}\left[{}^{\lfloor\tilde{d}|\tilde{\psi}}\Theta|V,\tilde{\Psi}\right] + \\
& +\delta_{\Psi,\tilde{\Psi}}\{\mathcal{E}\left[{}^{\lfloor d|\psi}\Theta|V,\Psi\right]\}^2\frac{1+{}^{\lfloor d|\psi}V^{-1}}{1+{}^{\lfloor\psi}\nu^{-1}} \\
& \operatorname{cov}\left[{}^{\lfloor d|\psi}\Theta,\ {}^{\lfloor\tilde{d}|\tilde{\psi}}\Theta|V,\Psi,\tilde{\Psi}\right] = \delta_{\Psi,\tilde{\Psi}}\left\{\mathcal{E}[{}^{\lfloor d|\psi}\Theta|V,\Psi]\right\}^2\frac{{}^{\lfloor d|\psi}V^{-1}-{}^{\lfloor\psi}\nu^{-1}}{1+{}^{\lfloor\psi}\nu^{-1}} \\
& \mathcal{E}\left[\ln\left({}^{\lfloor d|\psi}\Theta\right)|V,\Psi\right] = \frac{\partial}{\partial\,{}^{\lfloor d|\psi}V}\ln\left(\Gamma\left({}^{\lfloor d|\psi}V\right)\right) - \frac{\partial}{\partial\,{}^{\lfloor\psi}\nu}\ln\left(\Gamma\left({}^{\lfloor\psi}\nu\right)\right). && (10.5)\\
& \textit{The predictive pdf is} f(d|\psi,V) = \frac{{}^{\lfloor d|\psi}V}{{}^{\lfloor\psi}\nu} \equiv \mathcal{E}\left[{}^{\lfloor d|\psi}\Theta|V,\Psi\right]. && (10.6)
\end{aligned}$$

Proof: Evaluation of the normalization integral is the basic step. It can be done inductively starting with $\mathring{d} = 2$, using definition of Euler beta function and its relationship to gamma function [81]. The rest relies mostly on the known recursion $\Gamma(x+1) = x\Gamma(x)$.

The evaluation $\mathcal{E}\left[\ln\left({}^{\lfloor d|\psi}\Theta\right)|V,\Psi\right]$ is the only involved step. To demonstrate it, let us assume that $\Theta \sim Di_\Theta(m,n) = \mathcal{B}^{-1}(m,n)\Theta^{m-1}(1-\Theta)^{n-1}$. Then,

$$\begin{aligned} \mathcal{E}[\ln(\Theta)] &= \mathcal{B}^{-1}(m,n)\int_0^1 \ln(\Theta)\Theta^{m-1}(1-\Theta)^{n-1}\,d\Theta = \mathcal{B}^{-1}(m,n)\int_0^1 \frac{\partial}{\partial m}\Theta^{m-1}(1-\Theta)^{n-1}\,d\Theta = \\ &= \mathcal{B}^{-1}(m,n)\frac{\partial}{\partial m}\mathcal{B}(m,n) = \frac{\partial}{\partial m}\ln(\mathcal{B}(m,n)) = \frac{\partial}{\partial m}\ln(\Gamma(m)) - \frac{\partial}{\partial(m+n)}\ln(\Gamma(m+n)). \end{aligned}$$

Substitution $m = {}^{\lfloor d|\psi}V$ and $m + n = {}^{\lfloor\psi}\nu$ gives the result to be proved. □

Repeatedly, we need to evaluate the KL distance of a pair of Dirichlet pdfs. Parameters corresponding to different regression vectors are independent. Consequently, we can consider a fixed, and for notation simplicity, empty regression vector.

Proposition 10.1.2 (KL distance of ***Di*** **pdfs)** *Let $f(\Theta) = Di_\Theta(V)$, $\tilde{f}(\Theta) = Di_\Theta\left(\tilde{V}\right)$ be a pair of Dirichlet pdfs of parameters $\Theta \equiv \left({}^{\lfloor 1}\Theta, \ldots, {}^{\lfloor\mathring{d}}\Theta\right) \in \Theta^* = \left\{{}^{\lfloor i}\Theta > 0,\ \sum_{i\in i^*} {}^{\lfloor i}\Theta = 1\right\}$, $i^* \equiv \left\{1, \ldots, \mathring{d}\right\}$.*

Their KL distance is given by the formula

$$\begin{aligned}\mathcal{D}(f||\tilde{f}) &= \sum_{d=1}^{\mathring{d}}\left[\left({}^{\lfloor d}V - {}^{\lfloor d}\tilde{V}\right)\frac{\partial}{\partial\, {}^{\lfloor d}V}\ln\left(\Gamma\left({}^{\lfloor d}V\right)\right) + \ln\left(\frac{\Gamma\left({}^{\lfloor d}\tilde{V}\right)}{\Gamma\left({}^{\lfloor d}V\right)}\right)\right] - (\nu-\tilde{\nu})\frac{\partial}{\partial\nu}\ln(\Gamma(\nu)) + \\ &\quad + \ln\left(\frac{\Gamma(\nu)}{\Gamma(\tilde{\nu})}\right)\\ \nu &\equiv \sum_{d=1}^{\mathring{d}} {}^{\lfloor d}V,\quad \tilde{\nu} \equiv \sum_{d=1}^{\mathring{d}} {}^{\lfloor d}\tilde{V}. \end{aligned} \tag{10.7}$$

Proof: Majority of evaluations is prepared by Proposition 10.1.1. For the considered pdfs, the KL distance (2.26) gets the form (use (10.3), (10.4)) $\mathcal{D} =$

$$\begin{aligned}&= \mathcal{E}\left[\sum_{d=1}^{\mathring{d}}({}^{\lfloor d}V - {}^{\lfloor d}\tilde{V})\ln\left({}^{\lfloor d}\Theta\right)|V\right] - \ln\left(\frac{\mathcal{B}(V)}{\mathcal{B}(\tilde{V})}\right) = \sum_{d=1}^{\mathring{d}}({}^{\lfloor d}V - {}^{\lfloor d}\tilde{V})\mathcal{E}\left[\ln\left({}^{\lfloor d}\Theta\right)|\,{}^{\lfloor d}V\right] - \ln\left(\frac{\mathcal{B}(V)}{\mathcal{B}(\tilde{V})}\right) = \\ &\underbrace{=}_{(10.5)} \sum_{d=1}^{\mathring{d}}\left[\left({}^{\lfloor d}V - {}^{\lfloor d}\tilde{V}\right)\left(\frac{\partial}{\partial\,{}^{\lfloor d}V}\ln\left(\Gamma\left({}^{\lfloor d}V\right)\right) - \frac{\partial}{\partial\nu}\ln(\Gamma(\nu))\right)\right] - \ln\left(\frac{\mathcal{B}(V)}{\mathcal{B}\left(\tilde{V}\right)}\right) = \\ &= \sum_{d=1}^{\mathring{d}}\left({}^{\lfloor d}V - {}^{\lfloor d}\tilde{V}\right)\frac{\partial}{\partial\,{}^{\lfloor d}V}\ln\left(\Gamma\left({}^{\lfloor d}V\right)\right) - (\nu-\tilde{\nu})\frac{\partial}{\partial\nu}\ln(\Gamma(\nu)) - \ln\left(\frac{\mathcal{B}(V)}{\mathcal{B}(\tilde{V})}\right) = \\ &= \sum_{d=1}^{\mathring{d}}\left[\left({}^{\lfloor d}V - {}^{\lfloor d}\tilde{V}\right)\frac{\partial}{\partial\,{}^{\lfloor d}V}\ln\left(\Gamma\left({}^{\lfloor d}V\right)\right) + \ln\left(\frac{\Gamma\left({}^{\lfloor d}\tilde{V}\right)}{\Gamma\left({}^{\lfloor d}V\right)}\right)\right] - (\nu-\tilde{\nu})\frac{\partial}{\partial\nu}\ln(\Gamma(\nu)) + \ln\left(\frac{\Gamma(\nu)}{\Gamma(\tilde{\nu})}\right).\end{aligned}$$

There, we have used linearity of expectation, the definitions of the beta function $\mathcal{B}$ and of the statistic $\nu \equiv \sum_{d\in d^*} {}^{\lfloor d}V$, $\tilde{\nu} \equiv \sum_{d\in d^*} {}^{\lfloor d}\tilde{V}$. □

10.1.2 Estimation and prediction with Markov chain factors

As the Markov chain factors predicting discrete-valued d_t belong to the exponential family, their estimation and subsequent prediction with the conjugate prior pdf reduce to algebraic operations, see Proposition 3.2.1.

Proposition 10.1.3 (Estimation and prediction of the Markov chain factor) *Let us consider a fixed (discrete) value ψ of the regression vector. Let natural conditions of decision making, Requirement 2.5.1, hold, the treated factor (10.1) is Markov chain and a Dirichlet conjugate prior $Di_\Theta(V_0)$ (3.12), $V_0 \equiv \left[{}^{\lfloor 1|\psi}V_0,\ldots,{}^{\lfloor \mathring{d}|\psi}V_0\right]$ as well as conjugate alternatives $Di_\Theta({}^{\lfloor A}V_t)$, ${}^{\lfloor A}V_t \equiv \left[{}^{\lfloor 1A|\psi}V_t,\ldots,{}^{\lfloor \mathring{d}A|\psi}V_t\right]$, see Section 3.1, in stabilized forgetting, with the forgetting factor $\lambda \in [0,1]$, are used.*

Then, the posterior pdf is also Dirichlet pdf $Di_\Theta(V_t)$, $V_t \equiv \left[{}^{\lfloor 1|\psi}V_t,\ldots,{}^{\lfloor \mathring{d}|\psi}V_t\right]$, and its sufficient statistic evolves according to the recursion

$${}^{\lfloor d|\psi}V_t = \lambda\left({}^{\lfloor d|\psi}V_{t-1} + \delta_{\Psi,\Psi_t}\right) + (1-\lambda)\,{}^{\lfloor dA|\psi}V_t,\ \ {}^{\lfloor d|\psi}V_0 \text{ given, } [d,\psi']' \equiv \Psi \in \Psi^*. \tag{10.8}$$

The predictive pdf is

$$f(d|\psi,V_t) = \frac{{}^{\lfloor d|\psi}V_t}{\sum_{\tilde{d}\in d^*} {}^{\lfloor \tilde{d}|\psi}V_t}. \tag{10.9}$$

Proof: It is a direct consequence of the Bayes rule applied to the member of the exponential family. □

Remark(s) 10.1.1

1. *The array* $V = \left[{}^{\lfloor d|\psi}V \right]_{d\in d^*, \psi\in\psi^*}$ *is also known as the* occurrence table.
2. *The obtained prediction corresponds with understanding of the probability as a relative frequency of events. Here, the data are just split in subsets corresponding to different values of regression vectors* ψ. *Numerically, the Bayesian set up adds non-trivial initial conditions to the occurrence table.*
3. *The predictive pdf is the key element needed for approximate estimation, see Section 10.5.*

Problem 10.1.1 (Reduction of dimensionality) *Dimensionality of the occurrence table might be easily excessive, mostly because of the high cardinality* $\mathring{\psi}$. *Then, a reduced parameterization is necessary. Promising techniques exploiting a sparse nature of this table are described in [77, 91]. This could be reasonable start for solving this practically important problem.*

Problem 10.1.2 (Relationships to non-parametric Bayesian estimation) *Markov chains have a tight connection to* non-parametric Bayesian estimation, *[92]. Inspection whether this connection can be used in our context is worth of considering. It could extend applicability of the achieved results significantly.*

10.1.3 Likelihood on variants

Predictors based on Markov chains serve for selecting among competitive descriptions of the same discrete valued data. Proposition 10.1.1 gives immediately Markov-chain variant of Proposition 6.1.1.

Proposition 10.1.4 (Markov-chain mixture-based *one-step-ahead predictor*) *Use of on line estimation of a Markov chain mixture within the framework of the* adaptive advisory system *provides one-step-ahead predictor in the form*

$$
\begin{aligned}
f(d_{t+1}|d(t)) &= \sum_{c\in c^*}\prod_{i\in i^*} f(d_{ic;t+1}|d(t),c)\frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}} \equiv \sum_{c\in c^*}\prod_{i\in i^*}\frac{\mathcal{I}(d(t+1)|ic)}{\mathcal{I}(d(t)|ic)}\frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}} = \\
&= \sum_{c\in c^*}\hat{\alpha}_{c;t}\prod_{i\in i^*} f(d_{ic;t+1}|\psi_{ic;t+1},c), \quad \textit{where} \\
\hat{\alpha}_{c;t} &\equiv \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}} \qquad (10.10)\\
\mathcal{I}(d(t)|ic) &\equiv \mathcal{I}(V_{ic;t}) = \mathcal{B}(V_{ic;t}) \equiv \frac{\prod_{d_{ic}\in d_{ic}^*}\Gamma\left({}^{\lfloor d_{ic}|\psi_{ic}}V_{ic;t}\right)}{\Gamma\left(\sum_{d_{ic}\in d_{ic}^*}{}^{\lfloor d_{ic}|\psi_{ic}}V_{ic;t}\right)}, \\
\Gamma(x) &= \int_0^\infty z^{x-1}\exp(-z)\,dz < \infty \text{ for } x>0, \ \textit{where} \\
{}^{\lfloor d_{ic}|\psi_{ic}}V_{ic;t} &= \textit{the occurrence table of the factor } ic \textit{ collected up to and including time } t.
\end{aligned}
$$

The factor predictors can also be expressed as follows

$$
\begin{aligned}
f(d_{ic;t+1}|\psi_{ic;t+1},c) &= \frac{{}^{\lfloor d_{ic;t+1}|\psi_{ic;t+1}}V_{ic;t}}{\sum_{d_{ic}\in d_{ic}^*}{}^{\lfloor d_{ic}|\psi_{ic;t+1}}V_{ic;t}}, \textit{ for an adaptive advisory system, approximated by} \\
f(d_{ic;t+1}|\psi_{ic;t+1},c) &= \frac{{}^{\lfloor d_{ic;t+1}|\psi_{ic;t+1}}V_{ic;0}}{\sum_{d_{ic}\in d_{ic}^*}{}^{\lfloor d_{ic}|\psi_{ic;t+1}}V_{ic;0}}, \textit{ for a fixed advisory system.}
\end{aligned}
$$

Proof: Omitted. □

10.1.4 Branch and bound techniques

For Markov chain parameterized factors with conjugate prior pdfs, various $Di_\Theta(V)$ pdfs are alternative descriptions inspected and compared. Thus, alternative sets of sufficient statistics V are branched and/or bounded as outlined in Section 6.1.3. The maximized functional F then becomes the function, namely, the v-likelihood $f(d(\mathring{t})|V)$ of data $d(\mathring{t})$ conditioned on the sufficient statistics V determining the $Di_{\Theta}(V)$ pdf that are used for eliminating unknown parameters

$$f(d(\mathring{t})|V) = \int f(d(\mathring{t})|\Theta) Di_{\Theta}(V)\, d\Theta \underbrace{=}_{(10.10)} \prod_{t \in t^*} f(d_t|d(t-1)).$$

Evaluation of approximate v-likelihood values for Markov chain mixtures is implied by Proposition 10.1.4. Specific versions of branch and bound techniques are described in Section 10.4.

10.2 Data preprocessing

Except of handling of missing data, no general preprocessing is expected when the modelled data are "naturally" discrete-valued. Often, however, the discrete valued data arise by sampling underlying continuous valued signals. Then, full preprocessing arsenal described in Chapter 6 is needed. Here, the aspects specific to the intended discretization are briefly discussed.

10.2.1 Use of physical boundaries

Ability to respect hard physical bounds is a key advantage of Markov chain models applied to continuous-valued signals. When sampling a signal within a given range, it should be taken into account that the uneven occurrence of the underlying scalar signal values within the inspected range is more rule than exception. This calls for an uneven specification of discretization levels. The following simple Proposition gives the appropriate guidelines.

Proposition 10.2.1 (Signal discretization) *Let $f(x) > 0$ be pdf of a scalar continuous-valued variable $x \in x^* \equiv [\underline{x}, \overline{x}]$ with known finite boundaries $\underline{x}$, $\overline{x}$. Let us consider a grid x_i, $i = 0, \ldots, \mathring{i}$,*

$$\underline{x} = x_0 < x_1 < \ldots x_{\mathring{i}-1} < x_{\mathring{i}} = \overline{x}$$

with a given finite number of grid points $\mathring{i} + 1$.

Let $\hat{f} \in \hat{f}^* \equiv \{$ pdfs on x^* with constant values $\hat{f}_i$ on intervals (x_{i-1}, x_i) $\}$. The best approximation $\hat{f}(x)$ of the pdf $f(x)$ taken from $\hat{f}^*$ such that

$$\mathcal{D}\left(f||\hat{f}\right) \to \min_{\{\hat{f}_i\}_{i \in i^*}} \quad \text{and} \quad \mathcal{D}\left(\hat{f}||\frac{\mathbf{1}}{\mathring{i}}\right) \to \min_{\{x_i\}_{i \in i^*}}$$

is the uniform one on the grid specified by the requirement

$$\frac{1}{\mathring{i}} = \int_{x_{i-1}}^{x_i} f(x)\, dx. \tag{10.11}$$

Proof: First a fixed grid is considered, then the part of $\mathcal{D}\left(f||\hat{f}\right)$ influenced by values $\hat{f}_i$ has the form

$$\sum_{i=1}^{\mathring{i}} \int_{x_{i-1}}^{x_i} f(x) \ln\left(\frac{1}{\hat{f}_i}\right) dx = \sum_{i=1}^{\mathring{i}} \left(\int_{x_{i-1}}^{x_i} f(x)\, dx\right) \ln\left(\frac{1}{\hat{f}_i}\right) =$$

$$= \sum_{i=1}^{\mathring{i}} \left(\int_{x_{i-1}}^{x_i} f(x)\, dx\right) \ln\left(\frac{\int_{x_{i-1}}^{x_i} f(x)\, dx}{\hat{f}_i}\right) - \sum_{i=1}^{\mathring{i}} \left(\int_{x_{i-1}}^{x_i} f(x)\, dx\right) \ln\left(\int_{x_{i-1}}^{x_i} f(x)\, dx\right).$$

In this expression, the last term is independent of the optimized values and the previous term is the KL distance minimized by $\hat{f}_i = \int_{x_{i-1}}^{x_i} f(x)\, dx$.

The second KL distance measures uncertainty of the resulting approximation and it reaches the smallest zero value for uniform pdf $\hat{f}$. □

This simple Proposition hints how to discretize scalar signal $d(\mathring{t})$.

Algorithm 10.2.1 (Signal discretization)

1. *Select the number* $\mathring{i}$ *of discretization levels.*
2. *Fit a rich normal static mixture to the learning data* $d(\mathring{t})$.
3. *Evaluate corresponding distribution function as the integral of the pdf estimated in the previous step.*
4. *Split the range [0,1] of values of the distribution function into* $\mathring{i}$*-intervals of the same length.*
5. *Project these values on the axis of arguments of the distribution function. It gives the grid searched for.*

Remark(s) 10.2.1

1. *Any mixture can be fitted to experimental data for the discussed purpose. Even ordinary histogram can be used. Mixture estimation serves here as a smoothed version of histogram.*
2. *Dependencies are neglected in the construction. Essentially, a sort of ergodic behavior is implicitly assumed and the estimated pdf is the limiting distribution. This approach seems to be sufficient for the considered purpose.*
3. *Proposition 10.2.1 can be and should be applied to increments of signal, too. For smooth signals, that are expected in physical world, probability that the signal change is greater than some level is practically zero. This fact is revealed by the described algorithm that "discovers" a band of non-zero transition probabilities. It helps to decrease significantly the number of estimated parameters and makes treatment of such signals practicable.*

Problem 10.2.1 (Algorithmic solution of discretization) *The basic ideas presented above have to be converted into a complete set of algorithms.*

10.2.2 Removal of high frequency noise

Discretization of signal values suppresses small (within the discretization step) changes in signal values. High frequency noise has usually relatively small amplitudes. Consequently, discretization suppresses "naturally" high frequency noise.

This brief discussion describes more tendency than a general rule. Thus, a high-frequency-noise removal should be considered before discretization.

10.2.3 Suppression of outliers

In the discussed context, outliers are dangerous as they extend the range of the signal to be discretized. They increase the number of levels to be considered for achieving the target precision. Proposition 10.2.1 helps in this respect, too. By definition, outliers occur rarely so that their, even wide, range can be labelled by a few discrete values only.

10.2.4 Coding of regression vectors

Both predicted data d item and regression vector $\tilde{\psi}$ are discrete valued. The sufficient statistics V describing individual factors are arrays indexed by data vectors $\Psi = \left[d, \tilde{\psi}'\right]'$. For an easy software manipulation, it is reasonable to keep d as an independent index and to map the regression vector $\tilde{\psi}$ to a scalar variable ψ. The chosen mapping, called *coding*, influences the formal structure of the statistics $V = \left[{}^{\lfloor d|\tilde{\psi}}V\right] = \left[{}^{\lfloor d|\psi}V\right]$, $d \in d^* \equiv \{1, \ldots, \mathring{d}\}$, $\psi \in \psi^* \equiv \{1, \ldots, \mathring{\psi}\}$. Treatment of the potentially sparse matrix $[V_{d|\psi}]$ is simpler if its non-trivial entries are clustered in a few contiguous areas. Coding may either enhance or destroy this property. It seems that delayed items of a single variable, say $\tilde{d}_t, \ldots, \tilde{d}_{t-\partial}$, with cardinality $\mathring{d}$, should be coded to scalar d_t by the "thermometer" code

$$d_t \equiv \sum_{i=0}^{\partial} \mathring{d}^{\partial-i} \tilde{d}_{t-i} = \mathring{d}^{\partial} \tilde{d}_t + \mathring{d} d_{t-1} - \tilde{d}_{t-\partial-l}. \tag{10.12}$$

This coding reflects presumption that the older values have smaller influence on the predicted value. Universality of this option and combinations of regressors of a different nature are questionable and problem should be studied in detail. Vast amount of experience available in signal processing should definitely be exploited.

Description of graph-type dependencies is another aspect of the same problem. Experience from the field known as Bayesian networks [93] could help.

10.2.5 Coding of signal values

Mixture consisting of Markov-chains components only is just a Markov chain. It stimulates natural question whether it makes sense to use them at all. The affirmative answer follows from the possibility to spare parameters to be estimated by introducing a special structure into mixture components. The discussed coding of signal values serves as an example having its own merit.

Let us have a continuous-valued signal x_t whose evolution is described by the pdf $f(x_t|x(t-1)) \equiv f(x_t|x_{t-1})$ with a finite support covered say by $[0,1]^2$. Binary expansion of $x_t \approx \sum_{i=1}^{\mathring{d}} d_{i;t} 2^{-i}$ makes x_t (approximately) equivalent to discrete-valued multivariate data record d_t with entries $d_{i;t} \in \{0,1\}$. The data d_t forms the Markov chain that can be approximated by a specific mixture. For describing it, it is sufficient to consider $\mathring{d} = 2$. Then, the following approximation

$$\begin{aligned} f(d_t|d_{t-1}) &= f(d_{1;t}|d_{2;t}, d_{1;t-1}, d_{2;t-1}) f(d_{2;t}|d_{1;t-1}, d_{2;t-1}) \approx \\ &\approx f(d_{1;t}|d_{1;t-1}) \left[\alpha f(d_{2;t}|d_{2;t-1}, d_{1;t-1} = 0) + (1-\alpha) f(d_{2;t}|d_{2;t-1}, d_{1;t-1} = 1)\right] \equiv \\ &\equiv f(d_{1;t}|d_{1;t-1}) \left[\alpha f(d_{2;t}|d_{2;t-1}, c = 1) + (1-\alpha) f(d_{2;t}|d_{2;t-1}, c = 2)\right] \end{aligned}$$

can be justified by a dominant influence of $d_{1;t-1}$ on $d_{1;t}$ and by introducing "averaged" influence of $d_{1;t-1}$ on the evolution of $d_{2;t-1} \to d_{2;t}$. The original Markov chain on the left-hand side has 12 free parameters, the approximate right-hand side has 7 free parameters, including the probability α quantifying the average influence of $d_{1;t-1}$. This number increases to 9 if we do not assume that $f(d_{1;t}|d_{1;t-1})$ is a common factor. The difference is much more significant for higher $\mathring{d}$.

10.3 Use of prior knowledge at factor level

Here, general ideas of quantification of prior knowledge at the factor level, Section 6.3, are specialized to Markov chain factors.

10.3.1 Internally consistent fictitious data blocks

Processing of internally consistent data blocks coincides with learning Markov chain factors, see Section 10.1.2. Thus, we have to quantify individual data items and then to specialize merging of individual pdfs $f(\Theta|K_k)$ expressing knowledge items $K_k,\ k = 1, \ldots, \mathring{k}$.

10.3.2 Translation of input-output characteristics into data

Quantification of the knowledge of the initial moments (6.28) of the predictive pdf $f(d|\psi)$ given by a fixed regression vector ψ is assumed

$$\hat{d} = \sum_{d \in d^*} d f(d|\psi),\ \hat{r} = \sum_{d \in d^*} \left(d - \hat{d}\right)^2 f(d|\psi). \tag{10.13}$$

The general Proposition 6.3.1 gives the constructed prior out of the conjugate class Dirichlet pdfs. We need to stay within it. It makes us to derive the following specialized proposition.

Proposition 10.3.1 (Knowledge of input-output characteristics for Markov chain factors) *Let us fix a regression vector ψ. Consider Dirichlet pdfs $Di_\theta(V)$ pdfs of the unknown parameter θ with entries $\theta_d \equiv {}^{\lfloor d|\psi}\Theta$ and determined by the vector statistics V with entries ${}^{\lfloor d}V \equiv {}^{\lfloor d|\psi}V$, $d \in d^*$. The symbol ${}^{\lfloor d|\psi}V$ denotes entry of a* positive *occurrence array, see Section 10.1.1.*

We search for the $Di_\theta(V)$ pdf that fulfills restrictions (10.13) and minimizes the KL distance to the flat pre-prior $Di_\theta(\varepsilon \mathbf{1})$ given by $\varepsilon > 0$, $\varepsilon \to 0$ that multiplies vector of units $\mathbf{1} = [1, \ldots, 1]'$ *of the appropriate length.*

Such a pdf is determined by the statistics

$$ {}^{\lfloor d}V = \frac{1}{ad^2 + bd + c}, \; d \in d^* = \{1, \ldots, \mathring{d}\} \tag{10.14}$$

with constants a, b, c solving equations

$$\hat{d} = \frac{\sum_{d \in d^*} \frac{d}{ad^2+bd+c}}{\sum_{d \in d^*} \frac{1}{ad^2+bd+c}}, \; \hat{r} = \frac{\sum_{d \in d^*} \frac{\left(d-\hat{d}\right)^2}{ad^2+bd+c}}{\sum_{d \in d^*} \frac{1}{ad^2+bd+c}}, \tag{10.15}$$

while minimizing the resulting distance.

Proof: For $\nu \equiv \sum_{d \in d^*} {}^{\lfloor d}V$, $\bar{\nu} = \varepsilon \mathring{d}$, the optimized functional $\mathcal{D}\left(f||\bar{f}\right)$ is the function of statistics V, $\bar{V} = \varepsilon \mathbf{1}$, see Proposition 10.1.2,

$$\mathcal{D}\left(f||\bar{f}\right) = \sum_{d=1}^{\mathring{d}} \left[\left({}^{\lfloor d}V - {}^{\lfloor d}\bar{V} \right) \frac{\partial}{\partial\, {}^{\lfloor d}V} \ln\left(\Gamma\left({}^{\lfloor d}V \right)\right) + \ln\left(\frac{\Gamma\left({}^{\lfloor d}\bar{V}\right)}{\Gamma\left({}^{\lfloor d}V\right)} \right) \right] - (\nu - \bar{\nu}) \frac{\partial}{\partial \nu} \ln(\Gamma(\nu)) + \ln\left(\frac{\Gamma(\nu)}{\Gamma(\bar{\nu})} \right)$$

Its derivative with respect to ${}^{\lfloor d}V$ is

$$\frac{\partial \mathcal{D}\left(f||\bar{f}\right)}{\partial\, {}^{\lfloor d}V} = \frac{\partial \ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)}{\partial\, {}^{\lfloor d}V} + \left({}^{\lfloor d}V - {}^{\lfloor d}\bar{V} \right) \frac{\partial^2 \ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)}{\partial\, {}^{\lfloor d}V^2} - \frac{\partial \ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)}{\partial\, {}^{\lfloor d}V} - \frac{\partial \ln(\Gamma(\nu))}{\partial \nu} +$$
$$+ \frac{\partial \ln(\Gamma(\nu))}{\partial \nu} - (\nu - \bar{\nu}) \frac{\partial^2 \ln(\Gamma(\nu))}{\partial \nu^2} = \left({}^{\lfloor d}V - {}^{\lfloor d}\bar{V} \right) \frac{\partial^2 \ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)}{\partial\, {}^{\lfloor d}V^2} - (\nu - \bar{\nu}) \frac{\partial^2 \ln(\Gamma(\nu))}{\partial \nu^2}.$$

Similarly, $\dfrac{\partial \hat{d}}{\partial\, {}^{\lfloor d}V} = \nu^{-1}(d - \hat{d})$, $\dfrac{\partial \hat{r}}{\partial\, {}^{\lfloor d}V} = \nu^{-1} \left[\left(d - \hat{d}\right)^2 - \hat{r} \right]$.

This gives derivatives of Lagrangian function that, for $\varepsilon \to 0$ can be re-written into the form

$$0 = {}^{\lfloor d}V \frac{\partial^2 \ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)}{\partial V_d^2} + 0.5ad^2 + 0.5bd + 0.5(c-1), \text{ where}$$

the constants a, b, c are made of Lagrangian multipliers and terms independent of d. Using expansion of $\ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)$ [81], the involved second derivative can be approximated by

$$\frac{\partial^2 \ln\left(\Gamma\left({}^{\lfloor d}V\right)\right)}{\partial V_d^2} \approx 0.5 V_d^{-1}(1 - V_d^{-1}).$$

Putting this approximation into the condition for extreme, we get the linear equation for ${}^{\lfloor d}V$ that determines its dependence on d. The optional constants a, b, c are selected so that restrictions are fulfilled. The selection is made unique by selecting c so that the resulting distance is the smallest one. □

Remark(s) 10.3.1

1. *For a given a, b, the value c determines the sum $\nu = \sum_{d \in d^*} (ad^2 + bd + c)^{-1}$. The minimized its increasing function while $\nu > \bar{\nu}$. It hints the choice of c as the solution of the equation $\bar{\nu} = \sum_{d \in d^*} (ad^2 + bd + c)^{-1}$. This statement is not included into Proposition as it is not proved that such a (a, b, c) exist while guaranteeing that ${}^{\lfloor d}V > 0$. A non-trivial solution can be expected iff meaningful options $\hat{d} \in \left(1, \mathring{d}\right)$ and $\hat{r} < \left(0.5\mathring{d}\right)^2$ are made.*
2. *Other restrictions could be and should be considered, respecting the discrete nature of data.*

10.3.3 Merging of knowledge pieces

We have got a collection of pdfs $f(\Theta|K_k) = Di_\Theta(V_k)$ $k \in k^* \equiv \{1,\ldots,\mathring{k}\}$ after processing individual internally consistent data blocks and individual knowledge items. We use them for constructing of a single posterior pdf $\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k})) = Di_\Theta(V_{\hat{t}})$ that merge them all together with available real data $d(\mathring{t})$. For this, the general Proposition 6.3.2 applies. It leads directly to the Di_θ counterpart of Algorithm 6.3.2.

Algorithm 10.3.1 (Merging Dirichlet knowledge sources)

1. *Construct prior $Di_\Theta(V_k)$ pdfs reflecting internally consistent data blocks by applying ordinary Bayesian estimation and/or reflecting individual knowledge pieces according to Proposition 10.3.1.*
2. *Evaluate the likelihood function $\mathcal{L}(\Theta, d(\mathring{t})) = Di_\Theta(V_{0;\mathring{t}})$ by applying Proposition 10.1.3 with zero initial conditions for V.*
3. *Evaluate posterior pdfs $f(\Theta|d(t), K_k) = Di_\Theta(V_{0;\mathring{t}} + V_k)$ and v-likelihood*

$$f(d(\mathring{t})|K_k) = \frac{\mathcal{I}(V_{0;\mathring{t}} + V_k)}{\mathcal{I}(V_k)}$$

   corresponding to prior pdfs $f(\Theta|K_k) = Di_\Theta(V_k)$ $k \in k^*$. The integral $\mathcal{I}(V)$ is given by the formula (10.3).

   Regularization by a flat pre-prior pdf $Di_\Theta(\varepsilon\mathbf{1})$, given by small $\varepsilon > 0$ has to be considered if some V_k contains zero values.

4. *Evaluate weights*

$$\beta_{k|d(\mathring{t})} = \frac{f(d(\mathring{t})|K_k)}{\sum_{\tilde{k}\in k^*} f(d(\mathring{t})|K_{\tilde{k}})} = \frac{\frac{\mathcal{I}(V_{0;\mathring{t}}+V_k)}{\mathcal{I}(V_k)}}{\sum_{\tilde{k}\in k^*} \frac{\mathcal{I}(V_{0;\mathring{t}}+V_{\tilde{k}})}{\mathcal{I}(V_{\tilde{k}})}},\ k \in k^*.$$

5. *Determine the merger as the posterior pdf to be used*

$$\hat{f}(\Theta|d(\mathring{t}), K(\mathring{k})) \quad = \quad Di_\Theta\left(V_{0;\mathring{t}} + \sum_{k\in k^*} \beta_{k|d(\mathring{t})} V_k\right) = Di_\Theta(V_{0;\mathring{t}}) Di_\Theta\left(\sum_{k\in k^*} \beta_{k|d(\mathring{t})} V_k\right).$$

   Notice that the used "prior" pdf is seen in this formula.

10.4 Construction of the prior estimate

Here, Markov chain counterparts of Section 6.4 are presented.

10.4.1 Iterative construction of prior pdf

The danger of a plain repetitive use of the Bayes rule, discussed at general level in Section 6.4.1, gets the following form within the conjugate class of Dirichlet pdfs. After $\mathring{n}$-th application of the (approximate) Bayes rule we get the posterior pdf

$$Di_\Theta\left(\sum_{n=1}^{\mathring{n}} V_{n;\mathring{t}}\right), \text{ where } V_n \text{ is the sufficient statistics evaluated at } n\text{-th step.}$$

Taking into account basic properties of the Dirichlet pdf, Proposition 10.1.1, it can be seen that point estimates $\hat{\Theta}$ of unknown Θ would be around point estimate of non-weighted average of point estimates obtained in respective iterations. Their apparent uncertainty would be too small as it is inversely proportional to $\mathring{n}\mathring{t}$. This observation clarifies why the flattening, see Section 6.4.3 may help. It acts as forgetting that prefers newer, hopefully better, values of $V_{n;\mathring{t}}$ and prevents the estimated precision to grow too quickly.

10.4.2 Common bounding mapping

The common bounding mapping, see Section 6.4.2, just cancels the "least fit candidates". For Markov chain mixtures, the posterior pdfs are described by the vector statistic κ describing component weights and by the collection of the occurrence matrices V_{ic}. These are finite-dimensional objects. They may contain, however, a large amount entries. This fact makes us to search for a compromise between two contradictory requirements on the bounding mapping

- preserve as much information as possible from previous iteration steps,
- store as little as possible alternatives.

As said in previous chapters, we have no general guideline how to reach the adequate compromise. At present, we stress the second item and bound the number of alternatives to two, at most to three.

10.4.3 Flattening mapping

We specialize flattening operation to Markov chain mixtures.

Proposition 10.4.1 (Optimal flattening mapping for Di factors) *Let $\tilde{f} = Di_{\Theta}\left(\tilde{V}\right)$ and $\bar{f} = Di_{\Theta}\left(\bar{V}\right)$ be a pair of Di pdfs defined on the common support Θ^*. Then, the pdf $\hat{f}$ defined on Θ^* and minimizing the functional $\mathcal{D}\left(\hat{f}||\tilde{f}\right) + q\mathcal{D}\left(\hat{f}||\bar{f}\right)$, $q > 0$ is $Di_{\Theta}\left(\hat{V}\right)$ with*

$$\hat{V} = \Lambda\tilde{V} + (1-\Lambda)\bar{V} \text{ with } \Lambda \equiv 1/(1+q) \in (0,1). \tag{10.16}$$

Proof: Omitted. □

The application of this result to whole mixtures is straightforward as the posterior pdf of its parameters is product of the posterior pdfs related to weights α and individual factors.

Proposition 10.4.2 (Optimal flattening mapping for Markov chain mixtures) *Let parameters of a pair of Markov chain mixtures be described by factors $\tilde{f}_{ic}(\Theta_{ic}) = Di_{\Theta_{ic}}\left(\tilde{V}_{ic}\right)$, $\tilde{f}(\alpha) = Di_{\alpha}(\tilde{\kappa})$, $i \in i^*, c \in c^*$ and $\bar{f}_{ic}(\Theta_{ic}) = Di_{\Theta_{ic}}\left(\bar{V}_{ic}\right)$, $\bar{f}(\alpha) = Di_{\alpha}(\bar{\kappa})$, $i \in i^*, c \in c^*$. The factors with the same indices ic are defined on the common support Θ^*_{ic}. Then, the pdf $\hat{f}$ minimizing the functional*

$$\mathcal{D}\left(\hat{f}||\tilde{f}\right) + q\mathcal{D}\left(\hat{f}||\bar{f}\right),\ q > 0$$

preserves functional form of combined pdfs and it is given by the statistics, $i \in \{1, \ldots, \mathring{d}\}$, $c \in c^*$,

$$\hat{V}_{ic} = \Lambda\tilde{V}_{ic} + (1-\Lambda)\bar{V}_{ic},\ \hat{\kappa}_c = \Lambda\tilde{\kappa}_c + (1-\Lambda)\bar{\kappa}_c \text{ with } \Lambda \equiv 1/(1+q) \in (0,1). \tag{10.17}$$

Proof: Omitted. □

Flattening of the component-weights estimates is universal. It preserves the Dirichlet form with statistics being a convex combination of statistics of estimates among which the compromise is searched for. The same applies to estimates of Markov chain factors as they are of Dirichlet type, too. The choice of the optional flattening factor is fully described by Algorithms 6.4.2, 6.4.3 and Propositions 6.4.4, 6.4.6.

10.4.4 Geometric mean as branching mapping

Geometric-mean branching mapping $\mathcal{A}$ (6.17) maps a pair of arguments $\hat{f}_n(\Theta)$, $n = 1, 2$, of the maximized functional (6.40) $^{\lfloor h}\mathcal{L}\left(f_n(d(\mathring{t})|\Theta)\hat{f}_n(\Theta), d(\mathring{t})\right) \equiv \hat{f}_n(d(\mathring{t})) \equiv \int f_n(d(\mathring{t})|\Theta)\hat{f}_n(\Theta)\, d\Theta$ on a new, hopefully better, candidate $\hat{f}_n(\Theta)$.

Recall that $f_n(d(\mathring{t})|\Theta) \propto Di_{\Theta}(V_{n;\mathring{t}})$ is an approximate likelihood computed by approximate estimation when conjugate $\hat{f}_n(\Theta) = Di_{\Theta}(V_{n;0})$, $n = 1, 2$, are used as prior pdfs.

The new candidate is defined

$$\hat{f}_3(\Theta|d(\mathring{t})) \quad \propto \quad \hat{f}_1^{\lambda}(\Theta|d(\mathring{t}))\hat{f}_2^{1-\lambda}(\Theta|d(\mathring{t})) \equiv Di_{\Theta}(V_{3;\mathring{t}}) \equiv Di_{\mathring{t}}(\lambda V_{1;\mathring{t}} + (1-\lambda)V_{2;\mathring{t}}), \quad \text{with} \tag{10.18}$$

$$\lambda = \frac{\hat{f}_1(d(\mathring{t}))}{\hat{f}_1(d(\mathring{t})) + \hat{f}_2(d(\mathring{t}))}, \ \hat{f}_n(d(\mathring{t})) = \int \hat{f}_n(d(\mathring{t})|\Theta)\hat{f}_n(\Theta)\, d\Theta, \quad n = 1, 2. \tag{10.19}$$

Use of (10.18), (10.19) and Proposition 10.1.4 describes the resulting new candidate.

Proposition 10.4.3 (Branching geometric mean of a pair Dirichlet pdfs) *Let*

$$\hat{f}_n(\Theta) \equiv Di_{\alpha}(\kappa_{n;0}) \prod_{c\in c^*} \prod_{i\in i^*} Di_{\Theta_{ic}}(V_{icn;0}),$$

$n = 1, 2$, *be prior pdfs employed in approximate estimation, see Section 10.5, of the Markov chain mixture*

$$f(d_t|d(t-1), \Theta) \equiv \sum_{c\in c^*} \alpha_c \prod_{i\in i^*} {}^{\lfloor d_{ic;t}|\psi_{ic;t}}\Theta_{ic}.$$

The posterior pdfs preserve the functional forms of the prior pdfs and they are described by statistics $V_{icn;\mathring{t}}$ *and* $\kappa_{cn;\mathring{t}}$.

Their geometric mean $f_3(\Theta|V_{3;\mathring{t}})$ *keeps this form and its statistics are*

$$\begin{aligned} V_{3ic;\mathring{t}} &= \lambda_{1;\mathring{t}} V_{ic1;\mathring{t}} + (1-\lambda_{1;\mathring{t}})V_{ic2;\mathring{t}} \\ \kappa_{3ic;\mathring{t}} &= \lambda_{1;\mathring{t}}\kappa_{ic1;\mathring{t}} + (1-\lambda_{1;\mathring{t}})\kappa_{ic2;\mathring{t}} \\ \lambda_{1;\mathring{t}} &= \left[1 + \prod_{t\in t^*} \frac{\sum_{c\in c^*}\prod_{i\in i^*} \frac{\mathcal{I}(V_{ic2;t})}{\mathcal{I}(V_{ic2;t-1})} \frac{\kappa_{c2;t-1}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c}2;t-1}}}{\sum_{c\in c^*}\prod_{i\in i^*} \frac{\mathcal{I}(V_{ic1;t})}{\mathcal{I}(V_{ic1;t-1})} \frac{\kappa_{c1;t-1}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c}1;t-1}}}\right]^{-1}. \end{aligned} \tag{10.20}$$

The normalization integral $\mathcal{I}(V)$ *is given by the formula (10.3).*

Proof: Omitted. □

10.4.5 Random branching of statistics

The deterministic search algorithms are often stuck at a local optimum. This makes us to complement the deterministic search by a random one. Here, we discuss this possibility in connection with Markov chain factors.

Approximate estimation, Section 10.5 provides the posterior pdf $f(\Theta|d(\mathring{t}))$ as a product of the posterior Di_{Θ} pdfs corresponding to the individual, Markov-chain parameterized factors and to the posterior Dirichlet pdf on component weights. Thus, the following considerations and computations may deal with individual factors only and random step may be applied to their sub-selection.

The $Di_{\Theta}(V)$ factor is determined by the finite-dimensional (almost) sufficient statistics $V \in V^* \equiv \{$ collection of occurrence tables $V > 0$ with dimensions compatible with estimated $\Theta\,\}$. Knowing this, we generate a random sample $\tilde{V}$ in V^* describing the modified factor. It gives us a new guess $\tilde{f}(\Theta|d(\mathring{t})) \equiv Di_{\Theta}(\tilde{V})$. By flattening it towards a flat pdf $\bar{f} = Di_{\Theta}(\bar{V}$, given by a small positive $\bar{V}$, see Proposition 6.4.1, we get a new guess of the prior pdf $\hat{f}_{new}(\Theta) = Di_{\Theta}(\Lambda\tilde{V} + (1-\Lambda)\bar{V})$. Even for a single factor, the computational burden induced by the random choice is usually prohibitive. Thus, it is reasonable to exploit interpretation of parts of statistics and change them partially only. At present, it seems wise

- to estimate the factor structure before generating new statistics;
- to change the point estimate of parameters using normal approximation, based on coincidence of moments cf. Proposition 10.1.1, of this Dirichlet pdf

$${}^{\lfloor d|\psi}\tilde{\hat{\Theta}} = \max\left[{}^{\lfloor d|\psi}\bar{\hat{\Theta}},\ {}^{\lfloor d|\psi}\hat{\Theta}\left(1 + e\rho\sqrt{\frac{{}^{\lfloor d|\psi}\hat{\Theta}^{-1} - 1}{{}^{\lfloor\psi}\nu + 1}}\right)\right], \ e \sim \mathcal{N}_e(0, 1), \quad \text{where} \tag{10.21}$$

$$
\begin{aligned}
{}^{\lfloor d|\psi}\bar{\hat{\Theta}} > 0 & \quad \text{is a pre-selected lower bound },\ \rho \in \rho^* \equiv (1,5) \text{ re-normalize the point estimates} \\
{}^{\lfloor d|\psi}\tilde{\hat{\Theta}} & \equiv {}^{\lfloor d|\psi}\tilde{\hat{\Theta}} \frac{\nu}{\nu - {}^{\lfloor d|\psi}\hat{\Theta} + {}^{\lfloor d|\psi}\tilde{\hat{\Theta}}} \\
\tilde{\nu} & \equiv \nu(1 - {}^{\lfloor d|\psi}\hat{\Theta} + {}^{\lfloor d|\psi}\tilde{\hat{\Theta}}) \text{ and define the new occurrence table} \\
{}^{\lfloor d|\psi}V & \equiv \tilde{\nu}\, {}^{\lfloor d|\psi}\tilde{\hat{\Theta}}.
\end{aligned}
$$

Statistics κ determining the Dirichlet pdf of component weights are modified in the same ways as in (8.54). It gives to the components with $\hat{\alpha}_c \approx 0$ a chance to be modified. The normal pdf cut so that the smallest κ_c does not decrease is approximately used. It gives equivalent of (8.54)

$$
\tilde{\kappa}_c = \max\left[\kappa_c\left(1 + \frac{\rho}{\sqrt{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c}}}} e_c\right), \min_{\tilde{c}\in c^*}\kappa_{\tilde{c}}\right],\ \rho \in (1,5),\ f(e_c) = \mathcal{N}_{e_c}(0,1).
$$

Remark(s) 10.4.1

1. *The chosen scaling of new point estimates to the occurrence table preserves the value of* ${}^{\lfloor\psi}\nu = \sum_{d\in d^*} {}^{\lfloor d|\psi}V$ *that determines uncertainty of the inspected Dirichlet pdf.*
2. *Random generating of statistics can be used as a part of various compound algorithms constructing the prior pdf.*
3. *The optional value* ρ *is tuning knob of generators. Its recommended range stems from standard properties of the normal pdf used instead of the correct Dirichlet pdf. At present, values of* ρ *in a more narrow range* $\rho^* \equiv (1,2)$ *seem to be preferable.*

10.4.6 Prior-posterior branching

The prior-posterior branching, see Section 6.4.6, starts at some prior pdf, performs an approximate estimation, usually quasi-Bayes estimation, see Algorithm 6.5.2, and flatten the resulting posterior pdf so that it provides a new alternative to the prior pdf used. The specialization of the general results on flattening gives directly the following iterative Bayesian learning of Markov chain mixtures.

Algorithm 10.4.1 (Prior-posterior branching with geometric mean)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number of iterations* n *and set* $n = 1$.
- *Select sufficiently rich structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to considered components. The structure of each factor* ic *is determined by the structure of the corresponding data vector* Ψ_{ic}.
- *Select a flat pre-prior pdf* $\bar{f}(\Theta)$ *in the form (6.3) with Dirichlet pdfs describing individual factors. It means, select pre-prior statistics* $\bar{V}_{ic}$, $\bar{\kappa}_c$ *determining Dirichlet pdfs on the transition probabilities* ${}^{\lfloor d|\psi}\Theta$ *and on component weights* α *with a high variance, cf. Proposition 10.1.1. Typically,* $\bar{V}_{ic} = \bar{\kappa}_c = \varepsilon > 0$, ε *small. They serve as alternatives in flattening.*
- *Select a prior pdf* $f(\Theta)$ *in the form (6.3) with Dirichlet pdfs describing individual factors. It means, select prior statistics* V_{ic}, κ_c *determining the involved Dirichlet pdfs, cf. Proposition 10.1.1. Note that generally* $\bar{f}(\Theta) \neq f(\Theta)$.
- *Set* $\hat{f}_{1n}(\Theta) \equiv f(\Theta)$, *i.e. set* $V_{ic1n;0} = V_{ic}$, $\kappa_{c1n;0} = \kappa_c$ *giving* $\hat{f}_{1n}(\Theta_{ic}) = \prod_{\Psi\in\Psi^*} Di_{\,{}^{\lfloor d|\psi}\Theta_{ic}}\left({}^{\lfloor d|\psi}V_{ic1n;0}\right)$ *and* $\hat{f}_{1n}(\alpha) = Di_\alpha(\kappa_{1n;0})$.
- *Compute the posterior pdf*

$$
\tilde{f}_{1n}(\Theta|d(\mathring{t})) = Di_\alpha(\kappa_{1n;\mathring{t}}) \prod_{i\in i^*,\Psi\in\Psi^*} Di_{\,{}^{\lfloor d|\psi}\Theta_{ic}}\left({}^{\lfloor d|\psi}V_{ic1n;\mathring{t}}\right)
$$

using an approximate Bayesian estimation that starts at $\hat{f}_{1n}(\Theta)$, *see Section 10.5.*

- *Evaluate v-likelihood* l_{1n} *resulting from the use of* $\hat{f}_{1n}(\Theta)$.
- *Apply flattening operation to* $\tilde{f}_{1n}(\Theta|d(\mathring{t}))$ *according to Algorithm 6.4.2. Call the resulting pdf* $\hat{f}_{2n}(\Theta)$. *For the component weights* α, *the flattening factor is*

$$\Lambda_D \equiv \frac{\sum_{c\in c^*}(\kappa_c - \bar{\kappa}_c)}{\sum_{\tilde{c}\in c^*}(\kappa_{\tilde{c}1n;\mathring{t}} - \bar{\kappa}_{\tilde{c}})}.$$

For Dirichlet estimates of Markov chain factors, the flattening is determined by

$$\Lambda_{ic|\psi} \equiv \frac{\sum_{d\in d^*}\left({}^{\lfloor d|\psi}V_{ic} - {}^{\lfloor d|\psi}\bar{V}_{ic}\right)}{\sum_{\tilde{d}\in d^*}\left({}^{\lfloor \tilde{d}|\psi}V_{ic1n;\mathring{t}} - {}^{\lfloor \tilde{d}|\psi}\bar{V}_{ic}\right)}.$$

It provides a new guess of the prior pdf with the preserved form (6.3) given by

$$\kappa_{c2n;0} = \Lambda_D\kappa_{c1n;\mathring{t}} + (1-\Lambda_D)\bar{\kappa}_c, \quad {}^{\lfloor d|\psi}V_{ic2n;0} = \Lambda_{ic|\psi}\,{}^{\lfloor d|\psi}V_{ic1n;\mathring{t}} + (1-\Lambda_{ic|\psi})\,{}^{\lfloor d|\psi}\bar{V}_{ic}.$$

- *Compute the posterior pdf* $\tilde{f}_{2n}(\Theta|d(\mathring{t})) = Di_\alpha(\kappa_{2n;\mathring{t}})\prod_{i\in i^*,c\in c^*,d\in d^*,\psi\in\psi^*} Di_{\,{}^{\lfloor d|\psi}\Theta_{ic}}\left({}^{\lfloor d|\psi}V_{icn;\mathring{t}}\right)$ *using an approximate Bayesian estimation that starts at* $\hat{f}_{2n}(\Theta)$, *see Section 10.5.*
- *Evaluate v-likelihood* l_{2n} *resulting from the use of* $\hat{f}_{2n}(\Theta)$.
- *Set* $\bar{l}_n = \max(l_{1n}, l_{2n})$.

Iterative mode

1. *Apply geometric branching to the pair* $\tilde{f}_{1n}(\Theta|d(\mathring{t}))$, $\tilde{f}_{2n}(\Theta|d(\mathring{t}))$ *with v-likelihood* l_{1n}, l_{2n}, *respectively. For* $\lambda \equiv \frac{l_{1n}}{l_{1n}+l_{2n}}$, *it gives* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$ *of the form (6.3) with Dirichlet descriptions of factors. It is determined by the statistics*

$$\kappa_{c3n;\mathring{t}} = \lambda\kappa_{c1n;\mathring{t}} + (1-\lambda)\kappa_{c2n;\mathring{t}}, \; V_{ic3n;\mathring{t}} = \lambda V_{ic1n;\mathring{t}} + (1-\lambda)V_{ic2n;\mathring{t}}.$$

2. *Apply the flattening operation to* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$ *according to Algorithms 6.4.2. Call the resulting pdf* $\hat{f}_{3n}(\Theta)$. *It preserves the form (6.3). The flattening factors are*

$$\Lambda_D \equiv \frac{\sum_{c\in c^*}(\kappa_{c3n} - \bar{\kappa}_c)}{\sum_{\tilde{c}\in c^*}(\kappa_{\tilde{c}3n;\mathring{t}} - \bar{\kappa}_{\tilde{c}})}, \quad \Lambda_{ic|\psi} \equiv \frac{\sum_{d\in d^*}\left({}^{\lfloor d|\psi}V_{ic3} - {}^{\lfloor d|\psi}\bar{V}_{ic}\right)}{\sum_{\tilde{d}\in d^*}\left({}^{\lfloor \tilde{d}|\psi}V_{ic3n;\mathring{t}} - {}^{\lfloor \tilde{d}|\psi}\bar{V}_{ic}\right)}.$$

They provide the new statistics

$$\kappa_{c3n;0} = \Lambda_D\kappa_{c3n;\mathring{t}} + (1-\Lambda_D)\bar{\kappa}_c, \quad {}^{\lfloor d|\psi}V_{ic3n;0} = \Lambda_{ic|\psi}\,{}^{\lfloor d|\psi}V_{ic3n;\mathring{t}} + (1-\Lambda_{ic|\psi})\,{}^{\lfloor d|\psi}\bar{V}_{ic}.$$

3. *Evaluate v-likelihood* l_{3n} *resulting from the use of* $\hat{f}_{3n}(\Theta)$ *for determining of* $\tilde{f}_{3n}(\Theta|d(\mathring{t}))$. *The approximation prepared for the fixed advisory system, see Section 6.1.2, has to be used.*
4. *Choose among* $\tilde{f}_{in}(\Theta|d(\mathring{t}))$, $i \in \{1,2,3\}$ *the pair with the highest v-likelihood values and call them* $\tilde{f}_{1(n+1)}(\Theta|d(\mathring{t}))$, $\tilde{f}_{2(n+1)}(\Theta|d(\mathring{t}))$ *with v-likelihood* $l_{1(n+1)}$, $l_{2(n+1)}$.
5. *Go to the beginning of* Iterative mode *with* $n = n+1$ *if* $\bar{l}_{n+1} \equiv \max(l_{1(n+1)}, l_{2(n+1)}) > \bar{l}_n$ *or if* $\bar{l}_{n+1}, \bar{l}_n$ *are the same according to Proposition 6.1.2 and* $n < \mathring{n}$.
6. *Stop and select among* $\hat{f}_{in}(\Theta)$, $i = 1,2$ *that leading to the higher value of* l_{in} *and take it as the prior pdf constructed.*

Remark(s) 10.4.2

1. *No improvement can be expected when the values of* Λ *are around 0.5. This observation may serve as an additional stopping rule.*
2. *The structure of the estimated mixture does not change during iterations. Thus, it has to be sufficiently reflected in the prior pdf* $f(\Theta)$.
3. *Prediction of the v-likelihood of connected with the prior pdf obtained flattened geometric mean is possible in the way used in connection with component merging, Section 10.6.3.*

10.4.7 Branching by forgetting

Branching by forgetting, see Section 6.4.7, makes parallel recursive estimation with no and a fixed forgetting with forgetting factor smaller than one. The alternative pdfs are compared according to their v-likelihood values. At the moment, when one of them is a sure winner they are bounded into a single pdf and whole process is repeated. Here, the algorithm is specialized to Markov chain mixtures.

Algorithm 10.4.2 (On-line branching with forgetting for Markov chain mixtures)
Initial mode

- *Select a sufficiently rich structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to the considered components. The structure of each factor* ic *is determined by the structure of the corresponding data vector* Ψ_{ic}.
- *Select a constant* $\rho \approx 3-5$ *defining the significant difference of* v*-log-likelihood values.*
- *Set the record counter* $t = 0$ *and select the statistics determining the prior pdf*

$$\hat{f}(\Theta_{ic}) = \prod_{\Psi \in \Psi^*} Di_{\lfloor d|\psi}\Theta_{ic} \left({}^{\lfloor d|\psi}V_{ic;0} \right), \; i \in i^*, \; c \in c^*$$

for factors as well as of the component weights $\hat{f}(\alpha) = Di_{\alpha}(\kappa_0)$.

- *Choose a fixed relatively small forgetting factor* $\lambda < 1$.
- *Select a fixed pre-prior alternative pdf used in the stabilized forgetting, see Proposition 3.1.1. The alternative is usually taken as a flat pre-prior pdf of the Dirichlet version of (6.3) given by small* ${}^{\lfloor A}\kappa = {}^{\lfloor Ad|\psi}V_{ic} = \varepsilon > 0$, ε *small.*

Data processing mode, *running for* $t = 1, 2, \ldots,$

1. *Set* $\hat{f}_1(\Theta|d(t)) = \hat{f}_\lambda(\Theta|d(t)) = \hat{f}(\Theta|d(t))$.
2. *Initialize* v*-log-likelihood assigned to considered alternatives* $l_{1;t} = 0$, $l_{\lambda;t} = 0$.
3. *Collect new data* d_t *and construct the data vector* Ψ_t.
4. *Update* $\hat{f}_1(\Theta|d(t-1))$ *to* $\hat{f}_1(\Theta|d(t))$ *using approximate estimation, Section 10.5, with the forgetting factor 1.*
5. *Re-compute the* v*-log-likelihood* $l_{1;t-1}$ *to* $l_{1;t}$ *by adding the logarithm of the mixture prediction* $\ln(f(d(t)|d(t-1)))$ *obtained for the "prior" pdf* $\hat{f}_1(\Theta|d(t-1))$.
6. *Update* $\hat{f}_\lambda(\Theta|d(t-1))$ *to* $\hat{f}_\lambda(\Theta|d(t))$ *using approximate estimation with the forgetting factor* λ *and the chosen alternative. Thus, after obtaining statistics* $\kappa_{\lambda 1;t}$, $V_{ic\lambda 1;t}$ *through updating of* $\kappa_{\lambda;t-1}$, $V_{ic\lambda;t-1}$, *with forgetting 1, forget*

$$\kappa_{\lambda;t} = \lambda \kappa_{\lambda 1;t} + (1-\lambda)\, {}^{\lfloor A}\kappa, \quad V_{ic\lambda;t} = \lambda V_{ic\lambda 1;t} + (1-\lambda)\, {}^{\lfloor A}V_{ic}.$$

7. *Re-compute the* v*-log-likelihood* $l_{\lambda;t-1}$ *to* $l_{\lambda;t}$ *by adding logarithm of the mixture prediction* $\ln(f(d(t)|d(t-1)))$ *obtained for the "prior" pdf* $\hat{f}_\lambda(\Theta|d(t-1))$.
8. *Go to the step 3 with* $t = t+1$ *if* $|l_{1;t} - l_{\lambda;t-1}| < \rho$
9. *Set* $\hat{f}(\Theta|d(t)) = \hat{f}_1(\Theta|d(t))$ *if* $l_{1;t} > l_{\lambda;t}$ *otherwise set* $\hat{f}(\Theta|d(t)) = \hat{f}_\lambda(\Theta|d(t))$.
   This setting means assignment of the appropriate sufficient statistics.
10. *Go to the beginning of* Data processing mode *if* $t \leq \mathring{t}$. *Otherwise stop and take* $\hat{f}_1(\Theta|d(\mathring{t}))$ *as the final estimate.*

Remark(s) 10.4.3

1. *Speeding up of the learning is the main expectation connected with this algorithm. The model with no forgetting is expected to be winner in a long run.*

   The estimation with forgetting is to be switched off when the estimation without forgetting is better for a majority of time. This rule is implemented.

2. *The technique can be directly combined with the prior-posterior branching.*

3. *It will be necessary to get experience with choice of the fixed forgetting factor λ. The option $\lambda \approx 0.6$ seems to be satisfactory for normal mixtures. A similar value could be a reasonable start for a closer inspection in the Markov-chain case.*

10.4.8 Branching by factor splitting

This section specializes the algorithm described in Section 6.4.8. It covers important cases when we have no clue on structure of the mixture. The algorithm simply splits factors suspicious for hiding more modes of the correct description and makes a new learning attempt.

New factors should remain in class of Dirichlet pdfs. The flattened/sharpened factors found according to Proposition 6.4.7 have this property.

Proposition 10.4.4 (Flattened/sharpened Markov chain factors) *Let us consider pdfs $f = Di_{\Theta}(V)$, $\bar{f} = Di_{\Theta}\left(\bar{V}\right)$. The pdf minimizing the KL distance $\mathcal{D}\left(\hat{f}||f\right)$, and having the KL distance*

$$\begin{aligned} \mathcal{D}\left(\hat{f}||\bar{f}\right) &= \omega\mathcal{D}\left(f||\bar{f}\right),\ \omega \neq 1 \text{ has the form} \\ \hat{f} &= Di_{\Theta}(\lambda V + (1-\lambda)\bar{V}) \end{aligned} \tag{10.22}$$

with the scalar λ chosen so that the restriction (10.22) is met.

Proof: Proposition 6.4.7 implies the form of $\hat{f}$ and the restriction determines the values (10.22) of λ. □

Problem 10.4.1 (Shifted *Di* factor) *The explicit shift in expected value of some function $g(\Theta)$ of parameters Θ is intuitively preferable. The choice $g(\Theta) = [\ln(\Theta_1), \ldots, \ln(\Theta_{\mathring{d}})]'$ guarantees that the factor found according to Proposition 6.4.8 stay within the desirable Dirichlet class. The statistics of the resulting pdf solve complex algebraic equations resulting from necessary conditions on extreme. This direction was not elaborated further on but it is worth of considering.*

10.4.9 Hierarchical selection of split factors

The *hierarchical splitting* described here relies on our ability to estimate three factors instead of a single one.

During approximate estimation, Section 10.5, the i-th factor within c-th component is assigned a weight $w_{ic;t} \in [0,1]$ that expresses the expectation that the data item $d_{ic;t}$ is generated by this factor. Then, the modified parameterized factor

$$f_w(d_{ic;t}|d_{i+1\_\mathring{d};t}, d(t-1)) \equiv f_w(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) \propto [f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic})]^{w_{ic;t}}$$

is used in "ordinary" Bayes rule for updating of parameter estimates of this factor. Thus, for estimation purposes, we can inspect this factor independently of others. Thus, we can drop temporarily the subscripts w, c.

We want to check whether the discussed factor explains data that should be modelled by more factors while assuming that Requirement 6.4.1 is met, i.e. the split factor is conditioned by regression vector that includes the true one. We formulate hypothesis H_0 that the objective pdf has two components

$$^{\lfloor o}f(d_t|\psi_t) = \beta\, ^{\lfloor d_t|\psi_t}\Theta_1 + (1-\beta)\, ^{\lfloor d_t|\psi_t}\Theta_2,\ \beta \in (0,1),\ \text{where} \tag{10.23}$$

Θ_1, Θ_2 have structures differing from Θ in the split factor as they contain more zero entries.

The alternative hypothesis H_1 assumes that ${}^{\lfloor o}f(d_t|\psi_t) = {}^{\lfloor d_t|\psi_t}\Theta$, i.e. denies the need for splitting.

We estimate parameters of the mixture (6.61) together with the factor in question in order to decide on the need to split. With $f(H_0) = f(H_1)$, modelling no prejudice, the Bayes rule gives

$$Probability\left(\text{split is needed}|d(\mathring{t})\right) \equiv f(H_0|d(\mathring{t})) = \frac{f(d(\mathring{t})|H_0)}{f(d(\mathring{t})|H_0) + f(d(\mathring{t})|H_1)}. \tag{10.24}$$

The v-likelihood values $f(d(\mathring{t})|H_0)$, $f(d(\mathring{t})|H_1)$ are obtained as byproduct of the approximate estimation, Section 10.5.

The factor is split if the probability (10.24) is high enough. The estimated factors of the mixture (10.23) are natural candidates for its replacement. We have to define initial estimates of the "small mixture" (10.23). Its components should be dissimilar and their mixture should be close the mixture split. We shall reach this property by splitting (possibly in a random way)

$$d^* = d_1^* \cup d_2^*, \; d_1^* \cap d_2^* = \emptyset \tag{10.25}$$

so that cardinalities of both sets are approximately equal. For the given occurrence matrix $V_{d|\psi;0}$ defining prior pdf on the considered factor, we specify initial occurrence matrices, $\forall \psi \in \psi^*$ and $\varepsilon \in \left(0, \min_{d\in d^*, \psi \in \psi^*} {}^{\lfloor d|\psi}V_0\right)$,

$${}^{\lfloor d|\psi}V_{1;0} = \begin{cases} {}^{\lfloor d|\psi}V_0 - \varepsilon & \forall\, d \in d_1^* \\ \varepsilon & \forall\, d \in d_2^* \end{cases}, \quad {}^{\lfloor d|\psi}V_{2;0} = \begin{cases} {}^{\lfloor d|\psi}V_0 - \varepsilon & \forall\, d \in d_2^* \\ \varepsilon & \forall\, d \in d_1^* \end{cases}. \tag{10.26}$$

We also set the initial statistics $\kappa_{1;0} = \kappa_{2;0} = \kappa_0$, where κ_0 is the relevant entry of the initial statistics determining estimates of component weights α. With this definition, we can describe the Markov-chain version of Algorithm 6.4.9

Algorithm 10.4.3 (Hierarchical split of Markov chain factors)
Initial mode

- *Construct the very initial estimate of mixture parameters, see Algorithm 6.4.7,*

$$f(\Theta) \propto Di_\alpha(\kappa_0) \prod_{c\in c^*} \prod_{d=1}^{\mathring{d}} \prod_{\psi\in\psi^*} \left[{}^{\lfloor d|\psi}\Theta\right]^{{}^{\lfloor d|\psi}V_0 - 1}.$$

- *Select significance level $\bar{P} \in (0,1)$, $\bar{P} \approx 1$ controlling the need to split.*
- *Select $\varepsilon \in \left(0, \min_{d\in d^*, \psi\in\psi^*} {}^{\lfloor d|\psi}V_0\right)$.*
- *Define sets d_1^*, d_2^* according to (10.25).*
- *Assign to each factor $i = 1, \ldots, \mathring{d}$, $c \in c^*$, a prior pdf $f(\beta_{ic}) = Di_{\beta_{ic}}(0.5\kappa_{c;0}[1,1])$, and, for $j = 1,2$ cf. (10.26),*

$$f(\Theta_{jic}|d(0)) = \prod_{\psi\in\psi^*} Di_{\{{}^{\lfloor d|\psi}\Theta_{jic}\}_{d\in d^*}}\left({}^{\lfloor d|\psi}V_{jic;0},\, d \in d^*\right), \text{ on parameters of the mixture (10.23).}$$

Indices ic stress "membership" of this mixture to the factor ic.

Sequential mode, *running for $t = 1, 2, \ldots,$*

1. *Acquire data d_t and complement data vectors $\Psi_{i;t} = [d_{i;t}, \psi_{it}']'$.*
2. *Perform single step of an approximate estimation, i.e. update $f(\Theta|d(t-1)) \rightarrow f(\Theta|d(t))$, Section 10.5. It generates weights $w_{ic;t}$ measuring the degree with which data are assigned to respective factors.*

   Update at the same time respective values of the v-likelihood $l_{ic;t|H_1} \equiv f(d_{ic;t}|\psi_{ic;t}, d(t-1)) l_{ic;t-1|H_1}$. Here, $d(t-1)$ in condition stresses that the predictors for all possible discrete values $d_{ic;t}|\psi_{ic;t}$ are computed using the observed data $d(t-1)$.

3. *Weight new data by respective $w_{ic;t}$ and update estimates of parameters of the models (10.23)*

$$f(\beta_{ic}, \Theta_{1ic}, \Theta_{2ic}|d(t-1)) \rightarrow f(\beta_{ic}, \Theta_{1ic}, \Theta_{2ic}|d(t)).$$

Update also values of the v-likelihood $l_{ic;t|H_0} = f(d_{ic;t}|\psi_{ic;t}, d(t-1))l_{ic;t-1|H_0}$ using predictors $f(d_{ic;t}|\psi_{ic;t}, d(t-1))$ constructed from the estimated two-component mixtures.

4. *Go to the beginning of* Sequential mode *while $t < \mathring{t}$.*

Selection mode

1. *For $i = 1, \ldots, \mathring{d}$, $c \in c^*$.*
2. *Compare $P_{ic} \equiv \frac{l_{ic;\mathring{t}|H_0}}{l_{ic;\mathring{t}|H_0} + l_{ic;\mathring{t}|H_1}}$ with $\bar{P}$.*
3. *Split $f(\Theta_{ic}|d(\mathring{t})) \rightarrow \left(f(\Theta_{1ic}|d(\mathring{t})), f(\Theta_{2ic}|d(\mathring{t}))\right)$ if $P_{ic} \geq \bar{P}$.*

10.4.10 Techniques applicable to static mixtures

The specific Markov chain form of the considered model brings nothing specific to this counter-part of Chapter 6. Validity of the assumptions of Proposition 6.4.10 is only worth of mentioning.

10.5 Approximate parameter estimation

10.5.1 Quasi-Bayes estimation

A direct application of Algorithm 6.5.2 gives the specialized algorithm.

Algorithm 10.5.1 (Quasi-Bayes algorithm with common factors in Markov components)
Initial (off-line) mode

1. *Select statistics $V_{ic;0}$ determining prior distributions of the individual factors $Di_{\Theta_{ic}}(V_{ic;0})$.*
2. *Select initial values $\kappa_{c;0} > 0$ determining Dirichlet distribution $Di_{\alpha}(\kappa_0)$ of weights α, say, $\kappa_{c;0} \approx$ 10% of the length $\mathring{t}$ of the processed data.*
3. *Select forgetting factor and alternative statistics needed for stabilized forgetting, see (10.8).*

Sequential (on line) mode, *running for $t = 1, 2, \ldots,$*

1. *Acquire the data record d_t.*
2. *Evaluate values of the predictive pdfs (2.47)*

$$f(d_{ic;t}|\psi_{ic;t}, d(t-1), c) = \frac{\mathcal{I}(d(t)|ic)}{\mathcal{I}(d(t-1)|ic)} = \frac{{}^{d_{ic;t}|\psi_{ic;t}}V_{t-1}}{\sum_{d_{ic}=1}^{\mathring{d}_i} {}^{d_{ic}|\psi_{ic;t}}V_{t-1}}.$$

3. *Compute the values of predictive pdfs $f(d_t|d(t-1), c) = \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)$ for each individual component $c \in c^*$.*
4. *Compute the probabilities $w_{c;t}$, using the formula (6.83)*

$$\begin{aligned} w_{c;t} &\equiv f(c_t = c|d(t)) = \frac{\frac{\kappa_{c;t-1}}{\sum_{\tilde{c} \in c^*} \kappa_{\tilde{c};t-1}} \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)}{\sum_{\tilde{c} \in c^*} \frac{\kappa_{\tilde{c};t-1}}{\sum_{\tilde{c} \in c^*} \kappa_{\tilde{c};t-1}} \prod_{i \in i^*} f(d_{i\tilde{c};t}|\psi_{i\tilde{c};t}, d(t-1), \tilde{c})} = \\ &= \frac{\kappa_{c;t-1} \prod_{i \in i^*} f(d_{ic;t}|\psi_{ic;t}, d(t-1), c)}{\kappa_{\tilde{c};t-1} \sum_{\tilde{c} \in c^*} \prod_{i \in i^*} f(d_{i\tilde{c};t}|\psi_{i\tilde{c};t}, d(t-1), \tilde{c})}. \end{aligned}$$

5. *Update scalars* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}$, *cf. (6.85).*

6. *Update Bayesian parameter estimates of different factors, i.e. update the corresponding statistics* $V_{ic\Psi;t-1} \rightarrow V_{ic\Psi;t}$ *equivalent to the weighted version of (10.8)*

$$\begin{aligned} {}^{\lfloor d|\psi}Vic;t &= \lambda\left({}^{\lfloor d|\psi}V_{ic;t-1} + w_{ic;t}\delta_{\Psi_{ic;t},\Psi}\right) + (1-\lambda)\,{}^{\lfloor Ad|\psi}V_{ic},\ \ {}^{\lfloor d|\psi}V_{ic;0} > 0 \text{ given and} \\ w_{ic;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c};t}. \end{aligned} \tag{10.27}$$

The set c_i^* *consists of pointers to components that contain the* i*-th factor.*

7. *Evaluate, if need be, characteristics of the Bayesian estimate of the component weights or other parameters.*

8. *Go to the beginning of* Sequential mode *while data are available.*

10.5.2 EM estimation

A direct application of EM algorithm 6.5.4 to Markov chain factors gives the adequate specialization.

Algorithm 10.5.2 (EM estimation of Markov chain mixture with common factors)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 0$.
- *Select point estimates* $\hat{\Theta}_{icn} \equiv \left[{}^{\lfloor d|\psi}\hat{\Theta}_{icn}\right]$ *of parameters* $\left[{}^{\lfloor d|\psi}\Theta_{ic}\right] \equiv$ *probabilities of* $d_{ic} = d$ *when conditioned by the regression vector* ψ *within the* c*-th component.*
- *Select point estimates* $\hat{\alpha}_{cn} = 1/\mathring{c}$ *of components weights* α_c.

Iterative mode

1. *Fill the occurrence matrices* $V_{ic;0} = \varepsilon$ *and the statistic* $\kappa_{c;0} = \varepsilon$, *where* $\varepsilon > 0,\ \varepsilon \approx 0$.

   Sequential mode, *running for* $t = 1, 2, \ldots,$

   (a) *Construct the data vectors* $\Psi_{ic;t}$.

   (b) *Compute values of the predictive pdfs* $f\left(d_{ic;t}|\psi_{ic;t}, \hat{\Theta}_{icn}, c\right) \equiv {}^{\lfloor d_{ic;t}|\psi_{ic;t}}\hat{\Theta}_{icn}$ *for each individual factor* $i \in i^* = \left\{1, \ldots, \mathring{d}\right\}$ *in all components* $c \in c^*$ *using the parameter estimates* $\hat{\Theta}_{icn}$ *that are constant during time cycle of the sequential mode.*

   (c) *Compute the values of the predictive pdfs* $f(d_t|\phi_{c;t-1}, \hat{\Theta}_{cn}, c) \equiv \prod_{i\in i^*} {}^{\lfloor d_{ic;t}|\psi_{ic;t}}\hat{\Theta}_{icn}$ *for each individual component* $c \in c^*$.

   (d) *Compute the probabilities* $w_{c;t}$ *approximating* δ_{c,c_t}

   $$w_{c;t} = \frac{f\left(d_t|\phi_{c;t-1}, \hat{\Theta}_c, c\right)\hat{\alpha}_{cn}}{\sum_{\tilde{c}\in c^*} f(d_t|\phi_{\tilde{c};t-1}, \hat{\Theta}_{cn}, \tilde{c})\hat{\alpha}_{\tilde{c}n}}.$$

   (e) *Update the statistics determining log-likelihood values describing different factors*

   $$\begin{aligned} {}^{\lfloor d|\psi}V_{ic;t} &= {}^{\lfloor d|\psi}V_{ic;t-1} + w_{ic;t}\delta_{\Psi_{ic;t},\Psi}, \text{ where} \\ w_{ic;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c};t}. \end{aligned} \tag{10.28}$$

   The set $c_i^* \equiv$ *consists of pointers to components that contain the* i*-th factor.*

   (f) *Update the statistics* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}$, $c \in c^*$.

(g) *Go to the beginning of* Sequential mode *if $t \leq \mathring{t}$. Otherwise continue.*

2. *Find the new point estimates* $^{\lfloor d_i|\psi}\hat{\Theta}_{ic} = \frac{^{\lfloor d_i|\psi}V_{ic;\mathring{t}}}{\sum_{d_i \in d_i^*} {^{\lfloor d_i|\psi}V_{ic;\mathring{t}}}}$ *and* $\hat{\alpha}_c = \frac{\kappa_{c;\mathring{t}}}{\sum_{\tilde{c} \in c^*} \kappa_{\tilde{c};\mathring{t}}}$.

   These values maximize of the n-th approximate likelihood.

3. *Stop if the attained likelihood values*

$$\sum_{c \in c^*} \kappa_{c;\mathring{t}} \left[\ln(\hat{\alpha}_{c;\mathring{t}}) + \sum_{i \in i^*, \Psi \in \Psi^*} {^{\lfloor d|\psi}V_{ic;\mathring{t}}} \ln\left({^{\lfloor d|\psi}\hat{\Theta}_{ic}} \right) \right] \tag{10.29}$$

   do not increase any more. Otherwise set $n = n + 1$ and go to the beginning of Iterative mode.

10.5.3 Batch quasi-Bayes estimation

Here, Algorithm 6.5.5 is specialized to Markov chains. In this way, processing-order-independent, batch quasi-Bayes estimation, of Markov chains is gained.

Algorithm 10.5.3 (BQB estimation of Markov chain mixture with common factors)
Initial mode

- *Select the upper bound $\mathring{n}$ on the number n of iterations and set $n = 0$.*
- *Select the statistics $\bar{V}_{ic}$, $\bar{\kappa}_c$ determining the alternative used in flattening.*
- *Select the statistics V_{ic} determining distributions of individual factors $Di_{\Theta_{ic}}(V_{ic})$.*
- *Select initial values $\kappa_c > 0$.*

Iterative mode

1. *Make copies $V_{ic;0} = V_{ic}$ and $\kappa_{c;0} = \kappa_c$ and set v-log-likelihood l_0 of the mixture to zero.*

   Sequential mode, *running for $t = 1, 2, \ldots,$*

   (a) *Construct the data vectors $\Psi_{ic;t}$.*

   (b) *Compute values of the predictive pdfs*

$$f(d_{ic;t}|\psi_{ic;t}, c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c) f(\Theta_{ic}|V, \kappa)\, d\Theta_{ic} = \frac{^{\lfloor d_{ic;t}|\psi_{ic;t}}V_{ic}}{\sum_{\tilde{d} \in \tilde{d}_{ic}^*} {^{\lfloor \tilde{d}_{ic}|\psi_{ic;t}}V_{ic}}}.$$

   for each individual factor $i \in i^* = \{1, \ldots, \mathring{d}\}$ in all components $c \in c^*$ using the prior statistics V that are constant during the time cycle.

   (c) *Compute the values of predictive pdfs $f(d_t|\phi_{c;t-1}, c) \equiv \prod_{i \in i^*} f_n(d_{ic;t}|\psi_{ic;t}, c)$ for each individual component $c \in c^*$.*

   (d) *Compute the probabilities $w_{c;t}$ approximating δ_{c,c_t} $w_{c;t} \propto f(d_t|\phi_{c;t-1}, c)\kappa_c$.*

   (e) *Update statistics determining posterior pdfs evolving from copies of prior statistics $V_{ic;0}$ and $\kappa_{c;0}$.*

$$\begin{aligned} {^{\lfloor d|\psi}V_{ic;t}} &= {^{\lfloor d|\psi}V_{ic;t-1}} + w_{ic;t}\delta_{\Psi_{ic;t}, \Psi_{ic}} \\ \kappa_{c;t} &= \kappa_{c;t-1} + w_{c;t}, \text{ where} \\ w_{ic;t} &= \sum_{\tilde{c} \in c_i^*} w_{\tilde{c};t}. \end{aligned} \tag{10.30}$$

   The set $c_i^* \equiv$ consists of pointers to components that contain i-th factor.

(f) *Update v-log-likelihood of the mixture*

$$l_t = l_{t-1} + \ln\left[\sum_{c\in c^*} \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c};t}} f(d_t|\psi_{c;t}, c)\right].$$

(g) *Go to the beginning of* Sequential mode *if* $t \leq \mathring{t}$. *Otherwise continue.*

2. *Stop if the v-likelihood* $l_{\mathring{t}}$ *of the mixture does not increase among iterations. Otherwise apply flattening operation to* $f(\Theta_{ic}|d(\mathring{t})$ *and* $\kappa_{c;\mathring{t}}$

$$V_{ic} = \Lambda_n V_{ic|\mathring{t}} + (1-\Lambda_n)\bar{V}_{ic},\ \kappa_{ic} = \Lambda_n \kappa_{ic;\mathring{t}} + \Lambda_n \bar{\kappa}_c,\ \Lambda_n \to 0.5.$$

3. *Increase the counter to* $n = n+1$.

4. *Stop if* $n > \mathring{n}$ *otherwise go to the beginning of* Iterative mode.

10.5.4 Universal approximate estimation

Algorithm 10.5.4 (Universal estimation of Markov chain mixtures with common factors)
Initial mode

- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations and set* $n = 0$.
- *Select the structure of the mixture, i.e. specify the number of components* $\mathring{c}$ *and the ordered lists of factors allocated to the considered components. The structure of each factor* ic *is determined by the structure of the corresponding data vector* Ψ_{ic}.
- *Set the maximum of the v-log-likelihood* $\bar{l} = -\infty$.
- *Select the statistics* $\bar{V}_{ic}$ *determining (flat) pre-prior pdfs of the individual factors*

$$Di_{\Theta_{ic}}(\bar{V}_{ic}) \equiv \prod_{\Psi\in\Psi^*} Di_{\,{}^{\lfloor d|\psi}\Theta_{ic}}\left({}^{\lfloor d|\psi}\bar{V}_{ic}\right).$$

Select the values $\bar{\kappa}_c > 0$ *determining a (flat) pre-prior Dirichlet pdf on the component weights.*

- *Select the statistics* V_{ic} *determining prior pdfs of the individual factors*

$$Di_{\Theta_{ic}}(V_{ic}) \equiv \prod_{\Psi\in\Psi^*} Di_{\,{}^{\lfloor d|\psi}\Theta_{ic}}\left({}^{\lfloor d|\psi}V_{ic}\right).$$

- *Select the initial values* $\kappa_c > 0$ *determining a prior Dirichlet pdf on the component weights.*
- *Select the algorithmic variant* ***av*** $\in$ {**quasi-Bayes, quasi-EM, batch-quasi-Bayes, batch-EM**}.

Iterative mode

1. Make copies $V_{ic;0} = V_{ic}$ *and* $\kappa_{c;0} = \kappa_c$ *if* ***av*** *is a* batch variant.
2. *Compute the point estimates of the components weights* $\hat{\alpha}_c = \frac{\kappa_c}{\sum_{\tilde{c}\in c^*}\kappa_{\tilde{c}n}}$ *and point estimates*

$${}^{\lfloor d|\psi}\hat{\Theta}_{ic} = \frac{{}^{\lfloor d|\psi}V_{ic}}{\sum_{\tilde{d}\in d^*}{}^{\lfloor \tilde{d}|\psi}V_{ic}}$$

of parameters ${}^{\lfloor d|\psi}\Theta_{ic}$ *of the individual parameterized factors.*

3. *Set the value of v-log-likelihood* $l_{n;0} = 0$.
   Sequential mode, *running for* $t = 1, 2, \ldots,$

(a) *Construct the data vectors* $\Psi_{ic;t}$.

(b) *Compute values of the predictive pdfs*

$$f_n(d_{ic;t}|\psi_{ic;t}, c) = \int f(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c) f_n(\Theta_{ic})\, d\Theta_{ic} \propto {}^{\lfloor d_{ic;t}|\psi_{ic;t}}V_{ic}$$

for each individual factor $i \in i^* = \{1, \ldots, \mathring{i}\}$ *in all components* $c \in c^*$ *using the pdfs* $Di_{\Theta_{ic}}(V_{ic})$.[1]

(c) *Compute the values of the predictive pdfs* $f(d_t|\phi_{c;t-1}, c) \equiv \prod_{i\in i^*} f(d_{ic;t}|\psi_{ic;t}, c)$ *for each component* $c \in c^*$.[1]

(d) *Update v-log-likelihood*[1] $l_{n;t} = l_{n;t-1} + \ln\left(\sum_{c\in c^*} \hat{\alpha}_c f(d_t|\phi_{c;t-1}, c)\right)$.

(e) *Compute the probabilistic weights* $w_{c;t}$ *approximating* δ_{c,c_t} *according to the variant* **av** *used*

av= *-Bayes

$$w_{c;t} \equiv \frac{\hat{\alpha}_c f(d_t|\phi_{c;t-1}, c)}{\sum_{\tilde{c}\in c^*} \hat{\alpha}_{\tilde{c}} f(d_t|\phi_{\tilde{c};t-1}, \tilde{c})},$$

av=*-EM $w_{c;t} \propto \hat{\alpha}_c \prod_{i\in i^*} {}^{\lfloor d_{ic;t}|\psi_{ic;t}}\hat{\Theta}_{ic}$.

(f) *Perform data updating*

$$\begin{aligned} {}^{\lfloor d_{ic;t}|\psi_{ic;t}}V_{ic;t} &= {}^{\lfloor d_{ic;t}|\psi_{ic;t}}V_{ic;t-1} + w_{ic;t} \\ \kappa_{c;t} &= \kappa_{c;t-1} + w_{c;t}, \quad \textit{with} \\ w_{ic;t} &= \sum_{\tilde{c}\in c_i^*} w_{\tilde{c};t}. \end{aligned}$$

The set c_i^* *includes the pointers to the components that contain* i*-th factor.*

(g) *Correct sources of copies* $\kappa_c = \kappa_{c;t}$, ${}^{\lfloor d_{ic}|\psi_{ic;t}}V_{ic} = {}^{\lfloor d_{ic}|\psi_{ic;t}}V_t$, $d_{ic} \in d_{ic}^*$ *as well as the corresponding point estimates of parameters*

$$\hat{\alpha}_c = \frac{\kappa_c}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c}}}, \quad {}^{\lfloor d_{ic}|\psi_i}\hat{\Theta}_{ic} = \frac{{}^{\lfloor d_{ic}|\psi_i}V_t}{\sum_{\tilde{d}_{ic}\in d_{ic}^*} {}^{\lfloor \tilde{d}_{ic}|\psi_{ic}}V_t}.$$

if ***av=quasi***.

(h) *Update v-log-likelihood of the mixture.*

(i) *Go to the beginning of* Sequential mode *if* $t \leq \mathring{t}$. *Otherwise continue.*

4. *Stop* $l_{n;\mathring{t}} < \bar{l}$ *and* $l_{n;\mathring{t}}, \bar{l}$ *are not equal in the sense of Proposition 6.1.2 or* $n \geq \mathring{n}$.

5. *Set* $\bar{l} = l_{n;\mathring{t}}$, *increase the iteration counter* $n = n + 1$.

6. *Select flattening rate* Λ_n *according to Proposition 6.4.4 if* ***av =*-Bayes***, *otherwise set* $\Lambda_n \approx 0$.

7. *Apply the flattening operation*

$$^{\lfloor d|\psi}V_{ic} = \Lambda_n\, {}^{\lfloor d|\psi}V_{ic;\mathring{t}} + (1 - \Lambda_n)\, {}^{\lfloor d|\psi}\bar{V}_{ic}.$$

8. *Go to the beginning of* Iterative mode.

[1] This evaluation is not necessary for EM versions but suits for a use in a wider Bayesian context.

10.6 Structure estimation

10.6.1 Estimation of factor structure

Estimation of a factor structure can be and will be based on Bayesian structure estimation using the fact that Markov chain belongs to nested exponential family. The technique is applicable to Markov chain with a low dimensional regression vector ψ_r corresponding to the richest structure. We have to consider it anyway.

Proposition 10.6.1 (Structure estimation of nested Markov chains) *Let us consider Markov chain with the richest regression vector* ψ_r. $f(d|\psi_r, \Theta_r) = {}^{\lfloor d|\psi_r}\Theta_r$ *and another model* $f(d|\psi, \Theta) = {}^{\lfloor d|\psi}\Theta$ *describing the same data* $d(\mathring{t})$.

Let $\psi_s = N\psi_r$, *where* N *is time invariant matrix selecting entries of* ψ_s *from* ψ_r. *Let us consider Dirichlet pdfs* $\prod_{\psi_S \in \psi_S^*} Di_{\lfloor \cdot|\psi_S, S_\Theta}\left({}^{\lfloor \cdot|\psi_S, S}V_0\right)$ *as conjugate prior pdfs for both models* $S \in \{s, r\}$.

Let the initial V*-statistics be related by the nesting mapping*

$$ {}^{\lfloor d|\psi_s, s}V_t = \sum_{\psi_r \in \{\psi_s = N\psi_r\}} {}^{\lfloor d|\psi_r, r}V_t, \; t = 0. \tag{10.31}$$

Then the same property holds for all $t \in t^*$ *if we use Bayes estimation or quasi-Bayes estimation with common data weights for both models.*

The v*-likelihood for the regression vector having structure* $S \in \{s, r\}$ *is*

$$ f(d(\mathring{t})|S) = \prod_{\psi_S \in \psi_S^*} \frac{\mathcal{B}\left({}^{\lfloor \cdot|\psi_S, S}V_{\mathring{t}}\right)}{\mathcal{B}\left({}^{\lfloor \cdot|\psi_S, S}V_0\right)}. \tag{10.32}$$

The multi-variate beta function

$$ \mathcal{B}(v) = \frac{\prod_{i=1}^{\mathring{v}} \Gamma(v_i)}{\Gamma\left(\sum_{i=1}^{\mathring{v}} v_i\right)}$$

is well defined for vectors v *with positive entries* v_i *for which Euler gamma function* Γ *is defined.*

Let $f(s)$ *and* $f(r)$ *be prior probabilities of* s, r. *Then,*

$$ f(s|d(\mathring{t})) = \left(1 + \frac{f(d(\mathring{t})|r) f(r)}{f(d(\mathring{t})|s) f(s)}\right)^{-1}. \tag{10.33}$$

Proof: It is a specialized version of Proposition 3.3.1 and extended by use of Bayes rule on competitive structures $S \in \{s, r\}$. □

The following algorithm uses this proposition for computing posterior probability of all nested structures within a given richest one.

Algorithm 10.6.1 (Estimate of factor-Markov chain structure)
Initial mode

- *Select the structure of a richest regression vector.*
- *Specify the flat prior statistics* ${}^{\lfloor d|\psi_r, r}V_0 > 0$, ${}^{\lfloor d|\psi_r, r}V_0 \approx 0$.
- *Collect the posterior statistics* ${}^{\lfloor d|\psi_r, r}V_{\mathring{t}}$ *using Bayes, batch quasi-Bayes or quasi-Bayes estimation.*
- *Specify prior probabilities* $f(s)$ *of competitive structures, typically, uniform on a priori possible structures.*
- *Define the initial values of* v*-log-likelihood of competitive a priori possible structures* $l_s = \ln(f(s))$.

Structure estimation

1. *Do for all* $\psi_r \in \psi_r^*$.

(a) *Select the structure* s *determining* ψ_s *nested in* ψ_r *and having* $f(s) > 0$; ψ_r *has to be included.*

(b) *Evaluate for* $d \in d^*, t \in \{0, \mathring{t}\}$

$$^{\lfloor d|\psi_s}V_{s;t} = \sum_{\psi_r \in \{\psi_s = N\psi_r\}} {}^{\lfloor d|\psi_r, r}V_t.$$

(c) *Compute increments of the* v*-log-likelihood*

$$l_{s;\psi_s} = \sum_{d \in d^*} \ln \left(\frac{\Gamma\left({}^{\lfloor d|\psi_s, s}V_{\mathring{t}} \right)}{\Gamma\left({}^{\lfloor d|\psi_s, s}V_0 \right)} \right) - \ln \left[\frac{\Gamma\left(\sum_{\tilde{d} \in d^*} {}^{\lfloor \tilde{d}|\psi_s, s}V_{\mathring{t}} \right)}{\Gamma\left(\sum_{\tilde{d} \in d^*} {}^{\lfloor \tilde{d}|\psi_s, s}V_0 \right)} \right].$$

(d) *Update* $l_s = l_s + l_{s;\psi_s}$.

2. *Shift the* v*-log-likelihood* $l_s = l_s - \max_{\{s: f(s) > 0\}} l_s$

3. *Compute posterior probabilities of individual structures* $f(s|d(\mathring{t})) \propto \exp(l_s)$.

4. *Select an estimate of a structure, typically, the MAP estimate.*

Remark(s) 10.6.1

1. *The prior probabilities is an efficient tool for reducing the space, where the best structure is searched for. For instance, it can reflect additional information that arises from smoothness of the originally continuous-valued regression vector, cf. [94].*

2. *Flat prior pdf is chosen in order to violate the assumption on nested prior pdfs only slightly. This condition can be met by a more careful choice of* V_0. *Generally, this problem should be studied but expected practical consequences are of a second order of importance.*

3. *The assumption that updating of various structures is performed with a common weight is the most serious approximation we use. It is enforced by the wish to reduce the need for updating of statistics for all structures that can be generated from* ψ_r.

4. *The initial and terminal values of statistics are sufficient for structure estimation when no forgetting is used. Otherwise, the overall predictor is a product of one-stage ahead predictors that do not cancel in the product. This simple observation is often overlooked.*

Problem 10.6.1 (Connection with Bayesian networks) *Prior information resulting from the underlying dependence knowledge as studied in Bayesian-networks techniques could be used here [93]. Its use is, however, infrequent for the assumed applications in which a detailed inspection is a priori prevented by the problem complexity. It may be important in specific cases.*

10.6.2 Estimation of component structure

Similarly as in the normal case, this task is still unsolved in an efficient algorithmic way. Bayesian-networks technology [93] seems to be the proper direction to be inspected. In the mixed discrete-continuous case of the primary interest, the simple rule "place the Markov factors at the tail of the component" seems to be sufficient.

10.6.3 Merging and cancelling of components

The initiation by the factor splitting is almost always employed for determining the number of components. Usual excessive number of components has to be reduced. Thus, algorithms reducing the *number of components* form important part of mixture estimation. A reduced and tailored set of general algorithms given Section 6.6.4 is presented here. General discussion is skipped and algorithms predicting the values of v-log-likelihood are considered only. Factor-based merging of components and cancelling of components are described.

The formula (6.113) predicts v-likelihood after merging. Algorithm 6.6.8 uses it for a systematic merging of factors. It is specialized here.

Algorithm 10.6.2 (Systematic merging of Markov chain factors)
Initial mode

- *Estimate a mixture with Markov chain factors and conjugate Dirichlet prior pdf. The estimate is described by the collection of occurrence matrices*

$$\{V_{ic;t}\}_{c\in c^*, i=1,\ldots,\mathring{d}, t\in\{0,\mathring{t}\}}.$$

  The factors with the common i *are supposed to describe the same entry of* $d_{i;t}$ *irrespective of the component number.*

- *Initialize the list with rows* $\rho = (i, c, \tilde{c}) \equiv i$*-th factor is common for components* $c, \tilde{c}$*. Usually,* ρ *is initialized as the empty one.*

- *Evaluate the individual normalization factors, see Proposition 10.1.1,* $\ln(\mathcal{I}(V_{ic;t}),\ \forall c \in c^*, i = 1,\ldots,\mathring{d}, t \in \{0,\mathring{t}\}$.

Evaluation mode

For $i = 1,\ldots,\mathring{d}$

Set pointers $c = 1, \tilde{c} = 2$ *to trial components to be merged.*

Test of the common structure

Set the indicator of the common structure $cs = 0$.

Set $cs = -1$ *if the structures of* Θ_{ic} *and* $\Theta_{i\tilde{c}}$ *differ.*

Do if $cs = 0$

Create the trial merger

$\tilde{V}_{i;\mathring{t}} = V_{ic;\mathring{t}} + V_{i\tilde{c};\mathring{t}},\ \tilde{V}_{i;0} = V_{ic;0} + V_{i\tilde{c};0}.$

Evaluate increment $\tilde{l}$ *of the log-v-likelihood*

$\tilde{l} = +\left\{\ln(\mathcal{I}(\tilde{V}_{i;\mathring{t}})) - \ln(\mathcal{I}(V_{ic;\mathring{t}})) - \ln(\mathcal{I}(V_{i\tilde{c};\mathring{t}}))\right\}$

$-\left\{\ln(\mathcal{I}(\tilde{V}_{i;0})) - \ln(\mathcal{I}(V_{ic;0})) - \ln(\mathcal{I}(V_{i\tilde{c};0}))\right\}$

end of the test on $cs = 0$

Do if $\tilde{l} \le 0$ *or* $cs < 0$

Set $\tilde{c} = \tilde{c} + 1$.

Go to the Test of the common structure *if* $\tilde{c} \le \mathring{c}$*. Otherwise continue.*

Set $c = c + 1$ *and* $\tilde{c} = c + 1$.

Go to the beginning of Test of the common structure *if* $c < \mathring{c}$.

Otherwise go to the end of cycle over i.

else replace prior and posterior estimates of factors with indices ic *and* $i\tilde{c}$ *by the trial merger.*

Extend the list of common factors by $\rho = [\rho; (i, c, \tilde{c})]$.

end of the test on improvement of v-likelihood and of $cs < 0$

end of the cycle over i

Merging of components

For $c = 1,\ldots,\mathring{c} - 1$

For $\tilde{c} = c + 1,\ldots,\mathring{c}$

Set $\kappa_{\tilde{c};\mathring{t}} = \kappa_{\tilde{c};\mathring{t}} + \kappa_{c;\mathring{t}}\ \kappa_{\tilde{c};0} = \kappa_{\tilde{c};0} + \kappa_{c;0}$ *and*

cancel the component c *if the components consist of common factors only.*

end of the cycle over $\tilde{c}$

end of the cycle over c

Component cancelling

Cancelling of spurious components is also based on predicting of induced change of v-likelihood. The general approach of Chapters 6, 8 is just specialized here without going into detailed explanations.

Algorithm 10.6.3 (Systematic cancelling of Markov components)
Initial mode

- *Estimate a Markov chain mixture described by the collection of occurrence matrices* $\{V_{ic;t}\}_{c\in c^*, i=1,\ldots,\mathring{d}, t\in\{0,\mathring{t}\}}$ *. The factors with the common* i *are supposed to describe the same entry of* $d_{i;t}$ *irrespective of the component number.*
- *Evaluate the individual normalization factors* $\ln(\mathcal{I}(V_{ic;t}))$, $\forall c \in c^*, i = 1,\ldots,\mathring{d}, t \in \{0,\mathring{t}\}$, *see Proposition 10.1.1.*

Evaluation mode

Do while $c \le \mathring{c}$ *and* $\mathring{c} > 1$

Set $s = s - \kappa_{c;\mathring{t}},\ s_0 = s_0 - \kappa_{c;0}$.

If $\kappa_{c;\mathring{t}} > \kappa_{c;0}$

$$l = \ln\left(\mathcal{B}(s_0, \kappa_{c;0})\right) - \ln\left(\mathcal{B}(s, \kappa_{c;\mathring{t}})\right) - \ln\left(\mathcal{B}(s - s_0, \kappa_{c;\mathring{t}} - \kappa_{c;0})\right)$$

For $i = 1,\ldots,\mathring{d}$

$$l = l + \ln\left(\mathcal{I}(V_{ic;0})\right) - \ln\left(\mathcal{I}(V_{ic;\mathring{t}})\right).$$

end of the cycle over i

else set $l = 1$

end of the test on increase of κ_c

If $l > 0$

Swap c with $\mathring{c}$ *and set* $\mathring{c} = \mathring{c} - 1$, *i.e. cancel the component*

Stop if $\mathring{c} = 1$

else set $s = s + \kappa_{c;\mathring{t}},\ s_0 = s_0 + \kappa_{c;0},\ c = c + 1$

end of the test on v-log-likelihood increase

10.7 Model validation with Markov chain components

This part applies the general model validation presented in Section 6.7 to Markov chain components. The presentation is appropriately reduced.

10.7.1 Learning with labelled data

This part is a reduced counterpart of Section 6.7.1. We deal with the situation when quality of observed behavior is classified ex post and the following hypotheses are formulated:

$H_s \equiv$ The difference in observed consequences is due to the inseparable influence of external conditions and management.

Single mixture describes the standard $d(\mathring{t}_s)$ as well as the labelled $d(\mathring{t}_e)$ data, i.e. for $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$

$$f(d(\mathring{t})|H_s) = \int f(d(\mathring{t})|\Theta_s) f(\Theta_s)\, d\Theta_s. \tag{10.34}$$

$H_e \equiv$ The difference in observed consequences is caused by differences in management. Different mixtures should be used for the standard $d(\mathring{t}_s)$ and the labelled $d(\mathring{t}_e)$ data, i.e. for $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$

$$f(d(\mathring{t})|H_s) = \int f(d(\mathring{t}_s)|\Theta_s) f(\Theta_s)\, d\Theta_s \int f(d(\mathring{t}_e)|\Theta_e) f(\Theta_e)\, d\Theta_e \tag{10.35}$$

with possibly different structures of both mixtures.

With these elements and no prejudice, $f(H_s) = f(H_e)$, the Bayes rule provides the posterior pdf $f(H_s|d(\mathring{t}))$. The common model can be accepted as good one if this probability is high enough.

The test of hypotheses is described by the following algorithm.

Algorithm 10.7.1 (Test of Markov chain data homogeneity)

1. *Run the complete model estimations on the standard $d(\mathring{t}_s)$, labelled $d(\mathring{t}_e)$ and concatenated $d(\mathring{t}) \equiv (d(\mathring{t}_s), d(\mathring{t}_e))$ data. This provides the posterior pdfs*

$$f(\Theta|d(\mathring{t}_\iota)) = Di_\alpha(\kappa_{\mathring{t}_\iota}) \prod_{c\in c^*} \prod_{d\in d^*, \psi\in\psi^*} Di_{\,{}^{\lfloor d|\psi}\Theta_c}\left({}^{\lfloor d|\psi}V_{c;\mathring{t}_\iota}\right),\ \iota \in \iota^* \equiv \{s, e, \emptyset\}.$$

2. *The corresponding v-likelihood values indexed by $\iota \in \iota^*$ are obtained as by product of approximate estimations, see (10.10).*

3. *Determine the probability of that a single standard model should be used*

$$f(standard|d(\mathring{t})) \equiv f(H_s|d(\mathring{t})) = \frac{f(d(\mathring{t}))}{f(d(\mathring{t})) + f(d(\mathring{t}_s))f(d(\mathring{t}_e))}. \tag{10.36}$$

4. *Use the single model further on if $f(standard|d(\mathring{t}))$ is close to 1. The factors that were active on $f(d(\mathring{t}_e))$ denote as potentially dangerous.*

5. *Use both mixtures independently if $f(standard|d(\mathring{t}))$ is close to 0. A danger of arising the situation labelled by e should be signaled whenever the model fitted to $d(\mathring{t}_e)$ makes better predictions than the model fitted to the standard data.*

10.7.2 Learning results and forgetting-based validation

The test on homogeneous data, see Algorithm 10.7.1, can be directly used for model validation if we take $d(\mathring{t}_s)$ as learning data and $d(\mathring{t}_v)$ as validation data unused in learning. Of course, the results of the test have to be re-interpreted appropriately. This test was found rather sharp in normal case and similar behavior is expected in Markov chain case, too. Thus, we expect that the forgetting based validation, Section 6.7.3 will be mostly used. The application of Algorithm 6.7.4 is straightforward so that its specific Markov chain version need no development.

10.7.3 Other indicators of model validity

Other techniques mentioned in Chapter 6, Section 6.7.4 (human inspection of low dimensional projections) or Section 6.7.5 (checking of operating mode) have to be used for model validation. It should be stressed that the list of validation tests is still very much open.

Problem 10.7.1 (Completion of Markov-chain suite) *Algorithms outlined in Chapter 6 are surely applicable to Markov chains. This chapter indicates it clearly. It also shows that a lot of aspects have to be still elaborated in order to get a complete suite of algorithms covering this important class of models.*

Chapter 11

Design with Markov chain mixtures

Markov chain factors allow us to incorporate logical variables and to work with mixed real and discrete valued data. the description of design aspects related to mixtures consisting completely of Markov chains forms the content of Chapter. The mixed case is addressed in Chapter 13.

Discrete nature makes formally the majority of approximations used in connection with the normal case unnecessary as we operate on finite dimensional tables. The design is complex due the dimensionality of Markov chains. Similarly, as in normal case, we restrict the presentation to the case with the state in phase form, Agreement 5.2.1. This fits to the considered applications and decreases the computational load caused by dimensionality.

The restriction to the phase form allows us to exploit, without repetitive references, the relevant notation and relationships introduced in Agreement 9.0.1.

Chapter starts with a brief summary of common tools, Section 11.1. Then, academic, industrial and simultaneous designs are presented in Section 11.2. Concluding Section 11.3 describes strategies supporting interactions with operator.

11.1 Common tools

11.1.1 Model projections in design

Steady-state distribution

Evaluation of the steady state distribution of the observed data helps us to recognizing good and bad factors and components. The large amount of data available justifies the assumption that the uncertainty of parameters is negligible after estimation. It implies that we can evaluate the steady-state distribution of the state vector assuming that parameters Θ are given. For the adopted phase form, the stationary pf $f(\phi_t)$ of $\phi_t = [d_t', \ldots, d_{t-\partial+1}']'$ has to fulfill the identity

$$f(d_t, \ldots, d_{t-\partial+1}) = \sum_{d_{t-\partial}} {}^{\lfloor d_t|\phi_{t-1}}\Theta f(d_{t-1}, \ldots, d_{t-\partial}) = \sum_{d_{t-\partial}} \prod_{i=1}^{\mathring{d}} {}^{\lfloor d_{i;t}|\psi_{i;t}}\Theta f(d_{t-1}, \ldots, d_{t-\partial}). \tag{11.1}$$

Note that inclusion of unit in ϕ_t is superfluous in the Markov chain case.

The full solution of (11.1) is feasible in low dimensional cases only. Often, it is, however, sufficient to get some stationary characteristics, typically, moments. Then, it is reasonable simulate this Markov chain and to estimate the steady-state moments from the observed realization. We rely on ergodicity and, ideally, apply an adapted recursive estimation of steady-state moments with a sequential stopping rule [47].

Problem 11.1.1 (Analysis of the Markov chain with the phase state) *Classification of states of the classical Markov chain is well elaborated and gives a well understood picture of behaving of such dynamic system, see e.g. [26, 95]. Similar picture for higher order Markov chains with the state in the* *phase form would be useful. It does not belong to the standard text-book content and should be collected and if need be complemented.*

Marginal and conditional distributions

Low dimensional marginal and conditional pfs provide the main technical tool for presenting results of the design of the p-system. The algorithmic description of their evaluation is given here for a known (well estimated) Markov chain mixture.

Essentially, Proposition 7.1.2 is specialized here. The specialization is split in Proposition 11.1.2 dealing with marginal pfs and Proposition 11.1.3 addressing the conditioning. Both of them rely on a specific structure of components, Agreement 5.2.1. Thus, we need a tool for changing it.

Proposition 11.1.1 (Permutation of an adjacent pair of Markov chain factors) *Let us consider a pair of adjacent Markov chain factors with known parameters*

$$f(\Delta_1,\Delta_2|\psi_1,\psi_2) = {}^{\lfloor \Delta_1|\Delta_2,\psi_1}\Theta \, {}^{\lfloor \Delta_2|\psi_2}\Theta = {}^{\lfloor \Delta_1|\psi}\tilde{\Theta} \, {}^{\lfloor \Delta_2|\Delta_1,\psi}\tilde{\Theta}, \text{ where} \tag{11.2}$$

Θ *are known transition probabilities,*

$\tilde{\Theta}$ *are transition probabilities after permutation,*

ψ *contains union of entries in* ψ_1,ψ_2*; the entry, which is at least in one of them, is put on the corresponding position of* ψ.

Then, the parameters $\tilde{\Theta}$*, describing the permuted* Δ_1,Δ_2*, are generated by the following algorithm.*

$$\begin{array}{llll}
\textit{For} & \psi \in \psi_1^* \cup \psi_2^* & & \\
\quad \textit{For} & \Delta_1 \in \Delta_1^* & & \\
 & & \textit{Set } \tilde{\Theta}_{\Delta_1|\psi} = 0 & \\
\quad\quad \textit{For} & & \Delta_2 \in \Delta_2^* & \\
 & & \textit{Evaluate the joint probability } \Theta_{\Delta_1,\Delta_2|\psi} \equiv \Theta_{\Delta_1|\Delta_2,\psi}\Theta_{\Delta_2|\psi}, \textit{ where} & \\
 & & \Theta_{\Delta_1|\Delta_2,\psi} = \Theta_{\Delta_1|\Delta_2,\psi_1} \textit{ for } \psi_1 \textit{ included in } \psi \textit{ and } \Theta_{\Delta_2|\psi} = \Theta_{\Delta_2|\psi_2} \textit{ for } \psi_2 \textit{ in } \psi. & \\
 & & \textit{Add } \tilde{\Theta}_{\Delta_1|\psi} = \tilde{\Theta}_{\Delta_1|\psi} + \Theta_{\Delta_1,\Delta_2|\psi}. & \\
\quad\quad \textit{end} & \textit{of the cycle over } \Delta_2 & & \\
\quad\quad \textit{For} & & \Delta_2 \in \Delta_2^* & \\
 & & \textit{Set } \tilde{\Theta}_{\Delta_2|\Delta_1,\psi} = \dfrac{\Theta_{\Delta_1,\Delta_2|\psi}}{\tilde{\Theta}_{\Delta_1|\psi}} & \\
\quad\quad \textit{end} & \textit{of the cycle over } \Delta_2 & & \\
\quad \textit{end} & \textit{of the cycle over } \Delta_1 & & \\
\textit{end} & \textit{of the cycle over } \psi & &
\end{array} \tag{11.3}$$

Proof: The algorithm is directly implied by the alternative expressions for the joint probability $f(x,y) = f(x|y)f(y) = f(y|x)f(x)$. □

Any desired change of the component structure can be performed by such pair-wise permutations.

Proposition 11.1.2 (Marginal predictors for known Markov chain mixtures) *Let us consider the known Markov chain mixture in the factorized form*

$$f(\Delta_t|u_{o;t},d(t-1)) = \sum_{c\in c^*} \alpha_c \prod_{i\in i^*} \Theta_{\Delta_{i;t}|\psi_{ic;t},c}, \quad \text{where}$$

$\psi_{ic;t} = \left[\Delta'_{(i+1)\ldots\mathring{\Delta}c;t}, u'_{o;t}, \phi'_{c;t-1}\right]' \equiv \left[\Delta'_{(i+1)\ldots\mathring{\Delta}c;t}, \psi'_{\mathring{\Delta}c;t}\right]'$ *are regression vectors,* $i \in i^* \equiv \{1,\ldots,\mathring{\Delta}-1\}$, $c \in c^*$,

$\psi'_{\mathring{\Delta}c;t}$ *is the common part of regression vectors forming the component c,*

$\Delta_{i;t}$ *are entries of innovations* Δ_t *and* $u_{o;t}$ *are recognizable o-actions,*

$\Theta_{\Delta_{i;t}|\psi_{ic;t},c} =$ *are known transition probabilities of the c-th component.*

Then, the marginal pf of $\Delta_{\underline{i}\ldots\mathring{\Delta}}$, $\underline{i} \in \{1,\ldots,\mathring{\Delta}\}$, *conditioned on the common part of the regression vector* $\psi_{\mathring{\Delta}} = \left[u'_{o;t}, \phi'_{t-1}\right]'$ *is obtained by omitting leading factors in the above product, i.e.*

$$f(\Delta_{\underline{i}\ldots\mathring{\Delta};t}|u_{o;t}, d(t-1)) = \sum_{c\in c^*} \alpha_c \prod_{i=\underline{i}}^{\mathring{\Delta}} \Theta_{\Delta_{i;t}|\psi_{ic;t},c}.$$

Proof: Omitted. □

For reference purposes, evaluation of conditional pfs is described.

Proposition 11.1.3 (Conditional pfs of Markov chain mixture) *Let us consider the known Markov chain mixture in the factorized form*

$$f(\Delta_t|u_{o;t}, d(t-1)) = \sum_{c\in c^*} \alpha_c \prod_{i\in i^*} \Theta_{\Delta_{i;t}|\psi_{ic;t},c}, \quad \textit{where}$$

$\psi_{ic;t} = \left[\Delta'_{(i+1)\ldots\mathring{\Delta}c;t}, u'_{o;t}, \phi'_{c;t-1}\right]' \equiv \left[\Delta_{(i+1)c;t}, \psi'_{(i+1)c;t}\right]'$ *are regression vectors,* $i \in \{0,1,\ldots,\mathring{\Delta}-1\}$ *with the common part* $\psi_{\mathring{\Delta}c;t} \equiv \left[u'_{oc;t}, \phi'_{c;t-1}\right]'$,

$\Delta_{ic;t}$ *are entries of innovations* $\Delta_{c;t}$, $u_{oc;t}$ *are recognizable o-actions and* $\phi_{c;t-1}$ *are observable states of individual components,*

$\Theta_{\Delta_{i;t}|\psi_{i;t},c_t}$ *are known transition probabilities of the c-th component.*

Let for simplicity the order of factors for all components be common, i.e. $\Delta_{c;t} = \Delta_t$. *Let* $j \in j^* \equiv \{\underline{j},\ldots,\mathring{i}\}$, $\underline{j} \leq \mathring{i}$ *point to selected entries of* Δ_t. *Let* $\underline{k} \in (\underline{j},\mathring{i})$ *point to a subset of the selected entries in* Δ_t.

Then, the predictor of $\Delta_{\underline{j}\ldots\underline{k}-1;t}$ *conditioned on* $u_{o;t}, d(t-1)$ *and* $\Delta_{\underline{k}\ldots\mathring{i};t}$ *is the* ratio of Markov chain mixtures

$$f(\Delta_{\underline{j}\ldots\underline{k}-1;t}|\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)) = \frac{\sum_{c\in c^*} \alpha_c \prod_{j=\underline{j}}^{\mathring{i}} \Theta_{\Delta_{j;t}|\psi_{j;t},c}}{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c}} \prod_{\tilde{j}=\underline{k}}^{\mathring{i}} \Theta_{\Delta_{\tilde{j};t}|\psi_{\tilde{j};t},\tilde{c}}}. \tag{11.4}$$

For a given regression vector, the resulting predictor can be interpreted as the Markov chain mixture

$$\begin{aligned} f(\Delta_{\underline{j}\ldots\underline{k}-1;t}|\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)) &= \sum_{c\in c^*} A_c(\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)) \prod_{j=\underline{j}}^{\underline{k}-1} \Theta_{\Delta_{j;t}|\psi_{j;t},c} \quad \textit{with the weights} \\ A_c(\Delta_{\underline{k}\ldots\mathring{i};t}, u_{o;t}, d(t-1)) &= \frac{\alpha_c \prod_{j=\underline{k}}^{\mathring{i}} \Theta_{\Delta_{j;t}|\psi_{j;t},c}}{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c}} \prod_{j=\underline{k}}^{\mathring{i}} \Theta_{\Delta_{j;t}|\psi_{j;t},\tilde{c}}}. \end{aligned} \tag{11.5}$$

Proof: It is implied by the formula $f(\beta|\gamma) = f(\beta,\gamma)/f(\gamma) = f(\beta,\gamma)/\int f(\beta,\gamma)\,d\beta$. □

Remark(s) 11.1.1

1. *The formula (11.5) confirms that the model obtained through the estimation of the full predictor followed by marginalization is richer than the directly estimated low dimensional predictor. The mixture obtained through conditioning has data-dependent component weights.*

2. *Permutations of factors according to Proposition 11.1.1 have to be made if the considered entries are not at the assumed positions. This need also implies that the assumption* $\Delta_{c;t} = \Delta_t,\ \forall c \in c^*$, *is not restrictive.*

Problem 11.1.2 (Graphical presentation of advices) *General advice "try to reach data configurations having a high probability" applies in the Markov-chain case, too. The adequate and informative graphical representation has to be chosen or developed for the target low-dimensional pfs presented.*

11.1.2 Basic operations for fully probabilistic design

This section prepares basic evaluations needed in connection with the fully probabilistic design.

Proposition 11.1.4 (Expected loss for a factorized Markov chain component) *Let us consider a Markov chain component described by*

$$\begin{aligned}
f(\Delta_t|u_{o;t},\phi_{t-1},\Theta) &= \prod_{i\in i^*}\Theta_{\Delta_{i;t}|\psi_{i;t}} \quad \textit{with} && (11.6)\\
\Theta &\equiv \left\{\Theta_{\Delta_i|\psi_i}\geq 0\right\}_{\Delta_i\in\Delta_i,\psi_i\in\psi_i^*,i=1,\ldots,\mathring{\Delta}}, \ \textit{with finite sets of all involved indices}\\
\psi_{i;t}' &\equiv \left[\Delta_{(i+1)\ldots\mathring{\Delta};t}', u_{o;t}', \phi_{t-1}'\right] \equiv \left[\Delta_{i+1;t}, \psi_{i+1;t}'\right],\ i=0,1,\ldots,\mathring{\Delta}-1,\ \textit{recall}\ \psi_{0;t}\equiv\Psi_t,\\
\psi_{\mathring{\Delta};t}' &\equiv \left[u_{o;t}',\phi_{t-1}'\right].
\end{aligned}$$

Let $\omega_0(\Psi_t)$ *be a non-negative array indexed by* $\Psi_t = \psi_{0;t}$.

Then, the expected value

$$\begin{aligned}
\mathcal{E}[\omega_0(\Psi_t)|u_{o;t},\phi_{t-1}] &\equiv \omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta};t}), \quad \textit{where}\ \omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta};t})\ \textit{is obtained recursively}\\
\omega_i(\psi_{i;t}) &= \sum_{\Delta_{i;t}\in\Delta_i^*}\omega_{i-1}([\Delta_{i;t},\psi_{i;t}])\Theta_{\Delta_{i;t}|\psi_{i;t}}\ \textit{starting from}\ \omega_0(\psi_{0;t}). && (11.7)
\end{aligned}$$

Proof: The expectation is taken over entries of Δ_t as the remaining part of the data vector Ψ_t is fixed by the condition $\psi_{\mathring{\Delta};t} \equiv \left[u_{o;t}', \phi_{t-1}'\right]'$. The chain rule for expectations, Proposition 2.3.6, implies that we can evaluate conditional expectations of individual entries in Δ_t one-by-one starting from the first one. □

When performing fully probabilistic designs, we use the following auxiliary proposition.

Proposition 11.1.5 (The conditional KL distance) *Let* $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t})$ *and*

$$f(\Delta_t|u_{o;t},d(t-1)) = \prod_{i=1}^{\mathring{\Delta}}\Theta_{\Delta_{i;t}|\,{}^{\lfloor l}\psi_{i;t}}, \quad {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},d(t-1)) = \prod_{i=1}^{\mathring{\Delta}_o}{}^{\lfloor U}\Theta_{\Delta_{i;t}|\,{}^{\lfloor U}\psi_{i;t}}.$$

The learnt transition probabilities $\Theta_{\Delta_{i;t}|\,{}^{\lfloor l}\psi_{i;t}}$ *and the user ideal pf* ${}^{\lfloor U}\Theta_{\Delta_{i;t}|\,{}^{\lfloor U}\psi_{i;t}}$ *are extended as follows* $\Theta_{\Delta_{i;t}|\,{}^{\lfloor l}\psi_{i;t}} = \Theta_{\Delta_{i;t}|\psi_{i;t}}$, ${}^{\lfloor U}\Theta_{\Delta_{i;t}|\,{}^{\lfloor U}\psi_{i;t}} = {}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}$, *where* $\psi_{i;t}$ *contains union of entries from respective regression vectors* ${}^{\lfloor l}\psi_{i;t}$, ${}^{\lfloor U}\psi_{i;t}$. *In other words, the regression vectors* $\psi_{i;t}$ *for the* i*-th learnt factor and the corresponding factor of the user ideal pf are common.*

$$\begin{aligned}
\textit{Recall that}\quad \psi_{i;t} &= \left[\Delta_{(i+1)\ldots\mathring{\Delta};t}', u_{o;t}', \phi_{t-1}'\right]' \equiv \left[\Delta_{(i+1);t}, \psi_{i+1;t}'\right]',\ i\in\{0,\ldots,\mathring{\Delta}-1\}\\
\psi_{\mathring{\Delta};t} &= \left[u_{o;t}',\phi_{t-1}'\right]'.\\
\textit{Then,}\ \ \omega(u_{o;t},\phi_{t-1}) &\equiv \sum_{\Delta_t\in\Delta^*} f(\Delta_t|u_{o;t},\phi_{t-1})\ln\left(\frac{f(\Delta_{o;t}|\Delta_{p+;t},u_{o;t},\phi_{t-1})}{{}^{\lfloor U}f(\Delta_{o;t}|u_{o;t},\phi_{t-1})}\right) = \omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta};t}). && (11.8)
\end{aligned}$$

The conditional KL distance $\omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta};t})$ *is found recursively*

$$\omega_i(\psi_{i;t}) = {}^{\lfloor\psi}\omega_{i-1}(\psi_{i;t}) + \sum_{\Delta_{i;t}\in\Delta_i^*}\Theta_{\Delta_{i;t}|\psi_{i;t}}\left[{}^{\lfloor\Delta\Psi}\omega_{i-1}([\Delta_{i;t},\psi_{i;t}]) + \chi\left(i\leq\mathring{\Delta}_o\right)\ln\left(\frac{\Theta_{\Delta_{i;t}|\psi_{i;t}}}{{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}}\right)\right]$$

In the recursion, running for $i = 1, \ldots, \mathring{\Delta}$, *we use the following notation.*

$$\begin{aligned} {}^{\lfloor\psi}\omega_{i-1}(\psi_{i;t}) &= \text{the part of } \omega_{i-1}(\psi_{i-1;t}) \text{ independent of entries } \Delta_{i;t} \\ {}^{\lfloor\Delta\Psi}\omega_{i-1}([\Delta_{i;t}, \psi_{i;t}]) &= \text{the part of } \omega_{i-1}(\psi_{i-1;t}) \text{ that depends also on } \Delta_{i;t} \\ \omega_0(\psi_{0;t-1}) &= 0. \end{aligned}$$

Proof: Let the innovations be split $\Delta = (\Delta_o, \Delta_{p+})$. Then, $\omega(u_{o;t}, \phi_{t-1}) \equiv$

$$\equiv \sum_{\Delta_t \in \Delta^*} f(\Delta_t | u_{o;t}, \phi_{t-1}) \ln \left(\frac{f(\Delta_t | u_{o;t}, d(t-1))}{f(\Delta_{p+;t} | u_{o;t}, d(t-1)) \, {}^{\lfloor U}f(\Delta_{o;t} | u_{o;t}, d(t-1))} \right) =$$

$$= \sum_{\Delta_t \in \Delta^*} \Theta_{\Delta_t | u_{o;t}, \phi_t} \ln \left(\prod_{i=1}^{\mathring{\Delta}_o} \frac{\Theta_{\Delta_{i;t}|\psi_{i;t}}}{{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}} \right) \underbrace{=}_{Proposition\ 2.3.6}$$

$$= \sum_{\Delta_{\mathring{\Delta};t} \in \Delta^*_{\mathring{\Delta};t}} \Theta_{\Delta_{\mathring{\Delta};t}|\psi_{\mathring{\Delta};t}} \left\{ \chi(\mathring{\Delta} \le \mathring{\Delta}_o) \ln \left(\frac{\Theta_{\Delta_{\mathring{\Delta};t}|\psi_{\mathring{\Delta};t}}}{{}^{\lfloor U}\Theta_{\Delta_{\mathring{\Delta};t}|\psi_{\mathring{\Delta};t}}} \right) + \ldots + \right.$$

$$\left. + \underbrace{\sum_{\Delta_{2;t} \in \Delta^*_{2;t}} \Theta_{\Delta_{2;t}|\psi_{2;t}} \left[\chi(2 \le \mathring{\Delta}_o) \ln \left(\frac{\Theta_{\Delta_{2;t}|\psi_{2;t}}}{{}^{\lfloor U}\Theta_{\Delta_{2;t}|\psi_{2;t}}} \right) + \underbrace{\sum_{\Delta_{1;t} \in \Delta^*_{1;t}} \Theta_{\Delta_{1;t}|\psi_{1;t}} \chi(1 \le \mathring{\Delta}_o) \ln \left(\frac{\Theta_{\Delta_{1;t}|\psi_{1;t}}}{{}^{\lfloor U}\Theta_{\Delta_{1;t}|\psi_{1;t}}} \right)}_{\omega_1(\psi_{1;t})} \right]}_{\omega_2(\psi_{2;t})} \right\}.$$

The part of $\omega_{i-1}(\psi_{i-1;t})$ independent of $\Delta_{i;t}$, called ${}^{\lfloor\psi}\omega_i(\psi_{i;t})$, simply adds to the part whose expectation is taken. It consists of the sum part called ${}^{\lfloor\Delta\Psi}\omega_i(\psi_{i-1;t})$ and of the "penalization" term $\chi(i \le \mathring{\Delta}_o) \ln \left(\frac{\Theta_{\Delta_{i;t}|\psi_{i;t}}}{{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}} \right)$. □

While performing the fully probabilistic design, we work with a shifted version of the conditional KL distance. The following slight extension of Proposition 11.1.5 supports the cases for which the Bellman function is given by a table $\omega_\gamma(\phi_t)$.

Proposition 11.1.6 (The shifted conditional KL distance) *Let* $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t})$ *and*

$$f(\Delta_t | u_{o;t}, d(t-1)) = \prod_{i=1}^{\mathring{\Delta}} \Theta_{\Delta_{i;t}|\psi_{i;t}}, \quad {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d(t-1)) = \prod_{i=1}^{\mathring{\Delta}_o} {}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}.$$

The transition probabilities $\Theta_{\Delta_{i;t}|\psi_{i;t}}$ *and the pf* ${}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}$ *describing the user ideal pf are extended, see the formulation of Proposition 11.1.5, so that the regression vectors* $\psi_{i;t}$ *for the* i*-th factor and the corresponding factor of the user ideal pf are common. Recall that*

$$\psi_{i;t} = \left[\Delta'_{(i+1)\ldots\mathring{\Delta};t}, u'_{o;t}, \phi'_{t-1} \right]' \equiv \left[\Delta_{(i+1);t}, \psi'_{i+1;t} \right]', \ i \in \{0, \ldots, \mathring{\Delta} - 1\}, \ \psi_{\mathring{\Delta};t} = \left[u'_{o;t}, \phi'_{t-1} \right]'.$$

Let, moreover, the Bellman function be the table $\omega_\gamma(\phi_t)$ *with*

$$\phi_t \equiv \left[d'_{t\ldots(t-\partial+1)} \right]' \Rightarrow \psi_{0;t} \equiv \Psi_t \equiv \left[\Delta'_t, u'_{o;t}, \phi'_{t-1} \right]'.$$

Then, the shifted KL distance

$$\omega_\gamma(u_{o;t}, \phi_{t-1}) \equiv \sum_{\Delta_t \in \Delta^*} \Theta_{\Delta_t | u_{o;t}, \phi_{t-1}} \left[\ln \left(\frac{\Theta_{\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, \phi_{t-1}}}{{}^{\lfloor U}\Theta_{\Delta_{o;t}|u_{o;t}, \phi_{t-1}}} \right) + \omega_\gamma(\phi_t) \right] \equiv \omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta};t})$$

is found recursively

$$\omega_i(\psi_{i;t}) = {}^{\lfloor\psi}\omega_{i-1}(\psi_{i;t}) + \sum_{\Delta_{i;t} \in \Delta^*_i} \Theta_{\Delta_{i;t}|\psi_{i;t}} \left\{ {}^{\lfloor\Delta\Psi}\omega_{i-1}([\Delta_{i;t}, \psi_{i;t}]) + \chi\left(i \le \mathring{\Delta}_o\right) \ln \left(\frac{\Theta_{\Delta_{i;t}|\psi_{i;t}}}{{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}} \right) \right\}.$$

In the recursion, running for $i = 1, \ldots, \mathring{\Delta}$, *we denoted*

$^{\lfloor\psi}\omega_{i-1}(\psi_{i;t})$ = *the part of* $\omega_{i-1}(\psi_{i-1;t})$ *independent of* $\Delta_{i;t}$

$^{\lfloor\Delta\Psi}\omega_{i-1}([\Delta_{i;t},\psi_{i;t}])$ = *the part of* $\omega_{i-1}(\psi_{i-1;t})$ *that depends also on* $\Delta_{i;t}$.

The recursions start from the following initial condition

$\omega_0(\psi_{0;t}) = \omega_\gamma(\phi_t)$ *for any* $d_{t-\partial}$ *extending* $\phi_t \to \psi_{0;t} = \Psi_t$.

Notice, that the array ω_i *"looses" entries with indices* $\Delta_i, [\Delta_i, \psi_i]$. *On the other hand,* ω_0 *extends* ω_γ *so that the dimensionality loss is compensated.*

Proof: The table $\omega(\phi_t)$ just adds non-trivial initial conditions to the recursion described in Proposition 11.1.5. It arises by extending $\omega(\phi_t)$ so that it can be taken as the function Ψ_t. □

11.1.3 Dangerous components

Discrete and finite nature of the data space modelled by Markov chains imply that there is no such universally dangerous component like unstable one for the normal components. Thus, we have to define as dangerous those component that, when permanently active, lead to a much higher distance to the user ideal pf than the other components, cf. Section 7.2.2.

For a quantitative expression of this statement, we write down the Markov-chain version of Proposition 7.1.3. It evaluates recursively the KL distance of a pair of multi-variate Markov chains $^{\lfloor I}f(d(\mathring{t}))$, $^{\lfloor U}f(d(\mathring{t}))$. The transition matrices $^{\lfloor I}\Theta_{d|\psi}$, $^{\lfloor U}\Theta_{d|\psi}$ generating them

$$^{\lfloor I}f(d(\mathring{t})) = \prod_{t\in t^*}\prod_{i=1}^{\mathring{d}} {}^{\lfloor I\,\lfloor d_{i;t}|\psi_{i;t}}\Theta, \quad {}^{\lfloor U}f(d(\mathring{t})) = \prod_{t\in t^*}\prod_{i=1}^{\mathring{d}} {}^{\lfloor U\,\lfloor d_{i;t}|\psi_{i;t}}\Theta, \tag{11.9}$$

describe a fixed component and the user ideal pf, respectively. We assume that for surplus data of the p-system it holds

$$^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}} = {}^{\lfloor I}\Theta_{\Delta_{i;t}|\psi_{i;t}} \text{ for } i = \mathring{\Delta}_o + 1, \ldots, \mathring{\Delta}. \tag{11.10}$$

The transition probabilities $^{\lfloor I}\Theta_{\Delta_{i;t}|\psi_{i;t}}$ may result from optimization, for instance, of the recognizable actions. For the considered purpose, we need not distinguish innovations and recognizable actions explicitly. Thus, we can assume that $\Delta_t = d_t$.

Proposition 11.1.7 (Recursive evaluation of the KL distance) *Let* $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t}) = d_t$. *Let pfs* $^{\lfloor I}f(d(\mathring{t}))$, $^{\lfloor U}f(d(\mathring{t}))$ *be generated according to (11.9) and fulfill (11.10).*

Then, the KL distance of $^{\lfloor I}f(d(\mathring{t}))$ *and* $^{\lfloor U}f(d(\mathring{t}))$

$$\mathcal{D}\left({}^{\lfloor I}f(d(\mathring{t}))||\,{}^{\lfloor U}f(d(\mathring{t}))\right) \equiv \omega_\gamma(\phi_0) \text{ is obtained recursively for } t = \mathring{t}, \mathring{t}-1, \ldots, 1, \text{ starting at } \omega_\gamma(\phi_{\mathring{t}}) = 0 \tag{11.11}$$

$$\omega_\gamma(\phi_{t-1}) \equiv \sum_{\Delta_t\in\Delta^*}\prod_{i=1}^{\mathring{\Delta}} {}^{\lfloor I}\Theta_{\Delta_{i;t}|\psi_{i;t}}\left[\ln\left(\prod_{j=1}^{\mathring{\Delta}}\frac{^{\lfloor I}\Theta_{\Delta_{j;t}|\psi_{j;t}}}{^{\lfloor U}\Theta_{\Delta_{j;t}|\psi_{j;t}}}\right) + \omega_\gamma(\phi_t)\right] \equiv \omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta};t})$$

is found recursively (11.12)

$$\omega_i(\psi_{i;t}) = {}^{\lfloor\psi}\omega_{i-1}(\psi_{i;t}) + \sum_{\Delta_{i;t}\in\Delta_i^*}\Theta_{\Delta_{i;t}|\psi_{i;t}}\left\{{}^{\lfloor\Delta\Psi}\omega_{i-1}([\Delta_{i;t},\psi_{i;t}]) + \chi\left(i\le\mathring{\Delta}_o\right)\ln\left(\frac{\Theta_{\Delta_{i;t}|\psi_{i;t}}}{^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}}\right)\right\}.$$

In the recursion, running for $i = 1, \ldots, \mathring{\Delta}$, *we denoted*

$^{\lfloor\psi}\omega_{i-1}(\psi_{i;t})$ = *the part of* $\omega_{i-1}(\psi_{i-1;t})$ *independent of* $\Delta_{i;t}$,

$^{\lfloor\Delta\Psi}\omega_{i-1}([\Delta_{i;t},\psi_{i;t}])$ = *the part of* $\omega_{i-1}(\psi_{i-1;t})$ *that depends also on* $\Delta_{i;t}$.

The recursions start from the following initial condition

$\omega_0(\psi_{0;t}) = \omega_\gamma(\phi_t)$ *for any* $d_{t-\partial}$ *extending* $\phi_t \to \psi_{0;t} = \Psi_t$.

Proof: It combines directly Propositions 7.1.3, 11.1.6. □

Remark(s) 11.1.2

1. *The index γ is used in order to keep continuity with the notation used in Chapters 7, 9.*
2. *The recursion (11.12) is linear with respect to the array $\omega_\gamma(\cdot)$.*
3. *It makes sense to consider stationary version of (11.12), i.e. the case with $\mathring{t} \to \infty$. It is closely related to the average KL distance per step, i.e. $\lim_{\mathring{t}\to\infty} \frac{1}{\mathring{t}} \mathcal{D}\left({}^{\lfloor I}f(d(\mathring{t})) || {}^{\lfloor U}f(d(\mathring{t}))\right)$. The full art the Markov chain decision processes can be used for its analysis, [26, 95]. Even numerical techniques elaborated in this area, e.g. [96], should be used for its evaluation and optimization.*

11.2 Design of advising strategy

Particular design versions are based on a straightforward application of the fully probabilistic design with the special target resulting from difference of sets d_o^* and d_p^*, Proposition 7.1.4. Influence of advices on the inter-connection of the p- and o- systems is modelled as presented in Section 7.1.3.

11.2.1 Academic design

The recommended pointers c_t to components are the only actions of the academic p-system. They are generated by the optimized academic strategy $\left\{ {}^{\lfloor I}f(c_t|d(t-1))\right\}_{t\in t^*}$. It determines, together with the estimated Markov chain mixture of the o-system, the ideal pfs to be presented to the operator

$$ {}^{\lfloor I}f(d_t, c_t|d(t-1)) \equiv \prod_{i=1}^{\mathring{d}} \Theta_{d_{i;t}|\psi_{ic_t;t},c_t} \, {}^{\lfloor I}f(c_t|d(t-1)) \text{ with } d_t \equiv \Delta_t, \ \mathring{d} = \mathring{\Delta}. \tag{11.13} $$

The user ideal pf needed for the fully probabilistic design is selected as a Markov chain on the o-data $d_{o;t}^*$. This true user ideal pf is extended on d_t^* in the way discussed in Section 5.1.3. It is extended further on c_t^* by specifying the target pf ${}^{\lfloor U}\Theta_{c_t|\phi_{t-1}}$ for the academic advices. Altogether,

$$ {}^{\lfloor U}f(d_t, c_t|d(t-1)) = \prod_{i=1}^{\mathring{d}_o} {}^{\lfloor U \lfloor d_{i;t}|\psi_{i;t}}\Theta \, {}^{\lfloor I}f(d_{p+;t}|c_t, d(t-1)) \, {}^{\lfloor U}\Theta_{c_t|\phi_{t-1}}. $$

In evaluations, we use the freedom in the choice of the ideal pf ${}^{\lfloor U}f(c_t|d(t-1))$ for the actions c_t of the constructed academic p-system. Always, we evaluate initially dangerous components, see Section 11.1.3, and reduce the support of ${}^{\lfloor U}f(c_t|d(t-1))$ to non-dangerous ones. If the reduced set contains just a single component, the optimal design of the academic p-system is solved by this choice and the component is presented to the operator as the designed ideal pf. Otherwise, the Markov chain version of the optimal academic advising strategy described by Proposition 7.2.1 has to be searched for.

Proposition 11.2.1 (Academic advisory system minimizing the KL distance) *Let us consider the design of the academic advisory system for the o-system described by the mixture with Markov chain components having the state ϕ_t in the phase form. The innovation Δ_t and data record d_t coincide as no recognizable actions are available. Let $d_t = (d_{o;t}, d_{p+;t})$ =(o-data, surplus p-data) and the user ideal pf ${}^{\lfloor U}f(d(\mathring{t}))$ on $d^*(\mathring{t})$ be generated by*

$$ {}^{\lfloor U}f(d_t|d(t-1)) \equiv \prod_{i_o\in i_o^*} {}^{\lfloor U \lfloor d_{i;t}|\psi_{i;t}}\Theta \, {}^{\lfloor I}f(d_{p+;t}|d(t-1)), \ i_o^* \equiv \left\{1,\ldots,\mathring{d}_o\right\} \subset i^* \equiv \left\{1,\ldots,\mathring{d}\right\}. \tag{11.14} $$

The parameters $\Theta_{d_{i;t}|\psi_{i;t},c} \geq 0$, $\sum_{d_i\in d_i^*} \Theta_{d_i|\psi_{ic}} = 1$ of the Markov chain mixture as well those determining the Markov chain user ideal pf are assumed to be known. The probabilities $\Theta_{d_{i;t}|\psi_{i;t},c}$, ${}^{\lfloor U \lfloor d_{i;t}|\psi_{i;t}}\Theta$ are extended, cf. Proposition 11.1.5, so that the corresponding factors of the user ideal and the Markov chain mixture have common regression vectors $\psi_{i;t} \equiv \left[d_{i+1;t}, \psi'_{i+1;t}\right]' = \left[d'_{(i+1)\ldots\mathring{d};t}, \phi'_{t-1}\right]'$, $i < \mathring{d}$, $\psi_{\mathring{d};t} \equiv \phi_{t-1}$.

The recommended pointers c_t are allowed to have non-zero values at most for those indices in c^* that point to non-dangerous components, Section 11.1.3.

Then, the optimal causal academic advisory strategy – minimizing the KL distance of $^{\lfloor I}f(d(\mathring{t}), c(\mathring{t}))$ *to the user ideal* $^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) = \prod_{t\in t^*} {}^{\lfloor U}f(d_t|d(t-1))\, {}^{\lfloor U}f(c_t|d(t-1))$ *– is described by the following formulas initialized by* $\omega_\gamma(\phi_{\mathring{t}}) = 0, \ \ \forall \phi_{\mathring{t}} \in \phi^*$.

For $t = \mathring{t}, \ldots, 1$

For $\phi_{t-1} \in \phi^*$

Set $\omega_\gamma(\phi_{t-1}) = 0$

For $c_t = 1, \ldots, \mathring{c}$

For $d_t = 1, \ldots, \mathring{d}$

$$\textit{Set} \ \ \Psi_t \equiv \psi_{0;t} = [d_t', \phi_{t-1}']' \equiv [\phi_t', d_{t-\partial}']' \tag{11.15}$$

$\omega_0(c_t, \psi_{0;t}) \equiv \omega_0(\psi_{0;t}) = \omega_\gamma(\phi_t)$ *for any* $d_{t-\partial}$ *extending* $\phi_t \to \psi_{0;t} = \Psi_t$.

end of the cycle over d_t

For $i = 1, \ldots, \mathring{d}$

$\omega_i(c_t, \psi_{i;t}) = {}^{\lfloor \psi}\omega_{i-1}(c_t, \psi_{i;t})$

For $d_{i;t} = 1, \ldots, \mathring{d}_i$

$$\omega_i(c_t, \psi_{i;t}) = \omega_i(c_t, \psi_{i;t}) +$$
$$+\Theta_{\Delta_{i;t}|\psi_{i;t},c_t} \left[{}^{\lfloor d\Psi}\omega_{i-1}(c_t, [d_{i;t}, \psi_{i;t}]) + \chi\left(i \le \mathring{d}_o\right) \ln\left(\frac{\Theta_{d_{i;t}|\psi_{i;t},c_t}}{{}^{\lfloor U}\,{}^{\lfloor d_{i;t}|\psi_{i;t}}\Theta} \right) \right]$$

end of the cycle over $d_{i;t}$

In the above recursion, we use

${}^{\lfloor \psi}\omega_{i-1}(c_t, \psi_{i;t}) =$ *the part of* $\omega_{i-1}(c_t, \psi_{i-1;t})$ *independent of* $d_{i;t}$,

${}^{\lfloor d\Psi}\omega_{i-1}(c_t, [d_{i;t}, \psi_{i;t}]) =$ *the part of* $\omega_{i-1}(c_t, \psi_{i-1;t})$ *that depends also on* $d_{i;t}$.

end of the cycle over i

Set ${}^{\lfloor I}f(c_t|\phi_{t-1}) = {}^{\lfloor U}\Theta_{c_t|\phi_{t-1}} \exp[-\omega_{\mathring{d}}(c_t, \phi_{t-1})]$,

$\omega_\gamma(\phi_{t-1}) = \omega_\gamma(\phi_{t-1}) + {}^{\lfloor I}f(c_t|\phi_{t-1})$.

end of the cycle over c_t

For $c_t = 1, \ldots, \mathring{c}$

$$^{\lfloor I}f(c_t|\phi_{t-1}) = \frac{^{\lfloor I}f(c_t|\phi_{t-1})}{\omega_\gamma(\phi_{t-1})}, \quad \omega_\gamma(\phi_{t-1}) = -\ln(\omega_\gamma(\phi_{t-1}))$$

end of the cycle over c_t

end of the cycle over ϕ_{t-1}

end of the cycle over t

Proof: We show that the Bellman function is the table $\omega_\gamma(\phi_t)$. For $t = \mathring{t}$, it holds with $\omega(\phi_{\mathring{t}}) = 0$. Backward induction implies that for a generic t

$$\begin{aligned} \omega_\gamma(\phi_{t-1}) &= \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1}) \left[\ln\left(\frac{^{\lfloor I}f(c_t|\phi_{t-1})}{^{\lfloor U}f(c_t|\phi_{t-1})} \right) + \omega_\gamma(c_t, \phi_{t-1}) \right], \quad \text{with} \\ \omega_\gamma(c_t, \phi_{t-1}) &\equiv \sum_{d_t\in d^*} f(d_t|\phi_{t-1}, c_t) \left[\ln\left(\frac{f(d_t|\phi_{t-1}, c_t)}{^{\lfloor U}f(d_t|\phi_{t-1})} \right) + \omega_\gamma(\phi_t) \right]. \end{aligned}$$

It implies the form of the optimizing strategy with the achieved minimum being the table

$$\omega_\gamma(\phi_{t-1}) = -\ln\left[\sum_{c_t\in c^*} {}^{\lfloor U}f(c_t|\phi_{t-1}) \exp[-\omega_\gamma(c_t, \phi_{t-1})] \right].$$

The evaluation of the shifted KL distance is prepared in Proposition 11.1.6. □

The proved proposition combined with the certainty-equivalent strategy gives the following algorithm.

Algorithm 11.2.1 (Fixed academic advisory system minimizing the KL distance)
Initial (off line) mode

- *Estimate Markov chain mixture describing the o-system with the phase state ϕ_t, Chapter 10.*
- *Exclude dangerous components, Sections 11.1.3, 11.1.1.*
- *Return to the learning phase if all components are dangerous, i.e. if $\mathring{c} = 0$.*
- *Take the non-dangerous component as the ideal pf offered to the user and stop if $\mathring{c} = 1$.*
- *Specify the true user ideal pf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{d}_o} {}^{\lfloor U}{}^{\lfloor d_{i;t}|\psi_{i;t}}\Theta.$$

- *Specify* time invariant *user ideal pf $^{\lfloor U}f(c_t|\phi_{t-1}) = {}^{\lfloor U}\Theta_{c_t|\phi_{t-1}}$ on the recommended pointers. It is zero on dangerous components.*
- *Select the length of the design horizon $\mathring{t} \geq 1$.*
- *Initialize the iterative mode by setting $\omega_\gamma(\phi_{\mathring{t}}) = 0$.*

Iterative (off line) mode

- *Correct the arrays, for $t = \mathring{t}, \ldots, 1$, as given in (11.15).*
- *Store the terminal characteristics $\omega_\gamma(c,\phi)$ $c \in c^*$, $\phi \in \phi^*$ determining the optimal* steady-state strategy*, Chapter 3.*

Sequential (on line) mode*, running for $t = 1, 2, \ldots$,*

1. *Acquire the data record d_t and determine the state vector ϕ_t.*
2. *Evaluate the values of the pf describing the optimal academic advising strategy*

$$^{\lfloor I}f(c_{t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}|\phi_t)\exp[-\omega_\gamma(c_{t+1},\phi_t)].$$

3. *Present to the user projections of the ideal pf*

$$^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1}\in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) \prod_{i=1}^{\mathring{d}} \Theta_{d_{ic_{t+1};t+1}|\psi_{i;t+1}}.$$

4. *Go to the beginning of* Sequential mode.

The adaptive version of the academic design with certainty-equivalence strategy and IST patch is described by the following algorithm.

Algorithm 11.2.2 (Adaptive academic advisory system minimizing the KL distance)
Initial (off line) mode

- *Estimate Markov chain mixture describing the o-system with the phase state ϕ_t, Chapter 10.*
- *Specify the true user ideal pf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{d}_o} {}^{\lfloor U}{}^{\lfloor d_{i;t}|\psi_{i;t}}\Theta.$$

- *Specify <u>time invariant</u> user ideal pf* $^{\lfloor U}f(c_t|\phi_{t-1}) = {}^{\lfloor U}\Theta_{c_t|\phi_{t-1}}$ *on the recommended pointers.*
- *Select the length of the design horizon* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Complement state vector* ϕ_t *by newly acquired data* d_t *and construct the data vector* $\Psi_t = [d_t', \phi_{t-1}']'$.
2. *Update the estimates of the Markov mixture using quasi-Bayes estimation, Algorithm 6.5.2.*
3. *Initialize the iterative mode by setting* $\omega_\gamma(\phi_{\mathring{t}}) = 0$. *Skip this step if* $t > 1$ *and IST patch is used.*
4. *Correct iteratively the arrays* ω_γ *as given in (11.15) with* τ *in the role* t, $\tau = t+T, \ldots, t+1$.
5. *Evaluate the values of the pf describing the optimal academic advising strategy*
$$^{\lfloor I}f(c_{t+1}|\phi_t) \propto {}^{\lfloor U}f(c_{t+1}|\phi_t)\exp[-\omega_\gamma(c_{t+1}, \phi_t)].$$
6. *Present to the user projections of the ideal pf*
$$^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1}\in c^*} {}^{\lfloor I}f(c_{t+1}|\phi_t) \prod_{i=1}^{\mathring{d}} \Theta_{d_{ic_{t+1};t+1}|\psi_{i;t+1}}.$$
7. *Go to the beginning of* Sequential mode.

Remark(s) 11.2.1

1. *The strategy with maximum probable advices can simply be derived by copying general considerations presented in Chapter 7.*
2. *The strategy with grouped advices can also be simply derived by copying results presented in Chapter 7.*
3. *Design and application of the academic advising is simple. Manipulations with high-dimensional arrays are the most complex and restrictive operations. Current technology allows, however, handle rather extensive memories. It is important, especially in on line mode, when read-only operations dominate.*

11.2.2 Industrial design

The industrial design is used whenever component weights have objective meaning and the operator has no influence on them. It is addressed here.

Similarly, as in previous Chapters, the following order of entries in the data record d_t is supposed $d_t = (\Delta_{o;t}, \Delta_{p+;t}, u_{o;t}) \equiv$ (o-innovations, surplus p-innovations, recognizable actions). The Markov chain components with explicitly marked recognizable actions

$$\left\{ f(\Delta_t|u_{o;t}, d(t-1), c) f(u_{o;t}|d(t-1), c) \equiv \prod_{i=1}^{\mathring{\Delta}} \Theta_{\Delta_{i;t}|\psi_{ic;t}} \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{ic;t}} \right\}_{t\in t^*, c\in c^*} \tag{11.16}$$

and their weights $\{\alpha_c\}_{c\in c^*}$ are assumed to be known (well estimated).

The considered extension, see Section 5.1.3, of the true user ideal pf is

$$^{\lfloor U}f(d(\mathring{t})) \equiv \prod_{t\in t^*} \prod_{i=1}^{\mathring{\Delta}_o} {}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}} \prod_{i=\mathring{\Delta}_o+1}^{\mathring{\Delta}} {}^{\lfloor I}f(\Delta_{i;t}|\psi_{i;t}) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} {}^{\lfloor U}\Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{i;t}}. \tag{11.17}$$

Unlike in normal case, Chapter 9, the KL distance can be and is optimized.

Proposition 11.2.2 (Optimal industrial design for Markov chain mixtures) *Let the joint pf*

$$^{\lfloor I}f(\Delta(\mathring{t}), u_o(\mathring{t})) \equiv \prod_{t \in t^*} \frac{\sum_{c \in c^*} \alpha_c f(\Delta_t | u_{o;t}, d(t-1), c) f(u_{o;t}|d(t-1), c)}{\sum_{c \in c^*} \alpha_c f(u_{o;t}|d(t-1), c)} \, ^{\lfloor I}f(u_{o;t}|d(t-1))$$

with components (11.16) be determined by the optional industrial advising strategy described by pfs $\left\{ ^{\lfloor I}f(u_{o;t}|d(t-1)) \right\}_{t \in t^*}$.

Then, the optimal strategy, minimizing the KL distance $\mathcal{D}\left(^{\lfloor I}f || ^{\lfloor U}f \right)$ *to the user ideal pf (11.17), is generated by the following algorithm, initialized by* $\omega_\gamma(\phi_{\mathring{t}}) = 0, \quad \forall \phi_{\mathring{t}} \in \phi^*$.

For $t = \mathring{t}, \ldots, 1$

For $\phi_{t-1} = 1, \ldots, \mathring{\phi}$

$$\text{Set } \psi_{\mathring{d};t} = \phi_{t-1}, \quad \omega_\gamma(\phi_{t-1}) = 0 \tag{11.18}$$

For $i = \mathring{d}, \ldots, 1$

For $d_{i;t} = 1, \ldots, \mathring{d}_i$

$$\text{Set } \psi_{i-1;t} \equiv [d_{i;t}, \psi'_{i;t}], \quad \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} = 0.$$

For $c_t = 1, \ldots, \mathring{c}$

$$\text{Set } \Theta_{d_{i\_\mathring{d}}|\psi_{i;t}, c_t} = 1$$

For $j = i, \ldots, \mathring{d}$

$$\Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}, c_t} = \Theta_{d_{i\_\mathring{d}}|\psi_{i;t}, c_t} \Theta_{d_{j;t}|\psi_{j;t}, c_t}$$

end of the cycle over j

$$\Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} = \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} + \alpha_{c_t} \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}, c_t}.$$

end of the cycle over c_t

end of the cycle over $d_{i;t}$

end of the cycle over i

For $u_{o;t} = 1, \ldots, \mathring{u}_o$

For $\Delta_t = 1, \ldots, \mathring{\Delta}$

Set $\omega_0(\psi_{0;t}) \equiv \omega_\gamma(\phi_t)$ *for any* $d_{t-\partial}$ *extending* $\phi_t \rightarrow \psi_{0;t} = \Psi_t$.

end of the cycle over Δ_t

For $i = 1, \ldots, \mathring{\Delta}$

$$\omega_i(\psi_{i;t}) = \, ^{\lfloor \psi}\omega_{i-1}(\psi_{i;t})$$

For $\Delta_{i;t} = 1, \ldots, \mathring{\Delta}_i$

$$\omega_i(\psi_{i;t}) = \omega_i(\psi_{i;t}) + \frac{\Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}}}{\Theta_{d_{(i+1)\_\mathring{d};t}|\psi_{i;t}}} \times$$

$$\times \left[^{\lfloor \Delta\Psi}\omega_{i-1}([\Delta_{i;t}, \psi_{i;t}]) + \chi\left(i \leq \mathring{\Delta}_o\right) \ln\left(\frac{\Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}}}{\Theta_{d_{(i+1)\_\mathring{d};t}|\psi_{i;t}} \, ^{\lfloor U \lfloor d_{i;t}|\psi_{i;t}}\Theta} \right) \right].$$

end of the cycle over $\Delta_{i;t}$

In the above recursion, we use

$^{\lfloor \psi}\omega_{i-1}(\psi_{i;t}) =$ *the part of* $\omega_{i-1}(\psi_{i-1;t})$ *independent of entries* $\Delta_{i;t}$

$^{\lfloor \Delta\Psi}\omega_{i-1}([\Delta_{i;t}, \psi_{i;t}]) =$ *the part of* $\omega_{i-1}(c_t, \psi_{i-1;t})$ *that depends also on* $\Delta_{i;t}$

end of the cycle over i

$$\text{Set } \, ^{\lfloor I}f(u_{o;t}|\phi_{t-1}) = \, ^{\lfloor U}\Theta_{u_{o;t}|\phi_{t-1}} \exp[-\omega_{\mathring{\Delta}}(\psi_{\mathring{\Delta}})],$$

$$\omega_\gamma(\phi_{t-1}) = \omega_\gamma(\phi_{t-1}) + \, ^{\lfloor I}f(u_{o;t}|\phi_{t-1}).$$

end of the cycle over $u_{o;t}$

For $u_{o;t} = 1, \ldots, \mathring{u}_o$

$$^{\lfloor I}f(u_{o;t}|\phi_{t-1}) = \frac{^{\lfloor I}f(u_{o;t}|\phi_{t-1})}{\omega_\gamma(\phi_{t-1})}, \quad \omega_\gamma(\phi_{t-1}) = -\ln(\omega_\gamma(\phi_{t-1})).$$

end of the cycle over $u_{o;t}$

end of the cycle over ϕ_{t-1}

end of the cycle over t

Proof: We apply Proposition 7.1.4 with Bellman function in the form of the table $\omega_\gamma(\phi_t)$. Its initial value is zero. The optimal strategy has the form

$$\begin{aligned}
^{\lfloor I}f(u_{o;t}|\phi_{t-1}) &= {}^{\lfloor U}\Theta_{u_t|\phi_{t-1}} \frac{\exp[-\omega_\gamma(u_{o;t},\phi_{t-1})]}{\gamma(\phi_{t-1})} \quad \text{with} \\
\gamma(\phi_{t-1}) &\equiv \sum_{u_{o;t}\in u_o^*} {}^{\lfloor U}\Theta_{u_t|\phi_{t-1}} \exp[-\omega_\gamma(u_{o;t},\phi_{t-1})], \\
\omega_\gamma(\phi_{t-1}) &= -\ln(\gamma(\phi_{t-1})).
\end{aligned}$$

The decisive shifted conditional KL distance has the form

$$\begin{aligned}
\omega_\gamma(u_{o;t},\phi_{t-1}) &= \sum_{\Delta_t\in\Delta^*} \frac{\sum_{c_t\in c^*}\alpha_c\prod_{i=1}^{\mathring{d}}\Theta_{d_{i;t}|\psi_{i;t},c_t}}{\sum_{\tilde{c}_t\in c^*}\alpha_{\tilde{c}}\prod_{i=\mathring{\Delta}+1}^{\mathring{d}}\Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{i;t},\tilde{c}_t}} \times \\
&\times \left[\ln\left(\frac{\sum_{c_t\in c^*}\alpha_c\prod_{i=1}^{\mathring{d}}\Theta_{d_{i;t}|\psi_{i;t},c_t}}{\sum_{\tilde{c}_t\in c^*}\left[\alpha_{\tilde{c}}\prod_{i=\mathring{\Delta}_o+1}^{\mathring{d}}\Theta_{d_{i;t}|\psi_{i;t},\tilde{c}_t}\right]\prod_{i=1}^{\mathring{\Delta}_o}{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}}\right) + \omega_\gamma(\phi_t)\right].
\end{aligned}$$

Similarly as in Proposition 11.1.6, its evaluation is done recursively using chain rule for expectations, Proposition 2.3.6. Conditional and marginal pfs are evaluated according to Propositions 11.1.2, 11.1.3. For $i = 1,\ldots,\mathring{\Delta}$, it holds

$$\begin{aligned}
\omega_i(c_t,\psi_{i;t}) &= {}^{\lfloor\psi}\omega_{i-1}(c_t,\psi_{i;t}) + \sum_{\Delta_{i;t}\in\Delta_i^*} \frac{\sum_{c_t\in c^*}\alpha_c\prod_{j=i}^{\mathring{d}}\Theta_{d_{j;t}|\psi_{j;t},c_t}}{\sum_{\tilde{c}_t\in c^*}\alpha_{\tilde{c}}\prod_{j=i+1}^{\mathring{d}}\Theta_{d_{j;t}|\psi_{j;t},\tilde{c}_t}} \times \\
&\times \left[{}^{\lfloor\Delta\Psi}\omega(c_t,[\Delta_{i;t},\psi_{i;t}]) + \chi\left(i\le\mathring{\Delta}_o\right)\ln\left(\frac{\sum_{c_t\in c^*}\alpha_c\prod_{j=i}^{\mathring{d}}\Theta_{d_{j;t}|\psi_{j;t},c_t}}{\sum_{\tilde{c}_t\in c^*}\alpha_{\tilde{c}}\prod_{j=i+1}^{\mathring{d}}\Theta_{d_{j;t}|\psi_{j;t},\tilde{c}_t}\,{}^{\lfloor U}{}^{\lfloor d_{i;t}|\psi_{i;t}}\Theta}\right)\right].
\end{aligned}$$

Initial value of this recursion is obtained by extending $\omega_\gamma(\phi_t)$ to $\omega_0(\psi_{0;t})$ by taking them as equal whenever ϕ_t is a part of $\psi_{0;t}$. □

The proved proposition, combined with the receding-horizon certainty-equivalent strategy, justifies the following algorithm that adds just the estimation environment to the design algorithm.

Algorithm 11.2.3 (Optimal industrial design for Markov chain mixtures)
Initial (off line) mode

- *Estimate the Markov chain mixture describing the o-system with the state* ϕ_t *in the phase form, Chapter 10.*
- *Specify the true user ideal pf (11.17) on the response of the o-system.*
- *Select the length of the receding horizon* $T \geq 1$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Update estimates of the model parameters, Section 10.5, if you deal with the adaptive advisory system.*

3. *Initialize the iterative mode by setting $\tau = t + T$ and $\omega_\gamma(\phi_\tau) = 0$. Omit the zeroing of ω_γ if $t > 1$ and the IST strategy is used.*

Iterative mode

- *Apply algorithm given in Proposition 11.2.2 while replacing t by τ and stopping at $\tau = t + 1$.*

4. *Evaluate the pf ${}^{\lfloor I}f(u_{o;t+1}|\phi_t)$ resulting from this industrial design.*

5. *Present to the user projections of the ideal pf*

$$
{}^{\lfloor I}f(d_{t+1}|\phi_t) = \frac{\sum_{c_{t+1}\in c^*} \alpha_{c_{t+1}} \prod_{i=1}^{\mathring{d}} \Theta_{d_{i;t+1}|\psi_{i;t+1},c_{t+1}}}{\sum_{c_{t+1}\in c^*} \alpha_{c_{t+1}} \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \Theta_{d_{i;t+1}|\psi_{i;t+1},c_{t+1}}} \, {}^{\lfloor I}f(u_{o;t+1}|\phi_t).
$$

6. *Go to the beginning of* Sequential mode.

Remark(s) 11.2.2

The grouped version of the industrial design is to be implemented, see Proposition 7.2.4 and Remark 5.

11.2.3 Simultaneous academic and industrial design

The simultaneous academic and industrial design provides the best problem formulation and solution. The academic actions of the simultaneous p-system $c_t \in c^* \equiv \{1, \ldots, \mathring{c}\}$ are generated by a causal strategy $d^*(t-1) \to c^*$. The industrial part generates the recommended recognizable actions $d^*(t-1) \to u^*_{o;t}$. The potential ideal pfs are

$$
\begin{aligned}
&{}^{\lfloor I}f(d_t, c_t|d(t-1)) = f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t) \times \qquad (11.19)\\
&\quad \times \; f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1)). \text{ The user ideal pf is}\\
&{}^{\lfloor U}f(d_t, c_t|d(t-1)) = {}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1)) \times\\
&\quad \times \; f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t)\, {}^{\lfloor U}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor U}f(u_{o;t}|d_o(t-1)) \propto\\
&\quad \propto \prod_{i=1}^{\mathring{\Delta}_o} \left[{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}} \right] f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t) \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} \left[{}^{\lfloor U}\Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{i;t}} \right] {}^{\lfloor U}\Theta_{c_t|\psi_t}, \quad \text{where}
\end{aligned}
$$

$f(\Delta_{o;t}|\Delta_{p+;t}, u_{o;t}, d(t-1), c_t) \equiv \prod_{i=1}^{\mathring{\Delta}_o} \Theta_{\Delta_{ic_t;t}|\psi_{ic_t;t}}$ is the pf derived from the c_t-th learnt component describing the o-innovations;

$f(\Delta_{p+;t}|u_{o;t}, d(t-1), c_t) \equiv \prod_{i=\mathring{\Delta}_o+1}^{\mathring{\Delta}} \Theta_{\Delta_{i;t}|\psi_{ic_t;t}}$ is the pf also derived from the c_t-th learnt component and describing the surplus p-innovations;

$\left\{ {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1)) \equiv {}^{\lfloor I}f(c_t|u_{o;t}, d(t-1))\, {}^{\lfloor I}f(u_{o;t}|d(t-1)) \right\}_{t\in t^*}$ is the optimized simultaneous strategy;

${}^{\lfloor U}f(\Delta_{o;t}|u_{o;t}, d_o(t-1)) = \prod_{i=1}^{\mathring{\Delta}_o} {}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}$ is the true user ideal pf on the o-innovations, see Section 5.1.3;

${}^{\lfloor U}f(u_{o;t}|d_o(t-1)) = \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} {}^{\lfloor U}\Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{i;t}}$ is the true user ideal pf on the recognizable actions $u_{o;t}$, see Section 5.1.3;

${}^{\lfloor U}f(c_t|u_{o;t}, d(t-1)) = {}^{\lfloor U}\Theta_{c_t|\psi_t}$ is the pf representing the optional knob of the p-system that can respect special requirements like stability of advices. Notice that the desirable c_t may depend on $u_{o;t}$.

Proposition 11.2.3 (Simultaneous advisory system minimizing the KL distance) *Let us consider the simultaneous academic and industrial design for the o-system described by the Markov chain mixture with the state ϕ in the phase form. The data record d_t contains both innovations $\Delta_t = (\Delta_{o;t}, \Delta_{p+;t}) =$ (innovations in d_o^*, innovations in d_{p+}^*) and recognizable actions $u_{o;t}$. The o-data $d_{o;t} = (\Delta_{o;t}, u_{o;t}) =$ (o-innovations, recognizable actions). The data record is ordered as follows $d = (\Delta_o, \Delta_{p+}, u_o)$.*

The assumed influence of advices and the user ideal pf are described by (11.19). The parameters $\Theta_{d_{i;t}|\psi_{ic;t};c}$ of the Markov chain mixture as well those determining the true user ideal pf ${}^{\lfloor U}\Theta_{d_{i;t}|\psi_{ic;t},c}$ are known and fixed. They are extended as in Proposition 11.1.5 so that the corresponding factors of the learnt mixture and of the user ideal pf have the common regression vectors $\psi_{i;t} \equiv \left[d_{i+1;t}, \psi'_{i+1;t}\right]' = \left[d'_{(i+1)\ldots\mathring{d};t}, \phi'_{t-1}\right]'$, $i < \mathring{d}$, $\psi_{\mathring{d};t} \equiv \phi_{t-1}$.

Let us search for the advising strategy $\left\{ {}^{\lfloor I}f(c_t, u_{o;t}|d(t-1))\right\}_{t\in t^*}$, selecting both the recommended pointers c_t and the recognizable actions $u_{o;t}$, that minimizes the KL distance of

$$\begin{aligned} {}^{\lfloor I}f(d(\mathring{t}), c(\mathring{t})) &= \prod_{t\in t^*} {}^{\lfloor I}f(d_t, c_t|d(t-1)) \quad \textit{to the user ideal pf} \\ {}^{\lfloor U}f(d(\mathring{t}), c(\mathring{t})) &= \prod_{t\in t^*} {}^{\lfloor U}f(d_t, c_t|d(t-1)) \quad \textit{with factors given by (11.19).} \end{aligned}$$

Then, the optimal simultaneous advising strategy is described by pfs

$$ {}^{\lfloor I}f(c_t, u_{o;t}|\phi_{t-1}) \propto {}^{\lfloor U}\Theta_{c_t|\psi_t}\, {}^{\lfloor U}\Theta_{u_t|\phi_{t-1}} \exp\left[-\omega_\gamma(c_t, \psi_t)\right], \ \textit{where}$$

the array $\omega_\gamma(c_t, \psi_t)$ is generated by the following formulas initialized by $\omega_\gamma(\phi_{\mathring{t}}) = 0, \ \ \forall \phi_{\mathring{t}} \in \phi^*$.

$$\begin{array}{l}
\textit{For} \quad t = \mathring{t}, \ldots, 1 \\
\quad \textit{For} \quad \phi_{t-1} = 1, \ldots, \mathring{\phi} \\
\qquad\qquad\qquad \textit{Set } \omega_\gamma(\phi_{t-1}) = 0 \\
\qquad \textit{For} \quad (c_t, u_{o;t}) = (1,1), \ldots, (\mathring{c}, \mathring{u}_o) \\
\qquad\quad \textit{For} \quad \Delta_t = 1, \ldots, \mathring{\Delta} \\
\qquad\qquad\qquad \textit{Set} \quad \Psi_t \equiv \psi_{0;t} = \left[d'_t, \phi'_{t-1}\right]' \equiv \left[\phi'_t, d'_{t-\partial}\right]' \\
\qquad\qquad\qquad \omega_0(c_t, \psi_{0;t}) \equiv \omega_0(\psi_{0;t}) = \omega_\gamma(\phi_t) \quad \textit{for any } d_{t-\partial} \quad \textit{extending} \quad \phi_t \to \psi_{0;t} = \Psi_t. \\
\qquad\quad \textit{end} \quad \textit{of the cycle over } \Delta_t \\
\qquad\quad \textit{For} \quad i = 1, \ldots, \mathring{\Delta} \\
\qquad\qquad\qquad \omega_i(c_t, \psi_{i;t}) = {}^{\lfloor \psi}\omega_{i-1}(c_t, \psi_{i;t}) \\
\qquad\qquad \textit{For} \quad \Delta_{i;t} = 1, \ldots, \mathring{\Delta}_i \\
\qquad\qquad\qquad \omega_i(c_t, \psi_{i;t}) = \omega_i(c_t, \psi_{i;t}) + \Theta_{\Delta_{i;t}|\psi_{i;t},c_t} \times \\
\qquad\qquad\qquad \times \left[{}^{\lfloor \Delta\Psi}\omega_{i-1}(c_t, [\Delta_{i;t}, \psi_{i;t}]) + \chi\left(i \le \mathring{\Delta}_o\right) \ln\left(\dfrac{\Theta_{\Delta_{i;t}|\psi_{i;t},c_t}}{{}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}}}\right)\right]. \\
\qquad\qquad \textit{end} \quad \textit{of the cycle over } \Delta_{i;t} \\
\quad \textit{In the above recursion, we use} \\
\qquad\qquad\qquad {}^{\lfloor \psi}\omega_{i-1}(c_t, \psi_{i;t}) = \textit{the part of } \omega_{i-1}(c_t, \psi_{i-1;t}) \textit{ independent of entries } \Delta_{i;t}, \\
\qquad\qquad\qquad {}^{\lfloor \Delta\Psi}\omega_{i-1}(c_t, [\Delta_{i;t}, \psi_{i;t}]) = \textit{the part of } \omega_{i-1}(c_t, \psi_{i-1;t}) \textit{ that depends also on } \Delta_{i;t}. \\
\qquad\quad \textit{end} \quad \textit{of the cycle over } i \\
\qquad\qquad\qquad \textit{Set} \quad {}^{\lfloor I}f(c_t, u_{o;t}|\phi_{t-1}) = {}^{\lfloor U}\Theta_{c_t|\psi_t}\, {}^{\lfloor U}\Theta_{u_{o;t}|\phi_{t-1}} \exp[-\omega_{\mathring{\Delta}}(c_t, \phi_{t-1})], \\
\qquad\qquad\qquad \omega_\gamma(\phi_{t-1}) = \omega_\gamma(\phi_{t-1}) + {}^{\lfloor I}f(c_t, u_{o;t}|\phi_{t-1}), \\
\qquad \textit{end} \quad \textit{of the cycle over } (c_t, u_{o;t}) \\
\qquad \textit{For} \quad (c_t, u_{o;t}) = (1,1), \ldots, (\mathring{c}, \mathring{u}_o)
\end{array} \tag{11.20}$$

$$^{\lfloor I}f(c_t, u_{o;t}|\phi_{t-1}) = \frac{^{\lfloor I}f(c_t, u_{o;t}|\phi_{t-1})}{\omega_\gamma(\phi_{t-1})}, \quad \omega_\gamma(\phi_{t-1}) = -\ln(\omega_\gamma(\phi_{t-1})).$$

end of the cycle over $(c_t, u_{o;t})$

end of the cycle over ϕ_{t-1}

end of the cycle over t

Proof: Discrete nature of the recognizable actions implies that formally they play the same role as recommended pointers to components. Thus, the current Proposition copies the results of Proposition 11.2.1 just with appropriate changes in notation. □

The presented proposition, combined with the certainty-equivalent strategy, justifies the following algorithm for the design of the fixed simultaneous advisory system.

Algorithm 11.2.4 (Fixed simultaneous advising minimizing the KL distance)
Initial (off line) mode

- *Estimate Markov chain mixture describing the o-system with the phase state* ϕ_t*, Chapter 10.*
- *Specify the true user ideal pf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{\Delta}_o} {}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}} \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} {}^{\lfloor U}\Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{i;t}}.$$

- *Specify* <u>*time invariant*</u> *user ideal on the recommended pointers* $^{\lfloor U}\Theta_{c_t|\psi_t}$.
- *Select the length of the design horizon* $\mathring{t} \geq 1$.
- *Initialize the iterative mode by setting* $\omega_\gamma(\phi_{\mathring{t}}) = 0$.

Iterative (off line) mode

- *Correct the arrays* $\omega_\gamma(c_t, \psi_t)$*, for* $t = \mathring{t}, \ldots, 1$*, as given in (11.20).*
- *Take the final* ω_γ *as the array defining the optimal* steady-state strategy*, cf. Chapter 3.*

Sequential (on line) mode*, running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t.
2. *Evaluate the ideal pf*

$$^{\lfloor I}f(c_{t+1}, u_{o;t+1}|\phi_t) \propto {}^{\lfloor U}\Theta_{c_{t+1}|\psi_{t+1}} {}^{\lfloor U}\Theta_{u_{o;t+1}|\phi_t} \exp\left[-\omega(c_{t+1}, \psi_{t+1})\right].$$

3. *Present to the user projections of the ideal pf*

$$^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1}\in c^*} {}^{\lfloor I}\Theta_{c_{t+1}, u_{o;t+1}|\phi_t} \prod_{i=1}^{\mathring{\Delta}} \Theta_{\Delta_{i;t+1}|\psi_{t+1}, c_{t+1}}.$$

4. *Go to the beginning of* Sequential mode.

Proposition 11.2.3, combined with the receding-horizon, certainty-equivalent strategy, justifies the following adaptive design algorithm.

Algorithm 11.2.5 (Adaptive version of the optimal simultaneous advising)
Initial (off line) mode

- *Estimate Markov chain mixture describing the o-system with the phase state* ϕ_t*, Chapter 10.*

- *Specify the true user ideal pf on the response of the o-system*

$$^{\lfloor U}f(d_{o;t}|d_o(t-1)) \equiv {}^{\lfloor U}f(d_{o;t}|\phi_{t-1}) = \prod_{i=1}^{\mathring{\Delta}_o} {}^{\lfloor U}\Theta_{\Delta_{i;t}|\psi_{i;t}} \prod_{i=\mathring{\Delta}+1}^{\mathring{d}} {}^{\lfloor U}\Theta_{u_{o(i-\mathring{\Delta});t}|\psi_{i;t}}.$$

- *Specify the user ideal pf* $^{\lfloor U}\Theta_{c_t|\psi_t}$ *on the recommended pointers.*

- *Select the length of the receding horizon* $T \geq 1$*.*

Sequential (on line) mode*, running for* $t = 1, 2, \ldots,$

1. *Acquire the data record* d_t *and determine the state vector* ϕ_t *as well as the data vector* $\Psi_t = \left[d_t', \phi_{t-1}'\right]'$*.*

2. *Update estimates of the model parameters, Section 10.5.*

3. *Initialize the iterative mode by setting* $\tau = t + T$ *and* $\omega_\gamma(\phi_\tau) = 0$*. Omit the initialization of* ω_γ *if* $t > 1$ *and the IST strategy is adopted.*

Iterative mode

- *Correct the arrays* $\omega_\gamma(c_\tau, \psi_\tau)$ *defining the optimal strategy using Proposition 11.2.3 with the horizon* T *and* τ *in the role of* t*.*

4. *Evaluate the ideal pf on recommended pointers* c_{t+1} *and recognizable actions* $u_{o;t}$

$$^{\lfloor I}f(c_{t+1}, u_{o;t+1}|\phi_t) \quad \propto \quad {}^{\lfloor U}f(c_{t+1}|\psi_{t+1})\, {}^{\lfloor U}f(u_{o;t+1}|\phi_t) \exp\left[-\omega_\gamma(c_{t+1}, \psi_{t+1})\right].$$

5. *Present to the user projections of the ideal pf*

$$^{\lfloor I}f(d_{t+1}|\phi_t) = \sum_{c_{t+1} \in c^*} {}^{\lfloor I}f(c_{t+1}, u_{o;t}|\phi_t) \prod_{i=1}^{\mathring{\Delta}} \Theta_{\Delta_{ic_{t+1};t+1}|\psi_{ic_{t+1}}}.$$

6. *Go to the beginning of* Sequential mode.

Remark(s) 11.2.3

1. *It is worth to stress that the user ideal pf on* c_t *is conditioned on* ψ_t*. Thus, it can modify dependence of* c_t *and* $u_{o;t}$*.*

2. *Other design variants, like selection of the most probable advices or grouped version, can be constructed directly according to Chapter 7, using the above evaluations.*

11.3 Interaction with operator

Here, we outline design of strategies generating presentation and signaling actions using the model presented in Section 7.1.3 in the case of Markov chain mixtures.

11.3.1 Assigning priorities

Again, Markov chain mixtures make this task formally much easier than in the normal case. Dimensionality is the main obstacle. Thus, similarly as in Chapter 9, we restricts ourselves to the simplest case with $\mathring{p}_t = 1$. We also assume that there are no surplus p-innovations in order to simplify presentation of results.

First we find explicitly algorithm how to compute influence of the $p_t \in \{1, \ldots, \mathring{d}\}$. It requires evaluation of marginal pfs $\Theta_{d_{p_t;t}|\phi_{t-1}}$, ${}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1}}$. It is reasonable to evaluate them jointly in order to decrease the computational load.

Algorithm 11.3.1 (Marginal pfs for fixed p_t, $d_{p_t;t}$, t, ϕ_{t-1})

$$
\begin{array}{llll}
 & & & \textit{Set } \Theta_{d_{p_t;t}|\phi_{t-1}} = 0, \ {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1}} = 0 \\
\textit{For} \quad c_t \in c^* & & & \\
 & & & \textit{Set } \Theta_{d_{p_t;t}|\phi_{t-1},c_t} = 0, \ {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1},c_t} = 0 \\
 & \textit{For} \quad d_{1\_(p_t-1);t} \in d^*_{1\_(p_t-1)} & & \\
 & & & \textit{Create } \psi_{p_t;t} = [d_{1\_(p_t-1);t}, \phi'_{t-1}]' \\
 & & & \textit{Set } \Theta_{d_{1\_p_t;t}|\phi_{t-1},c_t} = 1, \ \Theta_{d_{1\_p_t;t}|\phi_{t-1},c_t} = 1 \\
 & & \textit{For} \quad i = 1, \ldots, p_t & \\
 & & & \Theta_{d_{1\_p_t;t}|\phi_{t-1},c_t} = \Theta_{d_{1\_p_t;t}|\phi_{t-1},c_t} \Theta_{d_{i;t}|\psi_{i;t},c_t} \\
 & & & {}^{\lfloor I}\Theta_{d_{1\_p_t;t}|\phi_{t-1},c_t} = {}^{\lfloor I}\Theta_{d_{1\_p_t;t}|\phi_{t-1},c_t} \, {}^{\lfloor I}\Theta_{d_{i;t}|\psi_{i;t},c_t} \\
 & & \textit{end} \quad \textit{of the cycle over } i & \\
 & & & \Theta_{d_{p_t;t}|\phi_{t-1},c_t} = \Theta_{d_{p_t;t}|\phi_{t-1},c_t} + \Theta_{d_{1\_(p_t-1);t}|\phi_{t-1},c_t} \\
 & & & {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1},c_t} = {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1},c_t} + {}^{\lfloor I}\Theta_{d_{1\_(p_t-1);t}|\phi_{t-1},c_t} \\
 & \textit{end} \quad \textit{of the cycle over } d_{1\_(p_t-1);t} & & \\
 & & & \Theta_{d_{p_t;t}|\phi_{t-1}} = \Theta_{d_{p_t;t}|\phi_{t-1}} + \alpha_{c_t}\Theta_{d_{p_t;t}|\phi_{t-1},c_t} \\
 & & & {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1}} = {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1}} + {}^{\lfloor I}\Theta_{c_t\phi_{t-1}} \, {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1},c_t} \\
\textit{end} \quad \textit{of the cycles over } c_t & & &
\end{array}
$$

The outputs of this algorithm provides the model relating presentation action to the data record

$$
f(d_t|\phi_{t-1}, p_t) = \sum_{c_t \in c^*} \alpha_{c_t} \prod_{i=1}^{\mathring{d}} \Theta_{d_{i;t}|\psi_{i;t},c_t} \frac{{}^{\lfloor I}\Theta_{d_{p_t}|\phi_{t-1}}}{\Theta_{d_{p_t}|\phi_{t-1}}}. \tag{11.21}
$$

Proposition 11.3.1 (Optimal presentation) *Let us consider that the academic, industrial or simultaneous design has provided the optimum model*

$$
{}^{\lfloor I}f(d_t|d(t-1)) = \sum_{c_t \in c^*} {}^{\lfloor I}\Theta_{c_t|\phi_{t-1}} \prod_{i=1}^{\mathring{d}} {}^{\lfloor I}\Theta_{d_{i;t}|\psi_{i;t},c_t}.
$$

Let the signaling strategy makes the operator fully alert, i.e. signaling actions $s(\mathring{t}) \equiv 1$. Let us assume that $\mathring{p}_t = 1$. Let us specify the user <u>*data invariant*</u> *ideal pf ${}^{\lfloor U}f(p_t)$ on the set of possible presentation actions $p^* \equiv \{1, \ldots, \mathring{d}_o\}$.*

The optimal presentation strategy assigns the higher priority to the entries of $d_{p_t;t}$ the higher are values of the following pf

$$
f(p_t|\phi_{t-1}) \propto {}^{\lfloor U}f(p_t) \exp\left[-\omega_\gamma(p_t, \phi_{t-1})\right]. \tag{11.22}
$$

The table $\omega_\gamma(p_t, \phi_{t-1})$ is generated by the following algorithm initialized by $\omega_\gamma(\phi_{\mathring{t}}) = 0, \quad \forall \phi_{\mathring{t}} \in \phi^*$

For $\quad t = \mathring{t}, \ldots, 1$

For $\quad \phi_{t-1} = 1, \ldots, \mathring{\phi}$

$$
\begin{aligned}
&\qquad\qquad \textit{Set } \omega_\gamma(\phi_{t-1}) = 0 \qquad (11.23)\\
&\textit{For} \quad i = \mathring{d}, \ldots, 1\\
&\quad \textit{For} \quad d_{i;t} = 1, \ldots, \mathring{d}_i\\
&\qquad\qquad \textit{Set} \quad \psi_{i-1;t} \equiv \left[d_{i;t}, \psi'_{i;t}\right], \quad \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} = 0.\\
&\qquad \textit{For} \quad c_t = 1, \ldots, \mathring{c}\\
&\qquad\qquad \textit{Set } \Theta_{d_{i\_\mathring{d}}|\psi_{i;t}, c_t} = 1.\\
&\qquad\quad \textit{For} \quad j = i, \ldots, \mathring{d}\\
&\qquad\qquad \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}, c_t} = \Theta_{d_{i\_\mathring{d}}|\psi_{i;t}, c_t} \Theta_{d_{j;t}|\psi_{j;t}, c_t}.\\
&\qquad\quad \textit{end} \quad \textit{of the cycle over } j\\
&\qquad\qquad \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} = \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} + \alpha_{c_t} \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}, c_t}.\\
&\qquad \textit{end} \quad \textit{of the cycle over } c_t\\
&\quad \textit{end} \quad \textit{of the cycle over } d_{i;t}\\
&\textit{end} \quad \textit{of the cycle over } i\\
&\textit{For} \quad d_t \equiv (d_{1;t}, \ldots, d_{\mathring{d};t}) = \mathbf{1}_{\mathring{d}}, \ldots, [\mathring{d}, \ldots, \mathring{d}]\\
&\qquad\quad \textit{Set } \omega_0(\psi_{0;t}) \equiv \omega_\gamma(\phi_t) \quad \textit{for any } d_{t-\partial} \quad \textit{extending} \quad \phi_t = [d'_t, \phi'_{t-1}]' \to \psi_{0;t} = \Psi_t.\\
&\textit{end} \quad \textit{of the cycle over } d_t \equiv (d_{1;t}, \ldots, d_{\mathring{d};t})\\
&\textit{For} \quad p_t = 1, \ldots, \mathring{d}\\
&\qquad\qquad \textit{Evaluate } \Theta_{d_{p_t,t}|\phi_{t-1}}, \quad {}^{\lfloor I}\Theta_{d_{p_t,t}|\phi_{t-1}} \quad \textit{using Algorithm 11.3.1.}\\
&\quad \textit{For} \quad i = 1, \ldots, \mathring{d}\\
&\qquad\qquad \omega_i(p_t) \equiv {}^{\lfloor \psi}\omega_{i-1}(p_t, \psi_{i;t})\\
&\qquad\qquad \omega_i(p_t, \psi_{i;t}) = \omega_i(p_t, \psi_{i;t}) + \Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} \frac{{}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1}}}{\Theta_{d_{p_t;t}|\phi_{t-1}}} \times\\
&\qquad\qquad \times \left[{}^{\lfloor \Delta\Psi}\omega_{i-1}(p_t, [\Delta_{i;t}, \psi_{i;t}]) + \chi\left(i \le \mathring{\Delta}_o\right) \ln\left(\frac{\Theta_{d_{i\_\mathring{d};t}|\psi_{i;t}} \; {}^{\lfloor I}\Theta_{d_{p_t;t}|\phi_{t-1}}}{\Theta_{d_{p_t;t}|\phi_{t-1}} \; {}^{\lfloor U}\Theta_{d_{i;t}|\psi_{i;t}}} \right) \right].\\
&\quad \textit{end} \quad \textit{of the cycle over } i\\
&\textit{In the above recursion, we use}\\
&\qquad\qquad {}^{\lfloor \psi}\omega_{i-1}(p_t, \psi_{i;t}) = \textit{the part of } \omega_{i-1}(p_t, \psi_{i-1;t}) \textit{ independent of entries } d_{i;t},\\
&\qquad\qquad {}^{\lfloor \Delta\Psi}\omega_{i-1}(p_t, [d_{i;t}, \psi_{i;t}]) = \textit{the part of } \omega_{i-1}(p_t, \psi_{i-1;t}) \textit{ that depends on } d_{i;t}.\\
&\qquad\qquad \textit{Set} \quad {}^{\lfloor I}f(p_t|\phi_{t-1}) = {}^{\lfloor U}\Theta_{p_t|\phi_{t-1}} \exp[-\omega_{\mathring{d}}(p_t, \psi_{\mathring{d};t})]\\
&\qquad\qquad \omega_\gamma(\phi_{t-1}) = \omega_\gamma(\phi_{t-1}) + {}^{\lfloor I}f(p_t|\phi_{t-1})\\
&\textit{end} \quad \textit{of the cycle over } p_t\\
&\textit{For} \quad p_t = 1, \ldots, \mathring{d}\\
&\qquad\qquad {}^{\lfloor I}f(p_t|\phi_{t-1}) = \frac{{}^{\lfloor I}f(p_t|\phi_{t-1})}{\omega_\gamma(\phi_{t-1})}, \quad \omega_\gamma(\phi_{t-1}) = -\ln(\omega_\gamma(\phi_{t-1})).\\
&\textit{end} \quad \textit{of the cycle over } p_t\\
&\textit{end} \quad \textit{of the cycle over } \phi_{t-1}\\
&\textit{end} \quad \textit{of the cycle over } t
\end{aligned}
$$

Proof: With the prepared model a version of industrial design has arisen. Consequently, the corresponding algorithm can be copied with an appropriate change of notation ∎

11.3.2 Stimulating the operator

The signaling strategy makes the operator alert. It asks him to follow the advised actions when the ideal pf resulting from an academic, industrial or simultaneous design gives significantly smaller KL distance to the user ideal pf than the distance of the estimated model to it.

The model relating the signaling action s_t to the guided-system response is

$$\begin{aligned}
{}^{\lfloor I}f(d_t, s_t|d(t-1)) &\equiv {}^{\lfloor I}f(d_t|s_t, d(t-1))\, {}^{\lfloor I}f(s_t|d(t-1)),\ s_t \in s^* \equiv \{0,1\} \text{ with} \quad (11.24)\\
{}^{\lfloor I}f(d_t|s_t = 0, d(t-1)) &\equiv f(d_t|d(t-1)) \equiv \sum_{c\in c^*} \alpha_c \prod_{i=1}^{\mathring{d}} \Theta_{d_{i;t}|\psi_{i;t},c} \equiv \text{learnt mixture}\\
{}^{\lfloor I}f(d_t|s_t = 1, d(t-1)) &\equiv {}^{\lfloor I}f(d_t|d(t-1)) \equiv\\
&\equiv \sum_{c_t\in c^*} {}^{\lfloor I}f(c_t|\phi_{t-1}) \prod_{i=1}^{\mathring{d}} {}^{\lfloor I}\Theta_{d_{i;t}|\psi_{i;t},c_t} \equiv \text{designed mixture and}\\
{}^{\lfloor I}f(c_t|\phi_{t-1}) &\propto {}^{\lfloor U}f(c_t|\phi_{t-1}) \exp[-\omega_\gamma(c_t, \phi_{t-1})].
\end{aligned}$$

The model (11.24) is a special version of the model (11.13) used in the academic design. Thus, the signalling design reduces completely to it.

Problem 11.3.1 (Completion of Markov-chain design suite) *This chapter covers all basic steps of the design taking advantage of the formal simplicity of the Markov-chain case. It is fair to underline that the presented algorithms are applicable in low dimensional cases only. All available art and possibly additional results will be needed in order to cope with high dimensional arrays treated. Sure steps to be taken are*

- *exploitation of the spare nature of the underlying transition matrices,*
- *use of the well developed art of graphs describing relationships among non-trivially occurring state transitions,*
- *orientation on search for approximate stationary solutions …*

Chapter 12

Sandwich BMTB for mixture initiation

The research described in this work has been stimulated by successes and limitations of the mean tracking (MT) clustering algorithm [11]. We have designed factors of a specific type, called *MT factors* when unifying the overall approach to the learning and design. These factors have allowed us to interpret the MT clustering algorithm in a Bayesian way. It has opened the possibility to use the tool set available within Bayesian paradigm to these factors and to strengthen the original algorithm as well as its exploitation. It was found that build-in independence of entries of d_t prevents the full use of such factors for advising. This makes us to use some excellent features of the MT algorithm only for supporting the decisive learning step, i.e. the learning initiation.

The considered *MT uniform factor* makes the static prediction of a real-valued data entry $d_t \in [-1, 1]$. The factor is described by the pdf

$$f(d_t|a_t, d(t-1), \Theta) = f(d_t|\mu) \equiv \mathcal{M}_{d_t}(\mu) \equiv \left\{ \begin{array}{ll} \frac{1-\varepsilon}{2b} & \text{if } |d_t - \mu| \leq b \\ \frac{\varepsilon}{2(1-b)} & \text{otherwise} \end{array} \right., \text{ where} \tag{12.1}$$

the unknown parameter $\Theta = \mu \in [-1 + b, 1 - b]$. The width $0 < b < 1$ and $0 < \varepsilon < 1$, $\varepsilon \approx 0$ are fixed optional parameters of this pdf.

The *sandwich initiation*, described here, uses Bayesian methodology for estimating the key fixed parameter, namely, the *box width* b. Then, it exploits simplicity with which the MT algorithm searches for the box position at local stationary points in the data space. The local maxima found among them then serve as guesses of initial positions for the final Bayesian step described in detail in Chapter 8.

The MT algorithm moves windows through data space without changing the width of the "inspecting" window. This important property can be used more generally by introducing *MT normal factors*. They are simply normal pdfs with a fixed noise variance. They allow us to make *dynamic clustering* in the vein of uniform MT factors. The normal MT factor is

$$f(d_t|\mathcal{P}_{d_t}, \Theta) = \mathcal{N}_{d_t}(\theta' \psi_t, r), \quad \text{where}$$

$\Theta \equiv \theta \equiv$ regression coefficients in $\theta^* \equiv \mathring{\psi}$-dimensional real space, ψ is the regression vector and r is a known positive noise variance. These factors serve to the same purpose as MT factors. They are more computationally demanding but – unlike MT factors – they suit to modelling of dynamic systems. Their use in the BMTB sandwich is straightforward and therefore is skipped here. They are discussed in Chapter 13 as they serve for modelling of dependence of discrete-valued variables on continuous ones.

Important but restricted role of MT factors implies that layout of this chapter differs from that of Chapters 8, 10. It focuses on facts related to the selection of the prior pdf $f(\Theta)$ for normal mixture estimation.

After preparing common tools, Section 12.1.1, the conceptual BMTB algorithm is described, Section 12.2. Then the steps creating it, distinguished by prefixes B-, MT and -B, are presented in Sections 12.3, 12.4 and 12.5, respectively.

12.1 Common tools

12.1.1 Properties of MT factor

The uniform MT factor is a bit non-standard pdf. Thus, it makes sense to evaluate its moments.

Proposition 12.1.1 (Moments of the uniform MT factor) *Let $f(d|\mu) = \mathcal{M}_d(\mu)$. Then, for $i = 0, 1, \ldots$*

$$\begin{aligned}
\mathcal{E}\left[d^i|\mu\right] &= \frac{\varepsilon}{2(i+1)(1-b)}\left[1-(-1)^{i+1}+(\mu-b)^{i+1}-(\mu+b)^{i+1}\right]+\frac{1-\varepsilon}{2(i+1)b}\left[(\mu+b)^{i+1}-(\mu-b)^{i+1}\right]\\
\mathcal{E}[d|\mu] &= \mu\left[1-\frac{b\varepsilon}{1-b}-\varepsilon\right] \to \mu \text{ for } \varepsilon \to 0, \quad \operatorname{cov}[d|\mu] = \frac{b^2}{3} \text{ for } \varepsilon \to 0. \qquad (12.2)
\end{aligned}$$

Proof: It follows from the identity

$$\begin{aligned}
m_i \equiv \mathcal{E}\left[d^i|\mu\right] \equiv \int d^i f(d|\mu)\, dd &= \frac{\varepsilon}{2(1-b)}\left[\int_{-1}^{\mu-b} d^i\, dd + \int_{b+\mu}^{1} d^i\, dd\right] + \frac{1-\varepsilon}{2b}\int_{\mu-b}^{\mu+b} d^i\, d = \\
&= \frac{\varepsilon}{2(i+1)(1-b)}\left[1-(-1)^{i+1}+(\mu-b)^{i+1}-(\mu+b)^{i+1}\right]+\frac{1-\varepsilon}{2(i+1)b}\left[(\mu+b)^{i+1}-(\mu-b)^{i+1}\right].
\end{aligned}$$

For $i = 1$, $\mathcal{E}[d|\mu] = \mu\left[1-\frac{b\varepsilon}{1-b}-\varepsilon\right]$. The variance is found in similar way using the identity $\operatorname{cov}[d|\mu] = \mathcal{E}\left[d^2|\mu\right] - \mathcal{E}^2[d|\mu]$. □

12.1.2 KL distance of MT factors

For judging proximity of a pair of MT factors, we use their KL distance.

Proposition 12.1.2 (KL distance of parameterized MT factors) *Let d_t be a scalar quantity in $[-1,1]$ and $f(d(t)) = \prod_{t\in t^*}\mathcal{M}_{d_t}(\mu)$, $\tilde{f}(d(t)) = \prod_{t\in t^*}\mathcal{M}_{d_t}(\tilde{\mu})$. The parameters $\mu, \tilde{\mu}$ are assumed to be known. Their KL distance is given by the formula*

$$\mathcal{D}(f||\tilde{f}) = \mathring{t}K\operatorname{abs}(\mu-\tilde{\mu}),\; K \equiv \frac{1-\varepsilon}{2b}\ln\left(\frac{(1-\varepsilon)(1-b)}{b\varepsilon}\right)+\frac{\varepsilon}{2(1-b)}\ln\left(\frac{b\varepsilon}{(1-b)(1-\varepsilon)}\right). \qquad (12.3)$$

Proof:

$$\begin{aligned}
\mathcal{D}(f||\tilde{f}) &\equiv \int f(d(t)|\mu)\ln\left(\frac{f(d(t)|\mu)}{f(d(t)|\tilde{\mu})}\right) dd(t) = \sum_{t\in t^*}\int \mathcal{M}_{d_t}(\mu)\ln\left(\frac{\mathcal{M}_{d_t}(\mu)}{\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t = \\
&= \sum_{t\in t^*}\left[\int_{\mu-b}^{\mu+b}\frac{1-\varepsilon}{2b}\ln\left(\frac{1-\varepsilon}{2b\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t + \int_{-1}^{\mu-b}\frac{\varepsilon}{2(1-b)}\ln\left(\frac{\varepsilon}{2(1-b)\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t + \right.\\
&+ \left.\int_{\mu+b}^{1}\frac{\varepsilon}{2(1-b)}\ln\left(\frac{\varepsilon}{2(1-b)\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t\right].
\end{aligned}$$

Let us evaluate a generic term assuming that $\tilde{\mu} \le \mu$

$$\begin{aligned}
&\int_{\mu-b}^{\mu+b}\frac{1-\varepsilon}{2b}\ln\left(\frac{1-\varepsilon}{2b\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t + \int_{-1}^{\mu-b}\frac{\varepsilon}{2(1-b)}\ln\left(\frac{\varepsilon}{2(1-b)\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t + \\
+ &\int_{\mu+b}^{1}\frac{\varepsilon}{2(1-b)}\ln\left(\frac{\varepsilon}{2(1-b)\mathcal{M}_{d_t}(\tilde{\mu})}\right) dd_t = \\
= &\int_{\tilde{\mu}+b}^{\mu+b}\frac{1-\varepsilon}{2b}\ln\left(\frac{(1-\varepsilon)2(1-b)}{2b\varepsilon}\right) dd_t + \int_{\tilde{\mu}-b}^{\mu-b}\frac{\varepsilon}{2(1-b)}\ln\left(\frac{2b\varepsilon}{2(1-b)(1-\varepsilon)}\right) dd_t = \\
= &(\mu-\tilde{\mu})\underbrace{\left[\frac{1-\varepsilon}{2b}\ln\left(\frac{(1-\varepsilon)(1-b)}{b\varepsilon}\right)+\frac{\varepsilon}{2(1-b)}\ln\left(\frac{b\varepsilon}{(1-b)(1-\varepsilon)}\right)\right]}_{K}.
\end{aligned}$$

For $\tilde{\mu} > \mu$, the role of these variables just exchanges. $\square$

12.1.3 Estimation and prediction with MT factors

MT factors do not belong to the exponential family but their estimation is still relatively simple. They are similar to the uniform pdf with unknown center μ and a fixed length $2b$. The fixed non-zero value $0.5\varepsilon/(1-b)$ considered out of the interval $[\mu-b, \mu+b]$ prevents the learning to break down due to outlying data.

Proposition 12.1.3 (Estimation and prediction of the MT factor) *Let natural conditions of decision making, Requirement 2.5.1, hold, and MT factor (12.1) be estimated. Let uniform prior pdf on $[-1+b, 1-b]$ be used for μ. Then, the posteriori pdf of μ has the form*

$$f(\mu|d(\mathring{t})) \propto \left[\frac{1-\varepsilon}{2b}\right]^{\nu(\mu,\mathring{t})} \left[\frac{\varepsilon}{2(1-b)}\right]^{\mathring{t}-\nu(\mu,\mathring{t})} \equiv \mathcal{M}_{\mu}(\nu(\mu,\mathring{t})), \ \textit{where} \tag{12.4}$$

$\nu(\mu,\mathring{t})$ denotes the number of data points among $d(\mathring{t})$ fulfilling $|d_t-\mu| \leq b$. This number is increased by 1 due to the prior pdf chosen.

The point estimate $\hat{(\mu)} \equiv \hat{\mu}(d(t))$ of μ maximizing a posterior probability (MAP estimate) maximizes $\nu(\mu,\mathring{t})$, i.e. $\nu(\hat{\mu},\mathring{t}) \geq \nu(\mu,\mathring{t}),\ \forall \mu \in \mu^* \equiv [-1+b, 1-b]$.

For a given MAP estimate of parameters the Bayesian prediction can be approximated by the "certainty-equivalence" predictor

$$f(d|d(t)) \approx \mathcal{M}_d(\hat{\mu}(d(t))). \tag{12.5}$$

Proof: By a direct evaluation. The prediction relies on the sharpness of the posterior pdf $f(\mu|d(\mathring{t}))$. $\square$

The fact that the MT algorithm [11] provides efficiently the MAP estimate $\hat{\mu}$ of μ is the main reason for discussing this special factor. Let us recall the essence of the original MT algorithm, [11].

Algorithm 12.1.1 (Basic MT algorithm)
Initial mode

- *Remove outliers from the raw data.*
- *Normalize data $d(\mathring{t})$ so that they belong to $[-1,1]$.*
- *Select the parameter $0 < b < 1$.*
- *Select the initial guess $\hat{\mu}_0$ in $\mu^* \equiv [b-1, 1-b]$.*
- *Select the upper bound $\mathring{n}$ on the number n of iterations.*

Iterative mode

For $n = 1, \ldots, \mathring{n}$
  Set $\hat{\mu}_n \equiv$ *sample mean of those data among $d(\mathring{t})$ belonging to* $[\hat{\mu}_{n-1}-b, \hat{\mu}_{n-1}+b]$.
  Break if $|\hat{\mu}_n - \hat{\mu}_{n-1}| < \varepsilon$.
end *of the cycle over n*

Remark(s) 12.1.1

1. *The algorithm is described for a scalar d_t. It is applicable in high dimensions without a change and with a very low additional computational cost.*
2. *The estimation stops at local maximum of the empirical pdf of data. This fits well to our wish to find all significant modes as the local extremes are of our direct interest.*

3. *The adequate choice of the fixed parameters of the MT factor decides on the achieved quality. Their tuning is described below. It is based on practical similarity of uniform and normal factors. Essentially, a normal static mixture is fitted to the considered scalar normalized data and its (averaged) standard deviation is taken as b. The found mean value of the normal pdf may serve as an initial guess of the unknown μ (B- step).*

   This connection is used in the opposite direction, too. The MT factors estimated in MT step provide initial positions of factors in the final normal mixture (-B step).

4. *The considered normalization is quite sensitive to outlying data. Their presence shrinks data points too much, Consequently, some modes are easily "over-looked" during clustering.*

5. *Estimation of the mixture consisting of MT component is extremely simple and fast. It serves well as a good starting point for estimation of normal mixtures that are able to describe correlation between entries of the multi-variate record d. It is important in order to grasp dynamic properties of the modelled system: MT factors perform static clustering of dynamic data, see Section 6.4.9.*

6. *MT factors are tightly related to histograms with a fixed box length distributed on a variable orthogonal grid.*

7. *Many window describing MT factors can be shifted in parallel.*

8. *The original MT algorithm lacks*

   - *systematic but feasible initiation of the MT algorithm for large data sets, for large $\mathring{t}$,*
   - *recognition whether the reached extreme is local minimum or maximum,*
   - *systematic merging and cancelling of components,*
   - *structure estimation whenever MT algorithm is applied to dynamic data as reflected in Proposition 6.4.10.*

   These aspects are also addressed here.

12.2 Conceptual BMTB algorithm

As said above, this chapter describes an attempt to use simplicity of the MT algorithm and its ability to find efficiently local stationary points in the data space. A simple preliminary one-dimensional Bayesian estimation, *B- step*, serves well for the choice of its critical parameters. The restricted results of the MT algorithm have to be extended to the initial mixture. It is obtained by using complete Bayesian estimation after the BMT estimation, during *-B step*.

Particular steps of the following conceptual BMTB algorithm are elaborated in subsequent sections in detail.

Algorithm 12.2.1 (Conceptual BMTB algorithm)
B- step, *run for $i = 1, \ldots, \mathring{d}$,*

Initial mode:

- *Remove outliers.*
- *Normalize the data so that the majority of them has entries in range $[-1, 1]$.*
- *Specify initial, one-dimensional normal mixture covering safely and uniformly whole range $[-1, 1]$.*

Learning mode: *Use an approximate Bayesian estimation on axes $d_i(\mathring{t})$, Section 8.5, for learning of this normal mixture.*

Terminal actions of B- step: *Specify box widths b_i specific to the handled entry $d_i(\mathring{t})$.*

MT step

Initial mode: *Select the structure of the data vectors Ψ to be described by the static normal mixture as justified by Proposition 6.4.10.*

Specify optional parameters of the MT step using estimation results obtained in B- step.

Learning mode: *Apply MT algorithm to get initial guesses of cluster positions in the space of data vectors Ψ^*.*

Terminal actions of MT step: *Use the results of the B- and MT steps for defining an initial guess of the mixture estimated in the -B step.*

-B step

Initial mode: *Specify optional parameters of the Bayesian initialization and subsequent estimation using the BMT results.*

Learning mode: *Apply an initiation algorithm for normal mixtures, see Section 8.4, using the initial mixture obtained in the BMT steps.*

Apply iterative learning to the mixture obtained, see Section 8.5.

Terminal actions of -B step: *Validate the model, see Section 8.7.*

Stop if validation is successful otherwise go to **MT step**.

12.3 B- step: preparation of MT parameters

One dimensional static mixtures are estimated in this step. For the considered purpose, it is sufficient to initialize the mixture estimation by selecting sufficiently high number of components $\mathring{c}$, typically a few tens. The initial positions of component estimates are equidistantly distributed over the dominant range of scaled data, i.e. over the interval [-1,1]. Components can be just shifted copies. Their common noise variance should make them distinguishable but overlapping, so that distance of centers should be about 4-6 standard deviations of noise and parameter variance should be of the order of the noise variance. These values should be sufficiently fixed, so that the standard recommended value of degrees of freedom $\nu_{i;0} \approx 0.1\mathring{t}$, Remark 6.4.2 is applicable. Universally, equal component weights respecting the recommendation $\kappa_{c;0} \approx 0.1\mathring{t}$ can be and should be chosen. Let us summarize it.

Algorithm 12.3.1 (Initialization of B- step (applied entry-wise))

1. *Perform data preprocessing with outlier removal.*
2. *Scale the data to have zero mean and unit variance.*
3. *Select the number $\mathring{c} \approx 20$ of one-dimensional components.*
4. *Define $\nu_{i;0} \approx 0.1\mathring{t},\ i = 1, \ldots, \mathring{d},\ \kappa_{c;0} \approx 0.1\mathring{t},\ c = 1, \ldots, \mathring{c}$.*
5. *Define distance between component centers $s = 2/(\mathring{c} - 1)$.*
6. *Complement the definition of the prior pdf by selecting $L'DL$ decomposition of respective extended information matrices as follows*

$$L_{c;0} = \begin{bmatrix} 1 & 0 \\ -1 + (c-1)s & 1 \end{bmatrix}, \quad D_{c;0} \equiv D_0 = \begin{bmatrix} \frac{\nu_0 s^2}{16} & \\ & 1 \end{bmatrix}, \quad c \in c^*.$$

Learning mode consists of a direct application of the mixture estimation, presented in Section 8.5, to this static one-dimensional mixture. Thus, it remains to use its results for selection of box widths and possibly of initial positions of boxes entering the MT algorithm. The discussion is formalized with the help of the following simple proposition.

Proposition 12.3.1 (Extension of single variate pdfs) *Let $f(x)$, $f(y)$ be a pair of pdfs and let us search for the joint pdf $f(x,y)$ that*

1. *has $f(x)$, $f(y)$ as its marginal pdfs, i.e. $f(x) = \int f(x,y)\, dy$, $f(y) = \int f(x,y)\, dx$,*
2. *is the most uncertain among such pdfs in order to respect just the considered restrictions: the negative KL distance to uniform pdf is taken as the measure of uncertainty.*

Then, $f(x,y) = f(x)f(y)$.

Proof: We minimize convex functional on a convex set so the we can formulate the problem in terms of the Lagrangian functional

$$\int f(x,y)\left[\ln\left(f(x,y)\right) + \lambda(x) + \lambda(y)\right] dxdy =$$

$$= \int f(x,y) \ln\left(\frac{f(x,y)}{\left(\frac{\exp[-\lambda(x)-\lambda(y)]}{\int \exp[-\lambda(\tilde{x})-\lambda(\tilde{y})]\, d\tilde{x}d\tilde{y}}\right)}\right) dxdy - \ln\left(\int \exp[-\lambda(\tilde{x}) - \lambda(\tilde{y})]\, d\tilde{x}d\tilde{y}\right), \text{ where}$$

$\lambda(x), \lambda(y)$ are Lagrangian multipliers to be chosen so that the requirement 1. is met. The right-hand-side form of the Lagrangian and properties of the KL distance imply that $f(x,y)$ is of the product form. The restriction 1. determines then the factors uniquely. $\square$

We would like to extend the mixtures on individual data entries to mixtures on overall data space. The above proposition shows why it is not possible exactly for the target dimensions $\mathring{d}$ of the order several tens. The number of multi-variate components that have the specified marginal pdfs is extremely large and majority of them are spurious. The spurious components are simply shadows of the true components. Their exclusion is called *shadow cancelling problem.* General solution of the above problem is computationally hard. A promising version is outlined in [97] and elaborated in Section 12.3.2. Before it, a simpler version fitting to the studied sandwich algorithm is proposed.

12.3.1 Simple choice of box width

The simple solution presented here neglects information on the component centers and for each individual data entry $d_{i;t},\ t \in t^*$ specifies a single box width used further on.

The scalar normal mixture estimated for a given data record axis says that with probability α_c the noise variance is r_c. The posterior pdf of α is Dirichlet pdf $Di_\alpha(\kappa_{\mathring{t}})$. It is determined by the statistic $\kappa_{\mathring{t}}$, see Proposition 10.1.1. The posterior pdf of $r_c,\ c \in c^*$, is the product of the inverse Wishart pdfs (8.23) $f(r_c|d(\mathring{t})) \equiv iW_{r_c}\left({}^{\lfloor d}D_{c;\mathring{t}}, \nu_{c;\mathring{t}}\right) \propto r_c^{-0.5(\nu_{c;\mathring{t}}+2)} \exp\left[-\frac{{}^{\lfloor d}D_{c;\mathring{t}}}{2r_c}\right]$. They are determined by the LS remainders ${}^{\lfloor d}D_{c;\mathring{t}}$ and the numbers of degrees of freedom $\nu_{c;\mathring{t}}$, see Proposition 8.1.7. We search for a single representant of these pdfs.

Proposition 12.3.2 (Representant of a group of iW pdfs) *The pdf $f(r|d(\mathring{t})) \equiv \equiv iW_r\left(\sum_{c\in c^*} \hat{\alpha}_{c;\mathring{t}}\, {}^{\lfloor d}D_{c;\mathring{t}}, \sum_{c\in c^*} \hat{\alpha}_{c;\mathring{t}}\nu_{c;\mathring{t}}\right)$ with $\hat{\alpha}_{c;\mathring{t}} = \frac{\kappa_{c;\mathring{t}}}{\sum_{c\in c^*}\kappa_{c;\mathring{t}}}$ minimizes the expected loss*

$$\mathcal{E}\left[\sum_{c\in c^*} \alpha_c \int f(r|d(\mathring{t})) \ln\left(\frac{f(r|d(\mathring{t}))}{f(r = r_c|d(\mathring{t}))}\right) dr\right].$$

Proof: Proposition 2.3.7 implies that we have to minimize the conditional expectation $\mathcal{E}[\cdot|d(\mathring{t})]$ of the loss. It reads

$$\sum_{c\in c^*} \hat{\alpha}_{c;\mathring{t}} \int f(r|d(\mathring{t})) \ln\left(\frac{f(r|d(\mathring{t}))}{f(r_c = r|d(\mathring{t}))}\right) dr = \int f(r|d(\mathring{t})) \ln\left(\frac{f(r|d(\mathring{t}))}{\prod_{c=1}^{\mathring{c}} \left[f(r_c = r|d(\mathring{t}))\right]^{\hat{\alpha}_{c;\mathring{t}}}}\right) dr$$

$$= \mathcal{D}\left(f(r|d(\mathring{t}))||\frac{\prod_{c\in c^*}\left[f(r = r_c|d(\mathring{t}))\right]^{\hat{\alpha}_{c;\mathring{t}}}}{\int \prod_{c\in c^*}\left[f(r = r_c|d(\mathring{t}))\right]^{\hat{\alpha}_{c;\mathring{t}}} dr}\right) - \ln\left(\int \prod_{c\in c^*} \left[f(r = r_c|d(\mathring{t}))\right]^{\hat{\alpha}_{c;\mathring{t}}} dr\right).$$

Properties of the KL distance implies that the minimizing pdf is the weighted geometric mean of pdfs corresponding to particular components. The weighted geometric mean of iW pdfs is iW pdfs with the statistics being weighted arithmetic means of the combined statistics.

Specifically, the expected variance $\hat{r}$ corresponding to the found representant is

$$\hat{r} = \frac{\sum_{c \in c^*} \hat{\alpha}_{c;\mathring{t}} \, {}^{\lfloor d}D_{c;\mathring{t}}}{\sum_{c \in c^*} \hat{\alpha}_{c;\mathring{t}} ({}^{\lfloor d}\nu_{c;\mathring{t}} - 2)}.$$

□

The found common noise variance $\hat{r}$ is converted into the box width b by using probabilistic interpretation (12.1) of MT factors. For a negligible ε, they are simply uniform pdfs with half width equal to b. The overall algorithm for determining the box width b we get by equating its variance $b^2/3$ to the adequate point estimate of r, which is the expected value of the pdf obtained in Proposition 12.3.2.

Algorithm 12.3.2 (Choice of the box widths for respective data axes)
Initial phase

- *Initiate one-dimensional normal mixture estimation according to Algorithm 12.3.1.*
- *Estimate approximately parameters of the normal mixture.*

Evaluation phase

Define box width
$$b = \sqrt{3 \frac{\sum_{c \in c^*} \hat{\alpha}_{c;\mathring{t}} \, {}^{\lfloor d}D_{c;\mathring{t}}}{\sum_{c \in c^*} \hat{\alpha}_{c;\mathring{t}} ({}^{\lfloor d}\nu_{c;\mathring{t}} - 2)}}. \tag{12.6}$$

12.3.2 Selection of centers and box widths through shadow cancelling

The solution presented above neglects the information on positions of one dimensional projections of cluster centers. Moreover, it works with an average variance estimate. The latter item was found critical when both very narrow and wide clusters are needed. This situation is often met, for instance, in the application on rolling mill, see Section 15.4.1. This made us to consider the shadow cancelling problem, see Chapter 6, and to apply the solution proposed in the paper [98] and relying on an algorithm presented in [97].

The solution is constructed step-wise.

1. The prior and posterior pdfs on a multi-variate component is constructed by using a fixed selection of marginal components as a building material. The selection is coded by an *index vector*. The corresponding component weight is also assigned prior and posterior pdfs. Propositions 12.3.1, 6.3.2 motivate the combination selected.
2. The v-likelihood corresponding to the inspected index vector is predicted using ideas developed in connection with merging of components, see Section 6.6.4.
3. The optimization algorithm, proposed in [97] and selecting a given number of index vectors leading to the highest v-likelihood, is applied.

   The algorithm allows us to find a relatively large number of potential multi-variate components that after appropriate flattening serve as initial positions and box widths for MT step or can be directly used as initial estimates of the normal mixture.

ad 1 We assume that the modelled system is described by the normal static mixture written in the matrix form

$$f(d_t|\Theta) = \sum_{c \in c^*} \alpha_c \mathcal{N}_{d_t}(\theta_c, R_c)$$

with the $\mathring{d}$-vector expected values θ_c and diagonal covariance matrices R_c. The marginal on i-th axis is the mixture

$$f(d_{i;t}|\Theta) = \sum_{c \in c^*} \alpha_c \mathcal{N}_{d_{i;t}}(\theta_{ic}, r_{ic}), \text{ where}$$

the probabilistic weights α_c coincide with that of the multi-variate mixture, θ_{ic} are entries of θ_c and r_{ic} are diagonal entries of R_c. The posterior pdfs obtained through the estimation of "marginal" mixtures on respective axes are

$$f(\theta_{i1\_\mathring{c}}, r_{i1\_\mathring{c}}, \alpha | d_i(\tau)) = \prod_{c \in c^*} GiW_{(\theta_{ic}, r_{ic})}(V_{ic;\tau}, \nu_{ic;\tau}) Di_\alpha(\kappa_{i1\_\mathring{c};\tau}), \text{ where}$$

$\tau = \mathring{t}$. The result is valid if the conjugate prior of the same form but with $\tau = 0$ is used. For notation and derivation, see Chapter 8.

For each i, we select the marginal pdf $f(\theta_{i\,\lfloor m_{c_i}}, r_{i\,\lfloor m_{c_i}}, \alpha_{\,\lfloor m_{c_i}} | d_i(\tau))$, ${}^{\lfloor m}c_i \in c^*$ and interpret it as a marginal pdf of the component estimates $f(\theta_{\lfloor m_c}, R_{\lfloor m_c})$, and partial information on the marginal pdf $f(\alpha_{\lfloor m_c} | d(\tau))$ of m-th multi-variate component, $m \in m^* \equiv \{1, \ldots, \mathring{m}\}$. The component is identified by the *index vector* ${}^{\lfloor m}c \equiv \left[{}^{\lfloor m}c_1, \ldots, {}^{\lfloor m}c_{\mathring{d}}\right]'$. It is chosen so that it points to a single marginal component on each axis i. The corresponding uni-variate pdf is a marginal pdf of the constructed component.

Using Proposition 12.3.1, the extension of these marginal pdfs to $\mathring{d}$-dimensional parameters, corresponding to the index vector ${}^{\lfloor m}c$, has the form

$$f(\Theta_{\lfloor m_c} | \alpha_{\lfloor m_c}, d(\tau)) = \prod_{i=1}^{\mathring{d}} GiW_{\theta_{i\,\lfloor m_{c_i}}, r_{i\,\lfloor m_{c_i}}}(V_{i\,\lfloor m_{c_i;\tau}}, \nu_{i\,\lfloor m_{c_i;\tau}}), \ \tau = 0, \mathring{t}.$$

For the considered static mixture, we deal with (2,2)-dimensional extended information matrices $V_{i\,\lfloor m_{c_i;\tau}}$ and their $L'DL$ decompositions are

$$L_{i\,\lfloor m_{c_i;\tau}} = \begin{bmatrix} 1 & 0 \\ \hat{\theta}_{i\,\lfloor m_{c_i;\tau}} & 1 \end{bmatrix}, \ D_{i\,\lfloor m_{c_i;\tau}} = \text{diag}\left[{}^{\lfloor d}D_{i\,\lfloor m_{c_i;\tau}}, {}^{\lfloor \psi}D_{i\,\lfloor m_{c_i;\tau}}\right], \text{ where}$$

$\hat{\theta}_{i\,\lfloor m_{c_i;\tau}}$ is the least-squares (LS) estimate of the offset, ${}^{\lfloor d}D_{i\,\lfloor m_{c_i;\tau}}$ is the LS remainder and ${}^{\lfloor \psi}D_{i\,\lfloor m_{c_i;\tau}}$ coincides with LS-estimates covariance factor, respectively.

The pdfs $f_i \equiv f(\alpha_{\lfloor m_c} | d_i(\tau))$, $i = 1, \ldots, \mathring{d}$, describe the same quantity $\alpha_{\lfloor m_c} \in (0,1)$ and represent the only available experience $\mathcal{P}_\tau$ for constructing the estimate $\hat{f} \equiv \hat{f}(\alpha_{\lfloor m_c} | d(\tau)) \equiv \hat{f}(\alpha_{\lfloor m_c} | \mathcal{P}_\tau)$ of the objective pdf $f \equiv f(\alpha_{\lfloor m_c} | d(\tau))$. If we assume that probabilities of the sets $\{f \equiv f(\alpha | d(\tau)) : f \approx f_i\}$ equal then $\hat{f}$ minimizing the expected KL distance $\mathcal{E}\left[\mathcal{D}\left(\hat{f} || f\right)\right]$ is simply geometric mean of the pdfs forming $\mathcal{P}$, cf. Proposition 6.3.2. Thus, for a chosen ${}^{\lfloor m}c$, the pdf of the weight $\alpha_{\lfloor m_c}$ assigned to the multi-variate component in question becomes

$$f(\alpha_{\lfloor m_c} | d(\tau)) \quad \approx \quad \hat{f}(\alpha_{\lfloor m_c} | d(\tau)) \equiv Di_{\alpha_{\lfloor m_c}} \Bigg[\underbrace{\frac{1}{\mathring{d}} \sum_{i=1}^{\mathring{d}} \kappa_{i\,\lfloor m_{c_i;\tau}}}_{\kappa_{\lfloor m_{c;\tau}}}, \underbrace{\frac{1}{\mathring{d}} \sum_{i=1}^{\mathring{d}} \left(\sum_{c \in \{1,\ldots,\mathring{c}\}} \kappa_{ic;\tau} - \kappa_{i\,\lfloor m_{c_i;\tau}} \right)}_{\rho_\tau - \kappa_{\lfloor m_{c;\tau}}} \Bigg].$$

ad 2 For the chosen index vector ${}^{\lfloor m}c$, the step 1 provides, for $\tau = 0, \mathring{t}$, the prior and posterior pdfs on parameters of the component ${}^{\lfloor m}c$. Bayes rule (2.8), applied to data belonging to ${}^{\lfloor m}c$-th component, implies that

$$f\left(\theta_{\lfloor m_c}, R_{\lfloor m_c} | d(\mathring{t}), {}^{\lfloor m}c\right) = \frac{\mathcal{L}\left(\theta_{\lfloor m_c}, R_{\lfloor m_c}, d(\mathring{t})\right) f\left(\theta_{\lfloor m_c}, R_{\lfloor m_c} | d(0), {}^{\lfloor m}c\right)}{f\left(d(\mathring{t}) | {}^{\lfloor m}c\right)}, \quad \text{where}$$

$\mathcal{L}\left(\theta_{\lfloor m_c}, R_{\lfloor m_c}, d(\mathring{t})\right)$ is the corresponding likelihood function. Let ${}^{\lfloor 0}c$ be a fixed index vector. Then, we can evaluate the ratio of a pair of such identities at the specific point $({}^{\lfloor 0}\theta = 0,\ {}^{\lfloor 0}R = I_{\mathring{d}})$. It gives

$$
\begin{aligned}
f\left(d(\mathring{t})|\,{}^{\lfloor m}c\right) &\propto \frac{\exp\left[-0.5\mathrm{tr}\left({}^{\lfloor d}V_{\lfloor m_{c;0}}\right)\right]}{\mathcal{I}(V_{\lfloor m_{c;0}},\nu_{\lfloor m_{c;0}})}\frac{\mathcal{I}(V_{\lfloor m_{c;\mathring{t}}},\nu_{\lfloor m_{c;\mathring{t}}})}{\exp\left[-0.5\mathrm{tr}\left({}^{\lfloor d}V_{\lfloor m_{c;\mathring{t}}}\right)\right]} = \\
&\propto \exp\left[0.5\sum_{i=1}^{\mathring{d}}\left({}^{\lfloor d}D_{\lfloor m_{c_i;\mathring{t}}} - {}^{\lfloor d}D_{\lfloor m_{c_i;0}} + \hat{\theta}^2_{\lfloor m_{c_i;\mathring{t}}}\,{}^{\lfloor \psi}D_{\lfloor m_{c_i;\mathring{t}}} - \hat{\theta}^2_{\lfloor m_{c_i;0}}\,{}^{\lfloor \psi}D_{\lfloor m_{c_i;0}}\right)\right]\times \\
&\times \prod_{i=1}^{\mathring{d}}\frac{\Gamma(0.5\nu_{\lfloor m_{c_i;\mathring{t}}})\,{}^{\lfloor d}D_{\lfloor m_{c_i;0}}^{0.5\nu_{\lfloor m_{c_i;0}}}\,{}^{\lfloor \psi}D^{0.5}_{\lfloor m_{c_i;0}}}{\Gamma(0.5\nu_{\lfloor m_{c_i;0}})\,{}^{\lfloor d}D_{\lfloor m_{c_i;\mathring{t}}}^{0.5\nu_{\lfloor m_{c_i;\mathring{t}}}}\,{}^{\lfloor \psi}D^{0.5}_{\lfloor m_{c_i;\mathring{t}}}}, \quad \text{where}
\end{aligned}
$$

factors independent of the choice ${}^{\lfloor m}c$ are omitted and formulas (8.22), (10.3) used.

ad 3 For any choice of the index vector ${}^{\lfloor m}c$, we are able to evaluate simply the corresponding v-likelihood and thus potentially search for the $\mathring{m}$ best extensions. The number the index vectors is, however, extremely large so that an adequate search algorithm is needed.

As a preparatory step, a specific *index-based maximization* task and its solution are presented. Then, we show how it can be used to the addressed problem.

Let us consider a linearly ordered set $(x^*, >)$ according to values of the function $l(x)$. We also consider $\mathring{d}$-fold Cartesian product ordered according to the values of a scalar function $L(x_1,\ldots,x_{\mathring{d}})$ defined on a priori given non-empty sequences $x_i \equiv (x_{i1},x_{i2},\ldots,x_{i\mathring{c}_i})$, $i=1,\ldots,\mathring{d}$, $x_{ic}\in x^*$. The function $L(\cdot)$ is assumed to be isotonic, i.e.

$$
l(x_{ic}) > l(\tilde{x}_{ic}) \;\Rightarrow L(x_{i1},\ldots,x_{ic},\ldots,x_{i\mathring{c}}) > L(x_{i1},\ldots,\tilde{x}_{ic},\ldots,x_{i\mathring{c}}). \tag{12.7}
$$

We inspect values of

$$
L\left({}^{\lfloor m}c\right) \equiv L\left({}^{\lfloor m}x_{1\,\lfloor m_{c_1}},\ {}^{\lfloor m}x_{2\,\lfloor m_{c_2}},\ldots,\ {}^{\lfloor m}x_{\mathring{d}\,\lfloor m_{c_{\mathring{d}}}}\right) \tag{12.8}
$$

while taking just one item ${}^{\lfloor m}x_{i\,\lfloor m_{c_i}}$ from each sequence x_i, $i=1,\ldots,\mathring{d}$. We search for indices forming *index vectors*

$$
{}^{\lfloor m}c \equiv \left[{}^{\lfloor m}c_1,\ {}^{\lfloor m}c_2,\ \ldots,\ {}^{\lfloor m}c_{\mathring{d}}\right] \tag{12.9}
$$

giving $\mathring{m}$ largest values of $L(\cdot)$ with arguments of the form (12.8).

The set containing all potential index vectors (12.9) has the form

$$
{}^{\lfloor \bullet}c^* = \left\{\ {}^{\lfloor \bullet}c \equiv [c_1,\, c_2,\, \ldots,\, c_{\mathring{d}}] \mid 1 \le c_i \le \mathring{c}_i,\ \forall\ i=1,2,\ \ldots,\mathring{d}\ \right\}.
$$

The addressed index-based maximization task is formalized as follows.

For $\mathring{m} \le {}^{\lfloor \bullet}\mathring{c}$, find such different index vectors ${}^{\lfloor m}c \in {}^{\lfloor \bullet}c^*$, $m=1,\ldots,\mathring{m}$, that

$$
\min_{\;{}^{\lfloor \bullet}c\in\left\{{}^{\lfloor 1}c,\,{}^{\lfloor 2}c,\ldots,\,{}^{\lfloor \mathring{m}}c\right\}} L\left({}^{\lfloor \bullet}c\right) \ge \max_{\;{}^{\lfloor \bullet}c\in\,{}^{\lfloor \bullet}c^*\setminus\left\{{}^{\lfloor 1}c,\,{}^{\lfloor 2}c,\ldots,\,{}^{\lfloor \mathring{m}}c\right\}} L\left({}^{\lfloor \bullet}c\right).
$$

Algorithm 12.3.3 presented below solves the formulated task. The paper [97] proves it. The following symbols are used.

$\mathcal{C}$ denotes a sequence of index vectors. It contains the optimal index vectors when algorithm stops.

${}^{\lfloor a}\mathcal{C}$ is an auxiliary sequence containing the candidates for the optimal index vectors.

$\mathcal{S}$ is the sequence of sons of the index vector $\mathcal{N}_{length(\mathcal{N})}$. A son of the index vector ${}^{\lfloor \bullet}c$ has the form ${}^{\lfloor \bullet}c + e_i$, where e_i is $\mathring{d}$-vector having the only non-zero value equal 1 on i-th position.

$\Sigma\mathcal{A}$ is the sequence of the values $L(\cdot)$ of members of the sequence $\mathcal{A}$ of index vectors.

Algorithm 12.3.3 (Algorithm solving the index-based maximization)

Initial mode

- *Order the input sequences in non-increasing manner* $l(x_{i1}) \geq l(x_{i2}) \geq \ldots \geq l(x_{i\mathring{c}_i})$, $i = 1,\ldots,\mathring{d}$.
- *Set* $\mathcal{C} = \mathbf{1}_{\mathring{d}} \equiv \underbrace{[1,1,\ldots,1]}_{\mathring{d}-times}$, ${}^{\lfloor a}\mathcal{C} = \emptyset$ *and* $\sum {}^{\lfloor a}\mathcal{C} = \emptyset$.

Iterative mode

1. *Output* $\mathcal{C}$ *and stop if* $length(\mathcal{C}) = \mathring{m}$.
2. *Set* $\mathcal{S} = \emptyset$ *and* $\Sigma\mathcal{S} = \emptyset$.
3. *For* $i = 1,2,\ldots,\mathring{d}$
   Put $\mathcal{C}_{length(\mathcal{C})} + e_i$ *as the last member of* $\mathcal{S}$ *and* $L(\mathcal{C}_{length(\mathcal{C})} + e_i)$ *as the last member of* $\Sigma\mathcal{S}$ *if* $\mathcal{C}_{ilength(\mathcal{C})} < \mathring{c}_i$.
4. *Sort both the sequence* $\Sigma\mathcal{S}$ *in a non-decreasing order and sort the sequence* $\mathcal{S}$ *using the same rule.*
5. *Merge the sequences* $\Sigma\,{}^{\lfloor a}\mathcal{C}$ *and* $\Sigma\mathcal{S}$ *while placing the items from* $\Sigma\,{}^{\lfloor a}\mathcal{C}$ *first and merge the sequences* ${}^{\lfloor a}\mathcal{C}$ *and* $\mathcal{S}$ *using the same rule.*
6. *Erase duplicates from the sequence* ${}^{\lfloor a}\mathcal{C}$ *as well as the corresponding members from* $\Sigma\,{}^{\lfloor a}\mathcal{C}$.
7. *Preserve the initial* $\min\{\mathring{m} - length(\mathcal{C}), length({}^{\lfloor a}\mathcal{C})\}$ *members of both* $\Sigma\,{}^{\lfloor a}\mathcal{C}$ *and* ${}^{\lfloor a}\mathcal{C}$, *erase all others.*
8. *Delete the first member from* $\Sigma\,{}^{\lfloor a}\mathcal{C}$ *while deleting the first member from* ${}^{\lfloor a}\mathcal{C}$ *and placing it as the last member of* $\mathcal{C}$.
9. *Go to the beginning of* Iterative mode.

Now we apply this algorithm in order to find index vectors ${}^{\lfloor \bullet}c$ that combine properly selected components on individual axes so that the constructed multi-variate components have high v-likelihood.

The linearly ordered set x^* consists of statistics

$$x_i \equiv \left(\nu_{c_i;\mathring{t}}, \nu_{c_i;0}, \hat{\theta}_{c_i;\mathring{t}}, \hat{\theta}_{c_i;0}, {}^{\lfloor d}D_{c_i;\mathring{t}}, {}^{\lfloor d}D_{c_i;0}, {}^{\lfloor \psi}D_{c_i;\mathring{t}}, {}^{\lfloor \psi}D_{c_i;0}\right).$$

The individual statistics are ordered according to the value of v-logarithmic likelihood

$$x_a > x_b \Leftrightarrow l(x_a) > l(x_b), \quad \text{where}$$

$$l(x_i) \equiv 0.5\left({}^{\lfloor d}D_{c_i;\mathring{t}} - {}^{\lfloor d}D_{c_i;0} + \hat{\theta}^2_{c_i;\mathring{t}}\,{}^{\lfloor \psi}D_{c_i;\mathring{t}} - \hat{\theta}^2_{c_i;t}\,{}^{\lfloor \psi}D_{c_i;0}\right) + \ln\left(\frac{\Gamma(0.5\nu_{c_i;\mathring{t}})\,{}^{\lfloor d}D^{0.5\nu_{c_i;0}}_{c_i;0}\,{}^{\lfloor \psi}D^{0.5}_{c_i;0}}{\Gamma(0.5\nu_{c_i;0})\,{}^{\lfloor d}D^{0.5\nu_{c_i;\mathring{t}}}_{c_i;\mathring{t}}\,{}^{\lfloor \psi}D^{0.5}_{c_i;\mathring{t}}}\right).$$

The sum of these marginal v-likelihood define the function $L(\cdot)$ assigned to each index vector. Summing meets obviously the monotonicity property (12.7). It confirms the possibility to use Algorithm 12.3.3 to our problem.

For a fixed index vector ${}^{\lfloor m}c$, the estimation on individual entries of data records provides pdfs

$$f(\alpha_{\lfloor m_c}|\Psi_{c_i}(\mathring{t})) = Di_{\alpha_{\lfloor m_c}}\left(\kappa_{i\,{}^{\lfloor m}c_i;\mathring{t}}, \sum_{c=1}^{\mathring{c}} \kappa_{ic;\mathring{t}} - \kappa_{i\,{}^{\lfloor m}c_i;\mathring{t}}\right)$$

of the same component weight $\alpha_{\lfloor m_c}$. Having no reasons to prefer any of data record entries, the geometric mean of the involved pdfs is their adequate compromise. At the same time, the collection $\alpha = (\alpha_{\lfloor m_c})$, $m \in m^* \equiv \{1,\ldots,\mathring{m}\}$ has to describe probabilistic weights of multi-variate mixture, i.e. $\sum_{m\in m^*} \alpha_{\lfloor m_c} = 1$.

This makes us to select $f(\alpha|d(\mathring{t})) = Di_\alpha(\kappa_{\mathring{t}})$. The statistics $\kappa_{\mathring{t}}$ is $\mathring{m}$ vector with ${}^{\lfloor m}c$-th entry chosen

$$\kappa_{\lfloor m c;\mathring{t}} = \frac{1}{\mathring{d}} \sum_{d=1}^{\mathring{t}} \kappa_{i\, \lfloor m c_i;\mathring{t}}.$$

This option finishes extension of the marginal posterior pdfs to the multi-variate description of the overall mixture. In order to avoid falsely low uncertainty, it is necessary to flatten it, see Section 6.4.3. It is wise to select separate flattening factors $\lambda_D \in (0,1), \lambda \in (0,1)$ so that the resulting sums of counters κ and of degrees of freedom ν equal to the sums of prior values, respectively.

Let us describe the overall algorithm.

Algorithm 12.3.4 (Index vector based initiation of MT algorithm)
Initial mode

- *Remove outliers and normalize the data so that the majority of them has entries in range* $[-1,1]$.
- *Specify prior and posterior pdfs on one-dimensional components*

For $i = 1,\ldots,\mathring{d}$

Specify the number of components $\mathring{c}_i$ *and determine shift in positions* $s_i = \dfrac{2}{\mathring{c}_i - 1}$

For $c = 1,\ldots,\mathring{c}_i$

Select statistics $\kappa_{ic;0} \approx 0.1\mathring{t},\ \nu_{ic;0} \approx 1,\ \hat{\theta}_{ic;0} = -1 + (c-1)s_i$

$${}^{\lfloor d}D_{ic;0} = \frac{\nu_{ic;0} s_i^2}{16},\quad {}^{\lfloor \psi}D_{ic;0} = 1$$

end of the cycle over c

end of the cycle over i

For $i = 1,\ldots,\mathring{d}$

Estimate one-dimensional mixture using data $d_i(\mathring{t})$.

For $c = 1,\ldots,\mathring{c}_i$

For the collection of statistics describing both prior and posterior pdf

$x_{ic} \equiv \left(\nu_{ic;\mathring{t}}, \nu_{ic;0}\hat{\theta}_{ic;\mathring{t}}, \hat{\theta}_{ic;0},\ {}^{\lfloor d}D_{ic;\mathring{t}},\ {}^{\lfloor d}D_{ic;0},\ {}^{\lfloor \psi}D_{ic;\mathring{t}},\ {}^{\lfloor \psi}D_{ic;0}\right)$ *and* $\kappa_{ic;\mathring{t}}, \kappa_{ic;0}$,

evaluate individual v-log-likelihood

$$l_{ic} = 0.5\left({}^{\lfloor d}D_{ic;\mathring{t}} - {}^{\lfloor d}D_{ic;0} + \hat{\theta}^2_{ic;\mathring{t}}\, {}^{\lfloor \psi}D_{ic;\mathring{t}} - \hat{\theta}^2_{ic;0}\, {}^{\lfloor \psi}D_{ic;0}\right) + \\ + \ln\left(\frac{\Gamma(0.5\nu_{ic;\mathring{t}})\, {}^{\lfloor d}D_{ic;0}^{0.5\nu_{ic;0}}\, {}^{\lfloor \psi}D_{ic;0}^{0.5}}{\Gamma(0.5\nu_{ic;0})\, {}^{\lfloor d}D_{ic;\mathring{t}}^{0.5\nu_{ic;\mathring{t}}}\, {}^{\lfloor \psi}D_{ic;\mathring{t}}^{0.5}}\right)$$

end of the cycle over c

Order $x_i = [x_{i1},\ldots,x_{i\mathring{c}_i}]$ *so that* $l_{i1} \geq l_{i2} \geq \ldots l_{i\mathring{c}_i}$

end of the cycle over i

- *Select the number* $\mathring{m}$ *of constructed multi-variate components.*
- *Apply Algorithm 12.3.3 on the constructed sequences* $x_i = [x_{i1},\ldots,x_{i\mathring{c}_i}]$ *to get index sequences* ${}^{\lfloor m}c = \left[{}^{\lfloor m}c_1,\ldots,\ {}^{\lfloor m}c_{\mathring{d}}\right],\ m = 1,\ldots,\mathring{m}$.

Evaluation mode

For $m = 1,\ldots,\mathring{m}$

Specify the constructed the posterior pdf on parameters of multi-variate components

$$f\left(\theta_{\lfloor m_c}, R_{\lfloor m_c} | {}^{\lfloor m}c, d(\mathring{t})\right) = \prod_{i=1}^{\mathring{d}} GiW_{\theta_i, r_i}\left(\hat{\theta}_{i\,\lfloor m_{c_i};\mathring{t}}, {}^{\lfloor d}D_{i\,\lfloor m_{c_i};\mathring{t}}, {}^{\lfloor \psi}D_{i\,\lfloor m_{c_i};\mathring{t}} \nu_{i\,\lfloor m_{c_i};\mathring{t}}\right)$$

Specify the statistics $\kappa_\tau = \left[\kappa_{\lfloor 1_{c;\tau}}, \ldots, \kappa_{\lfloor \mathring{c}_{c;\tau}}\right]$ *determining the pdf on component weights*

$$\kappa_{\lfloor m_{c;\tau}} = \frac{1}{\mathring{d}} \sum_{i=1}^{\mathring{d}} \kappa_{i\,\lfloor m_{c_i;\tau}}, \ \tau \in \{0, \mathring{t}\}.$$

Apply one-shot flattening operation, see Section 6.4.3, to get the constructed prior pdf $f(\Theta| {}^{\lfloor m}c)$.

Use the centers $\hat{\theta}_{\lfloor m_c}$ *as initial positions for MT algorithm and recompute the corresponding marginal variances of data to the box widths*

$$b_{\lfloor m_{c_i}} = \sqrt{3 \frac{{}^{\lfloor d}D_{i\,\lfloor m_{c_i}}}{\nu_{i\,\lfloor m_{c_i}} - 2} \left[1 + {}^{\lfloor \psi}D^{-1}_{i\,\lfloor m_{c_i}}\right]}.$$

end of the cycle over m

12.4 MT step: make MT algorithm feasible

The original MT algorithm proved to be unsuitable for the target extent of data sets $\mathring{t} \approx 10^5$. This stimulated a search for improvements described here. Algorithm 12.3.4 provides the promising solution. Here, alternative and complementary improvements are discussed.

12.4.1 Initialization

The original MT algorithm selects gradually each data record $d_t, \ t \in t^*$ as the initial center of the box for searching of stationary positions. This safest way is, however, too expensive for data files containing more than several tens of thousand records. Then, an incomplete search is necessary.

Random starts provide an alternative to exhaustive search. It can be speeded up significantly by exploiting the deterministic nature of the MT algorithm. Essentially, the visited data records that would lead to the same trajectory are excluded from a further search. We want to preserve simplicity of the algorithm. For this reason, we want to recognize such points treated in connection with the box in question.

Considering a fixed axis, we can analyze one-dimensional case. Let $\hat{d}$ be a center of the inspected box. Let us define

$$\begin{aligned}
\overline{D} &\equiv \min\{d_t : d_t \geq \hat{d} + b\}_{t \in t^*} \\
\underline{D} &\equiv \max\{d_t : d_t \leq \hat{d} - b\}_{t \in t^*} \\
\overline{d} &\equiv \max\{d_t : d_t \leq \hat{d} + b\}_{t \in t^*} \\
\underline{d} &\equiv \min\{d_t : d_t \geq \hat{d} - b\}_{t \in t^*}.
\end{aligned}$$

These bound specify ranges of data safely outside $(\underline{D}, \overline{D})$ and safely inside $(\underline{d}, \overline{d})$ of the inspected box trajectory.

We arrive to the same intermediate and final box center if we replace $\hat{d}$ by any $\tilde{d}_t$ (in this box) such that

$$\max\left(\overline{d} - b, \underline{D} + b\right) \leq \tilde{d}_t \leq \min\left(\overline{D} - b, \underline{d} + b\right).$$

Thus, all such data points can be excluded from the set of potential starting points. Evaluation of $\underline{D}, \overline{D}, \underline{d}, \overline{d}$ can be done simultaneously when searching for the points belonging to the inspected box. The comparisons needed for determination of points $\tilde{d}$ can again be made cheaply on the data belonging to the box. It is sufficient to store them temporarily.

12.4.2 Merging and cancelling of centers

Merging and cancelling of boxes are now performed in the MT step using ad hoc rules for their closeness or (almost) emptiness. Within the "sandwich" context, it is sufficient to postpone this action to the beginning of -B step, see Section 12.5. It is sufficient to apply cancelling and merging of Gaussian components, see Section 8.6.4, that arise from MT factors in the -B step.

If need be, we can use finiteness of the KL distance of a pair of MT factors, see Proposition 12.1.2 in order to decrease the number of components before entering the -B step, Section 12.5.

Let us consider the set f^* of MT components having the common box width and individual pdfs $f, \tilde{f}$ defined by centers $\mu, \tilde{\mu}$. Then,

$$\rho(\mu, \tilde{\mu}) \quad \equiv \quad \frac{\mathcal{D}\left(f||\tilde{f}\right)}{\sup_{f\in f^*, \tilde{f}\in f^*} \mathcal{D}(f||\tilde{f})} = \sum_{i=1}^{\mathring{d}} \frac{|\mu_i - \tilde{\mu}_i|}{2}. \tag{12.10}$$

Thus, we can take $\mu, \tilde{\mu}$ as identical if $\rho(\mu, \tilde{\mu})$ is a small number, say of the order 1e-3.

12.4.3 Recognition of a false local maximum

Updating of a box center ends when it does not shift any more. It may, however, happen that it sits between two peaks. We counteract this situation by formulating and testing the following hypothesis.

$$H_0: \text{ the data } d_k,\ k \in k^* \equiv \{1, \ldots, \mathring{k}\} \text{ in the box } [\mu - b, \mu + b] \text{ have the uniform pdf}$$
$$\mathcal{U}_d\left(\mu - b, \mu + b\right) = \equiv \frac{\chi(|d-\mu| \leq b)}{2b} \Rightarrow f(d(\mathring{k})|H_1) = (2b)^{-\mathring{k}} \chi\left(\mu - b \leq \min_{k\in k^*}(d_k) \leq \max_{k\in k^*}(d_k) \leq \mu + b\right).$$

The following alternative is adopted

$$H_1: \ f(d|b_1, b_2) = \mathcal{U}_d(\mu - b, \mu - b_1)\chi(d \leq \mu) + \mathcal{U}_d(\mu + b_2, \mu + b)\chi(d \geq \mu).$$

Taking uniform prior pdfs $f(b_i) = \mathcal{U}_{b_i}(0, b),\ i = 1, 2,\ \varepsilon > 0$, we get

$$\begin{aligned} f(d(\mathring{k})|H_1) &= \int_0^{\overline{d}_1} (b - b_1)^{-\mathring{k}_1}\, db_1 \int_0^{\underline{d}_2} (b - b_2)^{-\mathring{k}+\mathring{k}_1}\, db_2 = \\ &= \frac{1}{\mathring{k}_1 - 1}\left((b - \overline{d}_1)^{-\mathring{k}_1+1} - b^{-\mathring{k}_1+1}\right) \frac{1}{\mathring{k} - \mathring{k}_1 - 1}\left((b - \underline{d}_2)^{-\mathring{k}+\mathring{k}_1+1} - b^{-\mathring{k}+\mathring{k}_1+1}\right), \quad \text{where} \\ \overline{d}_1 &\equiv \max_{k\in k^*}(d_k \leq \mu),\ \ \underline{d}_2 \equiv \min_{k\in k^*}(d_k \geq \mu),\ \mathring{k}_1(d(\mathring{k})) = \text{the number of } d_k \leq \mu,\ k \in k^*. \end{aligned}$$

Using the Bayes rule without prejudice, we get

$$f(H_0|d(\mathring{k})) = \left(1 + \frac{2^{\mathring{k}}}{(\mathring{k}_1 - 1)(\mathring{k} - \mathring{k}_1 - 1)}\left(\frac{b^{\mathring{k}_1}}{(b - \overline{d}_1)^{\mathring{k}_1 11}} - b^{-1}\right)\left(\frac{b^{\mathring{k}-\mathring{k}_1}}{(b - \underline{d}_2)^{\mathring{k}-\mathring{k}_1-1}} - b^{-1}\right)\right)^{-1}. \tag{12.11}$$

The product over the entry-related quantities has to be used in the term added to 1, when dealing with multi-variate case.

12.4.4 Improved MT algorithm

The above ideas lead to the following version of the MT algorithm.

Algorithm 12.4.1 (MT algorithm with data exclusion, removal false maxima & merging)
Initial mode

- *Remove outliers from data sample* $d(\mathring{t})$*, see Section 6.2.2.*
- *Normalize resulting data to the range* $[-1, 1]^{\mathring{d}}$.

- *Attach to each data record* d_t *of the resulting data the flag* $f_t = 1 \equiv$ *ready_to_serve_as_starting_point.*
- *Select widths of boxes* $b_i,\ i = 1,\ldots,\mathring{d}$ *for individual data entries* $d_{i;t}$*, using the B-step, Section 12.3.*
- *Select the upper bound* $\mathring{n}$ *on the number* n *of iterations.*
- *Select the significance level used for cancelling of local minima and "merging" level, i.e. define small values of (12.10).*
- *Initialize the list of found centers* $\mathcal{C}$ *=empty list together with the numbers of points in them* κ *=empty list.*

Iterative mode

For $n = 1,\ldots,\mathring{d}$

Select a new center $\hat{d} =$ *a data record with flag* $f_t = 1$.

Set $\underline{d} = \underline{D} = +\infty \times \mathbf{1}_{\mathring{d}},\ \overline{d} = \overline{D} = -\infty \mathbf{1}_{\mathring{d}}$.

Initialize list of data in the box $\mathcal{D} =$ *empty list.*

Initialize the new center $\hat{d}_0 = 0_{\mathring{d},1}$, *and the new number of points in it* $\mathring{k} = 0$.

For $t = 1,\ldots,\mathring{t}$

Set the $flag\_in = 1 \equiv$ *point is in box.*

For $i = 1,\ldots,\mathring{d}$

IF $d_{i;t} > \hat{d}_{i\mathring{k}} + b_i$

$\overline{D}_i = \min\left(d_{i;t}, \overline{D}_i\right)$, $flag\_in = 0$.

ELSE

If $d_{i;t} < \hat{d}_{i\mathring{k}} - b_i$

$\underline{D}_i = \max\left(d_{i;t}, \underline{D}_i\right)$, $flag\_in = 0$

Else $\underline{d}_i = \min\left(d_{i;t}, \underline{d}_i\right)$, $\overline{d}_i = \max\left(d_{i;t}, \overline{d}_i\right)$.

end of the Else

end of the ELSE

end of the cycle over i

IF $flag\_in = 1$

Set $\mathcal{D} = [\mathcal{D}, (d_t, f_t)]$, $\mathring{k} = \mathring{k} + 1,\ \hat{d}_{\mathring{k}} = \hat{d}_{\mathring{k}-1} + \dfrac{d_t - \hat{d}_{\mathring{k}-1}}{\mathring{k}}$.

end of the IF

end of the cycle over t

Extend the list of centers and of numbers of points in them $\mathcal{C} = [\mathcal{C}, \hat{d}],\ \kappa = [\kappa, \mathring{k}]$.

Label the points not be used as new starts and cancel the local minima:

For $i = 1,\ldots,\mathring{d}$

$\underline{d}_i = \max\left(\overline{d}_i - b_i, \underline{D}_i + b_i\right)$, $\overline{d}_i = \min\left(\overline{D}_i - b_i, \underline{d}_i + b_i\right)$

Set $\mathring{k}_{i1} = 0,\ \overline{d}_{i1} = -\infty,\ \underline{d}_{i2} = +\infty$

end of the cycle over i

For $k = 1,\ldots,\mathring{k}$

For d_t *on* k*-th position of* $\mathcal{D}$ *set* $f_t = 0$.

For $i = 1,\ldots,\mathring{d}$

Set $\mathring{k}_{i1} = \mathring{k}_{i1} + 1$ *and* $\overline{d}_{i1} = \max(\overline{d}_{i1}, d_{i;t})$ *if* $d_{i;t} < \hat{d}_{i\mathring{k}}$ *else set* $\underline{d}_{i2} = \min(\underline{d}_{i2}, d_{i;t})$.

Set $f_t = 1$ *if* $d_{i;t} < \underline{d}_i$ *or* $d_{i;t} > \overline{d}_i$.

end of the cycle over i

end of the cycle over k

Evaluate the probability (12.11) (in multi-variate version) using vectors $\mathring{k}_1$, $\overline{d}_1$, $\underline{d}_2$ *found.*

Cancel the center found from the list $\mathcal{C}$ *and the counter from* κ *if this probability is small.*

end of the cycle over n

Cancel centers with mutual distance (12.10) smaller than chosen level.

Remark(s) 12.4.1

1. *The conditions on identity of the final boxes can be relaxed by setting* $flag_t = 0$ *even when conditions for* $flag_t = 1$ *are slightly violated.*
2. *The choice of initial points is not fixed here. Either a random choice from the data list with flag "free" can be made or Algorithm 12.3.3 employed.*

12.5 -B step: MT results as initial mixture

Important but partial information obtained from MT step has to be extended into the statistics determining the prior GiW pdf conjugate to the target normal mixture. This is theme of this section.

12.5.1 Position and noise covariance

The center of a multi-variate box on the data-vector space Ψ^* is a natural prior LS point estimate $\hat{\theta}_0$ of the offset in the static model modelling Ψ_t. The LS estimate of the matrix noise covariance is diagonal with respective scalar variance values equal to the square of the corresponding width divided by 3, cf. Algorithm 12.3.2.

12.5.2 Remaining statistics

For the static model, fixed component c and factor i, the covariance factor C_c of the LS estimate is scalar that is just a shifted version of the corresponding $\nu_{ic} = \nu_c$, $\forall\, i$. The scalar equals to the corresponding κ_c, again potentially up to some shift. Having no reasons for selecting different shifts, we set $C_{c;0} = \nu_{ic;0} = \kappa_{c;0}$. The value $\kappa_{c;0}$ is reasonable to select proportional to the number $\mathring{k}_{c;\mathring{t}}$ of data vectors captured in the inspected box. The proportion factor is chosen so that the general recommendation for initial values of $\kappa_{c;0} \approx \beta\mathring{t}$, $\beta \approx 0.1$ is met. It gives

$$C_{c;0} = \nu_{ic;0} = \kappa_{c;0} = \mathring{k}_{c;\mathring{t}} \frac{\beta \mathring{c} \mathring{t}}{\sum_{\tilde{c} \in \mathring{c}} \mathring{k}_{\tilde{c};\mathring{t}}}. \tag{12.12}$$

It complements the definition of the prior GiW pdf for the target normal mixture.

12.5.3 Conversion of static components to dynamic factors

MT algorithm provides a static model that is extended to dynamic case by applying it to the data vector Ψ_t. In this way, we get statistics of matrix static components $\nu_{c;0}, V_{c;0}$. We need to convert them into statistics of related dynamic factors. This is done here.

The decisive quadratic form in GiW pdf describing the static matrix parameter

$(\mu, L_{R^{-1}} D_{R^{-1}} L'_{R^{-1}}) =$
$= (\mathring{\Psi}\text{-vector of offsets, } LDL'\text{-decomposition of } \underline{\text{inversion}} \text{ of } (\mathring{\Psi}, \mathring{\Psi})\text{-noise-covariance matrix})$

has the form

$$\mathrm{tr}\left\{ \begin{bmatrix} -I_{\mathring{\Psi}} \\ \mu' \end{bmatrix}' L'DL \begin{bmatrix} -I_{\mathring{\Psi}} \\ \mu' \end{bmatrix} L_{R^{-1}} D_{R^{-1}} L'_{R^{-1}} \right\} =$$

$$= \operatorname{tr}\left\{ \underbrace{\begin{bmatrix} -1 & 0 & \ldots & 0 & \ldots & 0 \\ \theta_1 & -1 & \ldots & 0 & \ldots & 0 \\ & \theta_2 & \ldots & 0 & \ldots & 0 \\ & & \vdots & & & \\ & & \ldots & -1 & \ldots & 0 \\ & & \ldots & \theta_{\mathring{d}} & \ldots & \end{bmatrix}'}_{X'(\theta_1,\ldots,\theta_{\mathring{d}},\Theta_s)} L'DLX(\theta_1,\ldots,\theta_{\mathring{d}},\Theta_s)D_{R^{-1}} \right\}, \text{ where}$$

θ_i are column vectors of transformed regression coefficients, and Θ_s are superfluous parameters appearing due to the static treatment of the whole data vector. Note that they do not occur in the constructed predictive pdf $f(d_t|\phi_{t-1})$. Note also that diagonal matrix $D_{R^{-1}}$ has remained among newly introduced parameters. The resulting form gives the obvious hint how to transform the original $L'DL$ decomposition of the matrix static component on i-th dynamic factors: omit leading $i-1$ rows and columns in $L'DL$ to get $L_i'D_iL_i$. The relation $|R|^{-0.5\nu} = \prod_{i\geq 1} D_{iR^{-1}}^{0.5\nu}$ indicates that we should choose $\nu_i = \nu$.

Let us summarize the result.

Algorithm 12.5.1 (Conversion of static matrix component into dynamic factors)
Initial model

- *Collect the $L'DL$ decomposition $L_1'D_1L_1$ of the extended information matrix and degrees of freedom ν corresponding to matrix static component on data vector Ψ.*

For $i = 1,\ldots,\mathring{d}$

Define sufficient statistics describing dynamic factors with data vectors

$\Psi_i' = [d_i, \psi_i'] = [d_i, [d_{i+1}, \psi_{i+1}]]$

$L_i = \begin{bmatrix} 1 & 0 \\ \bullet & L_{i+1} \end{bmatrix},\ D_i = \begin{bmatrix} \bullet & 0 \\ 0 & D_{i+1} \end{bmatrix},\ \nu_i = \nu.$

end of the cycle over i

Remark(s) 12.5.1

Predominantly diagonal character of $L'DL$ decomposition of the extended information matrix is not exploited in Algorithm 12.5.1. Thus, the conversion can be used out of the discussed context whenever we need to transform estimate of the component parameters into estimates of factor parameters (with uncertainties included).

Problem 12.5.1 (BMTB algorithm with normal MT factors) *MT normal factors can be used instead of MT uniform factors to create an alternative initialization.*

It makes no sense to make one-to-one replacement of MT factors by normal one as it would just increase computational complexity without bringing any obvious advantage. On the other hand, it is possible to adopt an approximate mixture estimation on multivariate mixture made of MT-normal factors. The centers, diagonal covariances and component weights found when solving shadow cancelling problem may serve as the needed start.

Moreover, the possibility to use dynamic nature of MT normal components is worth of being inspected.

Obviously, there is a lot of aspects related to normal mixtures can be exploited while preserving mean-tracking ability of MT normal factors. Surely, this line is worth of elaborating.

Chapter 13

Mixed mixtures

Learning described at general level in Chapter 6 deals with the factorized version of the probabilistic component description. It allows us to use different models for different entries of data records and model jointly continuous and discrete data, to deal with *mixed mixtures.* The specific aspects related to mixtures of this type make content of the current chapter.

Continuous-valued variables are modelled by normal factors, see Chapter 8, discrete-valued ones by Markov chain factors, see Chapter 10. While the former one can be made easily dependent on later ones, the opposite case was not supported. This major and substantial restriction on the presented treatment is weakened here by introducing a special version of normal factors suitable to this purpose. These *MT normal factors* provide a novel practical bridge between the discrete world of Markov chains and that of continuous variables. They are simply normal factors with a small *fixed variance.* Consequently, their learning just tracks positions of the modelled variables similarly as the original MT factors. Their mixture approximates the pf of a discrete-valued variable. In this way, the difficult area of logit regression is avoided.

After usual Section 13.1 preparing common tools, learning of the mixed mixtures is addressed, estimation in Section 13.2 and remaining steps in Section 13.3. In final Section 13.4, a rather novel attempt to estimate ratio of mixtures is presented. It allows to model dependence of discrete quantities on continuous ones and, more importantly, to use data-dependent component weights.

The learning part is only presented. We have two-fold reasons for omission of the design part.

Firstly, it has not been elaborated in detail. The learning part and Chapters 9, 11 imply, however, that it will lead to a feasible blend of manipulations with quadratic forms and tables.

Secondly, the mild – and often inevitable – restriction on modelling of discrete variables by the MT normal factors makes the design part identical with that presented in Chapter 9. It is obvious as we deal anyway with constant point estimates of all parameters in the design.

13.1 Common tools

13.2 Learning with factors on mixed variables

Presence of the mixed discrete and continuous valued variables in respective factors is the key problem of the mixed mixtures. Both variants, i.e. the dependence of the continuous factor output on a discrete variable and the discrete factor output on a continuous variable are discussed in this section.

13.2.1 Factors in exponential family depending on discrete variables

We can always map several the discrete-valued entries of the regression vector on a scalar discrete variable, say $\iota_t \in \iota^* \equiv \{1, \ldots, \mathring{\iota}\}$. Then, it holds.

Proposition 13.2.1 (Estimation in exponential family with discrete valued regressor) *Let us consider i-th factor in the exponential family*

$$f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1), \Theta_i) = \prod_{k=1}^{\mathring{\iota}} A(\Theta_{ki}) \exp\left[\langle B_k(\Theta_{ki}), C_k(\Psi_{ki;t})\rangle\right].$$

The individual factors in the above product we call sub-factor*s.*

Then, the conjugate prior pdf $f(\Theta_i|d(0)) \equiv f(\Theta_i) \equiv f(\Theta_{1i}, \ldots, \Theta_{\mathring{\iota}i})$ has the form

$$f(\Theta_i|d(0)) = \prod_{k=1}^{\mathring{\iota}} A^{\nu_{ki;0}}(\Theta_{ki}) \exp\left[\langle V_{ki;0}, C_k(\Theta_{ki})\rangle\right].$$

The corresponding posterior pdf $f(\Theta_i|D(t))$ preserves this functional form and its sufficient statistics evolves as follows.

$$V_{ki;t} = V_{ki;t-1} + \delta_{\iota_t,k} B_k(\Psi_{ki;t}), \quad \nu_{ki;t} = \nu_{ki;t-1} + \delta_{\iota_t,k}, \quad k = 1, \ldots, \mathring{\iota}.$$

Proof: Omitted. □

Proposition says that the observed discrete-valued entry ι_t serves as a known pointer to the single pair of statistics V_{ki}, ν_{ki} updated at the time t. It influences just this data sorting. Thus, the individual sub-factors with indices ki are processed on the sub-selection of data for which the observed ι_t has the value k. Let us make this conclusion formal.

Proposition 13.2.2 (The overall processing of factors with discrete-valued regressor) *Let us consider learning of a mixed mixture with some factors, say $\{i_1, \ldots, i_{\mathring{j}}\}$, each depending on a discrete-valued observable variable $\iota_{i_j;t}$, $j = 1, \ldots, \mathring{j}$.*

Then, at time t, the factors containing sub-factors with indices $\iota_{i_j;t}$, $j = 1, \ldots, \mathring{j}$ are updated by the current data and forgotten. They are used for computing factor and component weights used in the approximate mixture estimation, Section 6.5. Other sub-factors are untouched and unused.

Proof: Omitted. □

Remark(s) 13.2.1

Often, the discrete-valued variable in the condition keeps its fixed value for a long period. Then, it is potentially possible to pre-segment the data according to it. Before doing it, it is reasonable to consider explicitly such a variable in condition as it may well happen that its different values lead to the identical model. Then the pre-segmentation can be done on aggregated values of this variable.

13.2.2 MT normal factors

The opposite case, when the *discrete-valued factor output* $d_{i;t}$ depends on continuous-valued variables is much more difficult. It is implied by the fact that in a non-trivial dynamic case a model out of the exponential family is needed. Essentially, Markov chains are the only dynamic models staying in the exponential family. Let us indicate why. Let us assume that the model of this type exists. Then,

$$f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1), \Theta) = A(\Theta)\exp[\langle B(d_{i;t}, \psi_{i;t}), C(\Theta)\rangle] \quad \underbrace{\Rightarrow}_{\sum_{d_{i;t}\in d^*_{i;t}} f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1), \Theta)=1} \tag{13.1}$$

$$\sum_{d_{i;t}\in d^*_{i;t}} \exp\left[\langle B(d_{i;t}, \psi_{i;t}), C(\Theta)\rangle\right] = 1/A(\Theta).$$

The right-hand side depends on parameters only, it is independent of the regression vector. Intuitively, it is obvious that no left-hand side of the prescribed form depending in a non-trivial way on continuous entries of $\psi_{i;t}$ exist.

Thus, we either have to enter the difficult area of the logit regression, e.g. [99], or to find other way out. The fixed box width of MT uniform factors, Chapter 12, has inspired us to approximate the desired parameterized model by a mixture of parameterized MT normal factors. The *MT normal factor* is a normal factor that should be able to follow just differences in data positions. Simply, even if we know that the output $d_{i;t}$ of the i-th factor is discrete valued one, we model it by the normal pdf

$$f(d_{i;t}|d_{i+1\_\mathring{d};t}, d(t-1), \Theta) = \mathcal{N}_{d_{i;t}}(\theta_i' \psi_{i;t}, r_i), \text{ where} \tag{13.2}$$

$\Theta_i \equiv \theta_i \equiv$ regression coefficients in $\theta_i^* \equiv \mathring{\psi}_i$-dimensional real space, $\psi_{i;t}$ is the regression vector and r_i is a <u>known</u> positive noise variance.

Let us assume that the predicted factor output $d_{i;t}$ is coded by integers. Thus, normal pdf can distinguishes sharply different values of the factor output if its say 10 times standard deviation is smaller than 0.5. It gives the fixed recommended variance r_i of the order 1e-4 and smaller.

Using results of Chapter 8, it is straightforward to describe the estimation and prediction of the MT normal factors.

Proposition 13.2.3 (Estimation and prediction of the MT normal factor) *Let natural conditions of decision making, Requirement 2.5.1, hold, the MT normal factor (13.2) is treated and a conjugate prior (3.12) as well as conjugate alternative (see Section 3.1) in stabilized forgetting are used. The notation of Chapter 8 is adopted and the factor subscript is suppressed.*

Then, prior, posterior and alternative pdf are normal pdfs

$$\begin{aligned} f(\theta|V) &= \mathcal{N}_\theta\left(\hat{\theta}, r\, {}^{\lfloor\psi}V^{-1}\right) \text{ with } \hat{\theta} \equiv \mathcal{E}[\theta|L, D, \nu] = {}^{\lfloor\psi}L^{-1}\, {}^{\lfloor d\psi}L \text{ and} \\ r &\equiv {}^{\lfloor d}D = {}^{\lfloor d}V - {}^{\lfloor d\psi'}V\hat{\theta} = \text{ fixed} \end{aligned}$$

The predictive pdf is also normal, see Proposition 3.2.1,

$$\begin{aligned} f(d|\psi, V) &= \mathcal{N}_d\left(\hat{\theta}'\psi, r\left(1 + \psi'\, {}^{\lfloor\psi}V^{-1}\psi\right)\right) \text{ or alternatively} \\ f(d|\psi, V) &= \frac{\exp\left\{-\frac{h_d^2}{2r(1+\zeta)}\right\}}{\sqrt{2\pi r(1+\zeta)}}, \text{ with} \end{aligned} \tag{13.3}$$

$$h = (L')^{-1}\Psi,\ h' = [h_d, h_\psi'],\ h_d \text{ is scalar},\ \zeta \equiv \sum_{i=2}^{\mathring{\Psi}} h_i^2/D_{ii}. \tag{13.4}$$

The involved sufficient statistics evolve according to the <u>*recursion with resetting*</u>

$$\begin{aligned} V_t &= \lambda(V_{t-1} + \Psi_t\Psi_t') + (1-\lambda)\, {}^{\lfloor A}V_t,\ V_0 \text{ given and} \\ {}^{\lfloor d}V_t &= r + {}^{\lfloor d\psi}V_t'\hat{\theta}_t \Leftrightarrow {}^{\lfloor d}D_t = r \end{aligned}$$

The information matrices are split as follows

$$\begin{aligned} V &= \begin{bmatrix} {}^{\lfloor d}V & {}^{\lfloor d\psi}V' \\ {}^{\lfloor d\psi}V & {}^{\lfloor\psi}V \end{bmatrix},\ {}^{\lfloor d}V \text{ is scalar. The } L'DL \text{ decomposition is practically used.} \\ V &= L'DL,\ L \text{ is lower triangular matrix with unit diagonal} \\ &\qquad D \text{ is diagonal matrix with non-negative entries} \\ L &= \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L & {}^{\lfloor\psi}L \end{bmatrix},\ D = \begin{bmatrix} {}^{\lfloor d}D & 0 \\ 0 & {}^{\lfloor\psi}D \end{bmatrix},\ {}^{\lfloor d}D \text{ is scalar.} \end{aligned} \tag{13.5}$$

The normalization is finite for all time moments iff ${}^{\lfloor\psi}V_0$ is positive definite, i.e. iff $D > 0$, and $r > 0$.

Proof: Majority of evaluations can be found in [14], [80] or can be derived as a simplified case of the general normal factor with unknown r, see Chapter 8. □

Remark(s) 13.2.2

1. *Algorithm for updating these factors is identical with that for normal factors, see Chapter 8. We just do change the value of the least-squares reminder ${}^{\lfloor d}D$ and the number of degrees of freedom ν. The main difference is in evaluation of predictive pdfs, cf. (8.31) and (13.3).*

2. *Concentration of the measured factors outputs on several values only influences surely information content of data. It is not fully clear whether the influence is positive or negative.*

13.3 Solution of particular learning steps

Previous Section describes estimation of the mixed mixture. It is important but particular step in the whole learning. Here, we review the remaining steps by referring to relevant Sections, where solutions of other steps are briefly presented. We also point to the problems that have to be solved in context of mixed mixtures.

Data preprocessing is mostly oriented on individual signals in question. Generally, it is addressed in Section 6.2. Section 8.2 specialized it to normal factors and thus also to the MT normal factors. Markov chain oriented preprocessing is in Section 10.2.

Problem 13.3.1 (Mixed counterpart of PCA) *Reduction of dimensionality is the only preprocessing step dealing jointly with several signals. It is based on continuous-signal oriented PCA. Thus, a specific solution is needed for the mixed case.*

Incorporation of physical knowledge is strictly signal-oriented up to know. Thus, no additional problems are invoked by the mixed case. General, normal, and thus MT normal, and Markov cases are addressed in Section 6.3, 8.3 and 10.3, respectively.

Construction of the prior pdf is solved predominantly factor-wise. Thus, adequate solutions are ready, see Sections 6.4, 8.4 and 10.4.

Structure estimation is again solved factor-wise. For particular solutions, see Sections 6.6, 8.6, 10.6.

Problem 13.3.2 (Unsolved structure estimation problems) *Majority of the unsolved problems is* <u>*not*</u> *induced by the mixed case. It is, however, worthwhile to list them here.*

Estimation of the structure of MT normal factors is not prepared but solution is clearly a special simplified case solved for normal factors, Section 8.6.1 and references [36, 76, 84].

Estimation of the structure of Markov chains is not practically solved. The particular solution presented in Proposition 10.6.1 has to be elaborated further on, for instance, in the direction outlined in [94] and definitely in the direction of Bayesian networks.

Efficient estimation of the component structure is open problem for all types factors and an adequate solution has to be searched for.

The estimation of mixture structure is conceptually solved through the branching by factor splitting, factor merging and component cancelling. All problems mentioned in relevant Sections of Chapters 6, 8 and 10 persist for the mixed case, too. It adds, however, no new problems in this respect.

Model validation, presented in respective Chapters, is applicable to the mixed case mostly without change. The forgetting based validation, Section 6.7.3 and judging of models in the role of multi-step predictors, Section 7.1.2 seem to be the most promising common techniques.

Problem 13.3.3 (Completion of the mixed-mixtures suite) *Both learning and design algorithms are clearly feasible for the mixed mixtures.*

This is especially true when the normal MT factors are employed for modelling of discrete data. In this case, the learning is obtained through simplification of the learning of normal mixtures and <u>*the design is completely unchanged*</u> *as we deal with fixed point parameter estimates anyway.*

Thus, we see not difficult problems but the job has to be simply done.

13.4 Extension of approximate estimation on mixture ratios

The normal MT factors help us to cope with modelling of dependence of discrete variables on continuous ones. The solution is based on the interpreting discrete variables as continuous ones with a very narrow variation range. In this, way normalization that spoils mixture character of the model is avoided. Estimation of a *ratio of mixtures* is another way how to approach the problem, cf. (13.1). A promising attempt in this respect is presented in this section. It extends descriptive nature of all mixtures as it allows us to deal with dynamic mixtures having data-dependent component weights. Significance of this extension cannot be exaggerated.

13.4.1 Stationary mixture modelling

Here, a parameterization of the joint pdf $f(d(\mathring{t}))$ of data $d(\mathring{t})$ is revised.

The chain rule implies the following expression of the joint pdf $f(d(\mathring{t})) = \prod_{t\in t^*} f(d_t|d(t-1))$ of the data sequence $d(\mathring{t}) \equiv (d_1,\ldots,d_{\mathring{t}})$. Limited memory of the majority real systems justifies the assumption that there is a finite dimensional state $\phi_{t-1} = \phi(d(t-1))$ such that $f(d_t|d(t-1)) = f(d_t|\phi_{t-1})$. We assume that the conditional pdf $f(d_t|\phi_{t-1}) = g(\Psi_t)$ is time and data invariant function of the data vector $\Psi_t = [d_t', \phi_{t-1}']'$. Modelling of the system specifies a parametric class of function $g(\Psi)$. Estimation uses data measured on a specific system in order to find the model within this class whose behavior is the closest one to the modelled system [28].

Here, we adopt an indirect parameterization of models. It allows us to use "universal approximation property" [100] for a "universal" black-box parameterization of dynamic models.

Let us assume that the stationary pdf $f(\Psi_t|\Theta)$ of the data vector Ψ_t, parameterized by an unknown parameter Θ, is at disposal. It defines the parameterized model of the system in question

$$f(d|\phi,\Theta) = \frac{f(\Psi|\Theta)}{f(\phi|\Theta)}, \tag{13.6}$$

where $f(\phi|\Theta) = \int f(\Psi|\Theta)\, dd$.

The possibility to parameterize time and data invariant function of Ψ is the main advantage of this indirect construction of the parameterized model (13.6). Under rather general conditions, almost any practically met invariant pdf $f(\Psi)$ can be approximated by a finite mixture

$$f(\Psi|\Theta) = \sum_{c\in c^*} \alpha_c f(\Psi|\Theta_c, c),\ \Theta \in \Theta^* \equiv \left\{\Theta(\mathring{c}),\ \alpha \in \alpha^* \equiv \left\{\alpha_c \geq 0,\ \sum_{c\in c^*} \alpha_c = 1\right\}\right\}. \tag{13.7}$$

The mixture form (13.7) and (13.6) give the parameterized model constructed

$$f(d|\phi,\Theta) = \frac{\sum_{c\in c^*} \alpha_c f(\Psi|\Theta_c,c)}{\sum_{c\in c^*} \alpha_c f(\phi|\Theta_c,c)} = \sum_{c\in c^*} \underbrace{\frac{\alpha_c f(\phi|\Theta_c,c)}{\sum_{\tilde{c}\in c^*} \alpha_{\tilde{c}} f(\phi|\Theta_{\tilde{c}},\tilde{c})}}_{\alpha_c(\phi)} f(d|\phi,\Theta_c,c). \tag{13.8}$$

The indirectly constructed *dynamic mixture* (13.8) has the time-invariant but *state-dependent weights* $\{\alpha_c(\phi)\}_{c\in c^*}$.

Further on, we use factorized versions of the components $f(\Psi|\Theta_c,c)$ in the mixture (13.7). It means that we apply the chain rule to them

$$f(\Psi|\Theta_c,c) = \prod_{i=1}^{\mathring{\Psi}} f(\Psi_i|\psi_{ic},\Theta_{ic},c). \tag{13.9}$$

The factor $f(\Psi_i|\psi_{ic},\Theta_{ic},c)$ predicts i-th entry Ψ_i of Ψ, i.e. factor output, using the regression vector ψ_{ic} formed by a sub-selection of entries $(\Psi_{i+1},\ldots,\Psi_{\mathring{\Psi}})$. It is parameterized by its unknown parameter Θ_{ic} formed by a sub-selection of Θ_c.

Using (13.9), the conditional pdf (13.8) gets the form

$$f(d|\phi,\Theta) = \frac{\sum_{c\in c^*} \alpha_c \prod_{i=1}^{\mathring{\Psi}} f(\Psi_i|\psi_{ic},\Theta_{ic},c)}{\sum_{c\in c^*} \alpha_c \prod_{i=\mathring{d}+1}^{\mathring{\Psi}} f(\Psi_i|\psi_{ic},\Theta_{ic},c)},$$

i.e. the marginal pdf in the denominator is obtained by omitting the initial factors in the products defining components. This property has motivated positioning of d in $\Psi = [d', \phi']'$.

13.4.2 Extended quasi-Bayes estimation

The mixture estimation is a well elaborated art, [49]. A specific estimation of the rational model (13.8) is, however, missing. It is developed here.

We deal with a special case but important case of the *state in phase form*, i.e.

$$\phi_{t-1} = [d'_{t-1}, \ldots, d'_{t-\partial}]' \Rightarrow \Psi_t = [d_t, d'_{t-1}, \ldots, d'_{t-\partial}, 1].$$

Note that this assumption can be relaxed at least partially but the its use simplifies the presentation.

The quasi-Bayes (QB) estimation, Section 6.5, takes the mixture as the marginal pdf of the pdf

$$f(\Psi, c|\Theta) = \alpha_c f(\Psi|\Theta_c, c) = \prod_{\bar{c}\in c^*} \left[\alpha_{\bar{c}} f(\Psi|\Theta_{\bar{c}}, \bar{c})\right]^{\delta_{c\bar{c}}}. \tag{13.10}$$

The variable c is interpreted as an unobserved random pointer to the active component. The Kronecker symbol $\delta_{c\bar{c}} \equiv 1$ if $c = \bar{c}$ and $\delta_{c\bar{c}} \equiv 0$ otherwise.

Assuming the prior pdf in the form

$$f(\Theta|d(t-1)) \propto \prod_{c\in c^*} \alpha_c^{\kappa_{c;t-1}-1} f(\Theta_c|d(t-1)) = \prod_{c\in c^*} \alpha_c^{\kappa_{c;t-1}-1} \prod_{i=1}^{\mathring{\Psi}} f(\Theta_{ic}|d(t-1)), \tag{13.11}$$

the QB algorithm uses $w_{\bar{c};t} \equiv f(c = \bar{c}|d(t))$ as an approximation of the unknown $\delta_{c\bar{c}}$ and uses the product form of the model (13.10), with replacement of $\delta_{c\bar{c}} \approx w_{\bar{c};t}$ for the Bayesian updating of $f(\Theta|d(t-1))$ to $f(\Theta|d(t))$. It preserves its advantageous product form (13.11) of the pdf on parameters.

In the inspected case, the dynamic model is the ratio of two finite mixtures with constant weights. The numerator is the mixture describing the data vector Ψ and the denominator is the mixture describing the state ϕ. Thus, it seems to be reasonable to

- assume the prior pdf also in the form (13.11),
- compute independently the weight $w_{\bar{c};t}$ for the product version of the numerator and $\tilde{w}_{\bar{c};t}$ for the product version of the denominator,
- update the prior pdf by the ratio of these product forms with the unknown Kronecker symbols replaced by respective weights $w_{\bar{c};t}$, $\tilde{w}_{\bar{c};t}$.

Let us develop the above plan.

The considered pdf in the numerator is the marginal pdf of the pdf (13.10) $f(\Psi, c|\Theta) = \alpha_c f(\Psi|\Theta_c, c)$ with a random unobserved pointer $c \in c^*$. We multiply the equality (13.10) by the prior pdf and integrate it over Θ. Factors corresponding to components different from the c-th one integrate to unity. Thus,

$$f(\Psi, c|d(t-1)) \quad \propto \quad \kappa_{c;t-1} \prod_{i=1}^{\mathring{\Psi}} \underbrace{f(\Psi_i|\psi_i, d(t-1), c)}_{\int f(\Psi_i|\psi_i, d(t-1), \Theta_{ic}, c) f(\Theta_{ic}|d(t-1))\, d\Theta_{ic}}. \tag{13.12}$$

Here, we use the basic property of Dirichlet pdf describing unknown component weights α, namely, $\mathcal{E}[\alpha_c|d(t-1)] = \mathcal{E}[\alpha_c|\kappa_{t-1}] = \frac{\kappa_{c;t-1}}{\sum_{c\in c^*} \kappa_{c;t-1}}$.

Inserting the measured Ψ_t into (13.12) and using Proposition 2.3.4, we get

$$w_{\bar{c};t} \equiv \mathcal{E}[\delta_{c\bar{c}}|d(t)] = f(c = \bar{c}|d(t)) = \frac{\kappa_{\bar{c};t-1} \prod_{i=1}^{\mathring{\Psi}} f(\Psi_{i;t}|\psi_{i;t}, d(t-1), \bar{c})}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t-1} \prod_{i=1}^{\mathring{\Psi}} f(\Psi_{i;t}|\psi_{i;t}, d(t-1), \tilde{c})}. \tag{13.13}$$

We proceed similarly for the denominator taking into account that ϕ is part of Ψ with the oldest data item d omitted. It gives

$$f(\phi, c|d(t-1)) \quad \propto \quad \kappa_{c;t-1} \int \prod_{i=1}^{\mathring{\Psi}} f(\Psi_i|\psi_i, d(t-1), c)\, d\Psi_{(\mathring{d}\partial+1)\ldots\mathring{d}(\partial+1)}. \tag{13.14}$$

Using the formula (13.14) and Proposition 2.3.4 for the measured ϕ_t, we get the approximation of the Kronecker symbol in the denominator in the form

$$\tilde{w}_{\bar{c};t} \quad = \quad \mathcal{E}[\delta_{c\bar{c}}|d(t)] = \frac{\kappa_{\bar{c};t-1} \int \prod_{i=1}^{\mathring{\Psi}} f(\Psi_i|\psi_i, d(t-1), \bar{c})\, d\Psi_{(\mathring{d}\partial+1)\ldots\mathring{d}(\partial+1)}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t-1} \int \prod_{i=1}^{\mathring{\Psi}} f(\Psi_i|\psi_i, d(t-1), \tilde{c})\, d\Psi_{(\mathring{d}\partial+1)\ldots\mathring{d}(\partial+1)}} \tag{13.15}$$

with $\Psi_i = \Psi_{i;t}$ for $i = 1, \ldots, \mathring{d}\partial$.

Substitution of the derived weights into the model (13.10) instead of unknown $\delta_{c\bar{c}}$ gives

$$\begin{aligned}
f(\Psi|\Theta) \quad &\approx \quad \prod_{c\in c^*} [\alpha_c f(\Psi|\Theta_c, c)]^{w_{c;t}} \\
f(\phi|\Theta) \quad &\approx \quad \prod_{c\in c^*} [\alpha_c f(\phi|\Theta_c, c)]^{\tilde{w}_{c;t}} \Rightarrow \\
f(d|\psi, \Theta) \quad &\approx \quad \prod_{c\in c^*} \alpha_c^{w_{c;t}-\tilde{w}_{c;t}} \frac{[f(\Psi|\Theta_c, c)]^{w_{c;t}}}{[f(\phi|\Theta_c, c)]^{\tilde{w}_{c;t}}}.
\end{aligned}$$

Factorized version of individual components (13.9) and the Bayes rule provides the final updating formula posterior one

$$\begin{aligned}
f(\Theta|d(t)) \quad &\propto \quad \prod_{c\in c^*} \alpha_c^{\kappa_{c;t-1}+w_{c;t}-\tilde{w}_{c;t}-1} \prod_{i=1}^{\mathring{d}} [f(\Psi_{i;t}|\psi_{i;t-1}, \Theta_{ic}, c)]^{w_{c;t}} \times \\
&\times \quad \prod_{i=\mathring{d}+1}^{\mathring{\Psi}} [f(\Psi_{i;t}|\psi_{i;t-1}, \Theta_{ic}, c)]^{w_{c;t}-\tilde{w}_{c;t}} \prod_{i=1}^{\mathring{\Psi}} f(\Theta_{ic}|d(t-1)).
\end{aligned} \tag{13.16}$$

Notice that the factors describing the state vector get weights $w_{c;t} - \tilde{w}_{c;t}$. Some of those differences are necessarily negative. While, the original quasi-Bayes estimation uses only positive weights here, also a decrease occurs. It resembles much the estimation strategies known as *reinforcement learning.* Note that troubles may arise when the updated function cannot be normalized to pdf. It happens for those components for which too many decreases have occurred. Intuitively, such a component should be cancelled from mixture.

Straightforward extension of Algorithm 6.5.2 gives the final algorithm.

Algorithm 13.4.1 (Extended Quasi-Bayes estimation)
Initial (off line) mode

- *Select the complete structure of the mixture.*
- *Select prior pdfs $f(\Theta_{ic})$ of the individual factors, ideally, in the conjugate form (3.12) with respect to the parameterized factors $f(\Psi_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)$, $c \in c^*$, $i = 1, \ldots, \mathring{\Psi}$!*
- *Select initial values $\kappa_{c;0} > 0$, say, about 0.1$\mathring{t}$, describing prior pdf of the component weights α.*

- *Compute the initial point estimates of the component weights* $\hat{\alpha}_{c;0} = \frac{\kappa_{c;0}}{\sum_{\tilde{c}c^*} \kappa_{\tilde{c};0}}$.

Sequential (on line) mode, *running for* $t = 1, 2, \ldots$,

1. *Acquire the data record* d_t.
2. *Compute values of the predictive pdfs*
$$f(\Psi_{ic;t}|\psi_{ic;t}, d(t-1), c) = \int f(\Psi_{ic;t}|\psi_{ic;t}, \Theta_{ic}) f(\Theta_{ic}|d(t-1))\, d\Theta_{ic} = \frac{\mathcal{I}(d(t)|ic)}{\mathcal{I}(d(t-1)|ic)},$$
see (2.47), for each individual factor $i \in i^* = \{1, \ldots, \mathring{\Psi}\}$ *in all components* $c \in c^*$ *and the measured data record* d_t.

   The adaptive, one-stage-ahead predictor, see Section 6.1.2, is thus used.
3. *Compute the values of predictive pdfs* $f(d_t|d(t-1), c) = \prod_{i=1}^{\mathring{\Psi}} f(\Psi_{ic;t}|\psi_{ic;t}, d(t-1), c)$ *for the measured data and each individual component* $c \in c^*$.
4. *Compute the probabilistic weights* $w_{c;t}$ *using the formula (6.83) and compute the probabilistic weights* $\tilde{w}_{c;t}$ *using the formula (13.15).*
5. *Update the scalars* $\kappa_{c;t} = \kappa_{c;t-1} + w_{c;t} - \tilde{w}_{c;t}$ *according to (13.16).*
6. *Update the Bayesian parameter estimates of different factors*
$$\begin{aligned} f(\Theta_{ic}|d(t)) &\propto [f(\Psi_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{ic;t}} f(\Theta_{ic}|d(t-1)),\ i = 1, \ldots, \mathring{d} \\ f(\Theta_{ic}|d(t)) &\propto [f(\Psi_{ic;t}|\psi_{ic;t}, \Theta_{ic}, c)]^{w_{ic;t}-\tilde{w}_{ic;t}} f(\Theta_{ic}|d(t-1)),\ i = \mathring{d}+1, \ldots, \mathring{\Psi}, \quad \text{where} \\ w_{ic;t} &= \sum_{\tilde{c}\in c_i^*},\ \tilde{w}_{ic;t} = \sum_{\tilde{c}\in c_i^*} \tilde{w}_{\tilde{c};t}. \end{aligned} \tag{13.17}$$
The set c_i^* *includes pointers to components that contain* i*-th factor.*

   The presence of the factors with indices $i > \mathring{d}$ and their modified updating distinguishes this Algorithm from Algorithm 6.5.2.
7. *Evaluate, the point estimate of the mixing weights* $\mathcal{E}[\alpha_c|d(t)] = \frac{\kappa_{c;t}}{\sum_{\tilde{c}\in c^*} \kappa_{\tilde{c};t}} \equiv \hat{\alpha}_{c;t}$ *and, if need be, characteristics of* $f(\Theta_{ic}|d(t))$ *describing other parameters* Θ_{ic}.
8. *Go to the beginning of* Sequential mode *while data are available, while* $t \le \mathring{t}$.

Problem 13.4.1 (Completion of extension of approximate estimation) *Extension of the quasi-Bayes estimation concerns only manipulation of weights of data. Thus, it is obvious that similar extensions are possible for all remaining versions of approximate estimation, Section 6.5. It should be done.*

Chapter 14

Software aspects

Theory and algorithms described in this work are materialized in software tools. Basic off line evaluations are performed in Matlab-related toolbox *Mixtools* [101]. Matlab serves as a development environment. Decisive algorithms are first coded as m-files, then their mex versions are created. After that, general purpose tools, Section 14.2, get the software rid off any Matlab-related artifacts so that plain ANSI C code is created. It is able to work in parallel and/or industrial environments. For this, the use of advanced MATLAB 6 tools is restricted to simple one-dimensional software structures and cell vectors. Moreover, all *default values* are concentrated in a single function "defaults".

All discussed tools are described in independent guides [101]. Here, we just give their characterization that helps in connecting the theory and its software image Mixtools, Section 14.1.

Mixtools exploits long term experience with ABET toolbox [102] that covers basic decision-making tasks related to normal, multi-variate regression models and (partially) to controlled Markov models. In a sense, ABET can be seen as toolkit working on a single component. Thus, its basic logical structure and component-related algorithms have made logically the basis of Mixtools [12].

The off line computations are enhanced by *Graphical User Interface (GUI)* tools supporting experimentation with algorithms, processing of data, verification of the off line design and design of the form of GUI to be used in the on line mode. A short characterization is given in Section 14.1.4.

The number crunching connected with off line processing can exploit parallel implementation that is again described in an independent report [103] and here just briefly characterized in Section 14.3.

On line software is now constructed by the final user [104, 105] so that here we can just list the basic algorithms that are offered to him, see Section 14.4.

14.1 Off line processing by Mixtools

14.1.1 Construction of names

A heterogenous research team contributes to the software materializing the theory of this work. Its efficient communication requires well defined formal agreements.

Agreement 14.1.1 (Construction of identifiers – cryptonyms) *The names constructed according to the following rules are called* cryptonyms.

- *Global variables have upper case letters in name.*

  They should be eliminated from processing as much as possible.

- *Structures and cell vectors begin with an upper case letter, the rest of the name is made of lower case letters.*

- *Names are selected from the* term base *listed in the next subsection.*

- *Function names have maximum length of 8 symbols and consist of lower case letters and digits*

- *The following conventions are used*

prefix conventions

fac	*the item relates to factors*
com	*the item relates to components*
mix	*the item relates to mixtures*
n	*number of the items*

suffix conventions

s	*array of items*
wg	*weight*
ll	*v-log-likelihood*
d	*difference*
0	*initial item*
digit	*distinquishes items*
t	*trajectory*

Agreement 14.1.2 (List of cryptonyms)

Data management	
TIME	*processing time*
DATA	*data sample*
ndat	*length of data*
psi	*regression vector*
Psi	*data vector*
npsi	*length of regression vector*
nPsi	*length of data vector*

Factors	
Fac	*factor*
Facs	*array of factors*
fac	*position of a factor in an array of factors*
ychn	*modeled channel*
str	*structure of regression vector*
dfm	*degrees of freedom of a factor*
	Standard ARX factors
LD	*L'DL decomposition of the extended information matrix*
L	*triangular part of L'DL decomposition*
D	*diagonal part of L'DL decomposition of extended information matrix*
V	*information matrix*
	ARX factors in least squares representation
Eth	*point estimate of regression coefficients*
Cth	*covariance of regression coefficients*
cove	*point estimate of noise covariance*

Components	
com	*component*
coms	*array of components*
dfcs	*vector of degrees of freedom of components*
dfcs0	*initial degrees of freedom of components*
alphas	*normalized vector of degrees of freedom of components*
Com	*matrix ARX or ARX LS component*
Coms	*array of matrix ARX or ARX LS components*
Can	*component in matrix factorized ARX LS form*
Cans	*array of components in matrix factorized ARX LS form*
ychns	*modeled channels in component*
nychn	*number of modeled channels*

Mixtures	
`Mix`	*mixture estimate*
`Sim`	*mixture simulator*
`pMix`	*mixture predictor*
`pMixfix`	*mixture prediction*
`facs`	*list of factors*
`ncom`	*number of components*
`nchn`	*number of modeled channels*
`nfac`	*number of active factors*[a]

[a]dimensions are computed as :
`[ncom, nchn] = size(Mix.coms); nFacs = length(Mix.Facs); nfac = length(Mix.states.facs);`

Mixture estimation	
`frg`	*forgetting rate*
`frgd`	*default forgetting rate*
`rate`	*mixture flattening rate*
`maxtd`	*maximum time delay of factors in a mixture*
`nruns`	*number of runs in iterative mixture estimation*
`relerr`	*relative error*
`maxerr`	*maximum possible error*
	states in mixture estimation [a]
`faclls`	*trial factor predictions* $log(f(d_{t+1}\|fac, d(t)))$
`comlls`	*component predictions* $log(f(d_{t+1}\|com, d(t)))$
`mixll`	*mixture prediction* $log(f(d_{t+1}\|mix, d(t)))$
`comwgs`	*component weights*
`facwgs`	*factor weights*

[a]refer to mixupdt.m for meaning of the statistics

Mixture projection	
`pchns`	*predicted channels*
`cchns`	*channels in condition*
`psi0`	*value of zero-delayed regressor*

Advisory system design	
`aMix`	*advised mixture of the type ARX LS + control states*
`aMixu`	*desired mixture of the type ARX LS + control states*
`strc`	*common control structure*
`kc`	*lift of quadratic forms*
`UDc`	*cell vector of u'du decompositions of KLD kernels*
`udca`	*u'du decomposition of average KLD kernel in UDc*
`kca`	*average lift of quadratic forms kc*
`uchn`	*list of channels with recognisable actions*
`pochn`	*list of channels with o-innovations*
`outs`	*list of channels with innovations*
`npochn`	*number of channels with o-innovations*
`ufc`	*normalised vector qualifying components*
`nstep`	*parameter determining design horizon*

Structure estimation	
`maxstr`	*guess of the richest structure*
`maxFac`	*richest factor*
`maxMix`	*richest mixture*
`belief`	*belief on a guess of richest structure*
`chbelief`	*belief on factors of a channel*
`nrep`	*number of random starts*
`MAPstr`	*MAP estimate of the factor structure*

General cryptonyms	
`DEBUG`	*global debugging flag*
`chn`	*channel (data row)*
`std`	*standard deviation*
`pdf`	*probability density function*
`kld`	*Kullback-Leibler distance*
`ll`	*log of posterior likelihood on data: v-log-likelihood*
`niter`	*number of iterations*
`opt`	*option*
`options`	*computational options*
`seed`	*seed of random generator*
`sig`	*standard deviation of output noise*
`CUMTAB`	*transition table of components*
`ACTIVE`	*active component*

Agreement 14.1.3 (Software elements) *The following functions are distinguished in Mixtools.*

- Mixtools design base *is an implementation of basic algorithms that are at disposal to user for data processing.*
- Mixtools user's functions *are constructions exploiting Mixtools data base for data analysis.*

The following data processing is distinguished in Mixtools.

- *Batch processing that is invoked whenever the function is called with the argument "data" referring to the data processed or "ndat" denoting their number. Data have to be stored in the global variable "DATA" in the latter case.*
- *Recursive processing (≈ adaptive) is applied when neither "ndat" nor "data" are taken as argument of the called function.*

Factors in Mixtools correspond with the description of the posterior pdfs of unknown parameters and should not be mixed with parameterized factors. The following factors are distinguished at present.

1. *ARX factor with coded type = 1 corresponding to the normal parameterized factor, see Proposition 8.1.7.*
2. *ARX LS factor with coded type = 2 corresponding to the normal parameterized factor and having sufficient statistics in the least-squares form, see Proposition 8.1.7.*

   Components in Mixtools are list of factors for individual modelled channels. The following basic components are distinguished.
3. *ARX component, type = 11, – all factors are ARX factors, with type = 1.*
4. *ARX LS component, type = 12, – all factors are ARX factors, with type = 2.*

If moreover all ARX factors have a common regression vector, the matrix type *of components is also considered. These software constructs are used for input and output to data processing.*

- *matrix ARX component, type = 13, – matrix version mimic to the ARX factor of type = 1,*
- *matrix ARX LS component, type = 14, – matrix LS version mimic to ARX LS factor of type = 2, but regression coefficients and noise covariance estimates are matrices.*

A *mixture* is a structure. It is realized as a list of components together with *degrees of freedom of components*, i.e. the vector statistic κ determining Dirichlet pdf (6.3).

Mixture may contain components of different type – then its type = 0. The following special cases are supported if all components are of the same type

- ARX mixture type = 21, all components have type = 11,
- ARX LS mixture type = 22, all components have type = 12,
- matrix ARX mixture, type = 23, all components have type = 13,
- matrix ARX LS mixture, type = 24, all components have type = 14.

Prediction counter-parts of the software element describing estimates are given the same names prefixed by p- so we have *p-factors*, *p-components* and *p-mixtures*. Types of the p-elements are obtained by adding 100 to the codes of estimation counterparts. Thus,

type code	*software element*
101	ARX factor,
102	ARX LS factor,
111	ARX component,
112	ARX LS component,
113	matrix ARX component,
114	matrix ARX LS component,
121	ARX mixture,
122	ARX LS mixture,
123	matrix ARX mixture,
124	matrix ARX LS mixture.

14.1.2 Relationships of the term base to theory

The following table lists terms used and their relationships to those of the present text.

name	**description**	**counterpart**	**reference**
Com	ARX component	estimate of normal component	Chapter 8 *GiW* pdfs, Proposition 8.1.7
Coms	array of ARX components	list of Com	
Cth	covariance of regression coefficients	covariance factor	(8.19)
DATA	data sample	$d(\mathring{t})$	Chapter 5
DEBUG	global debugging flag	debugging tool	
D	diagonal part of $L'DL$ decomposition	D	Section 8.1.3
Eth	point estimate of regression coefficients	$\hat{\theta}$	(8.18)
Fac	factor	$f(\Theta_{ic})$	*GiW* pdf, Proposition 8.1.7
Facs	array of factors		
LD	$L'DL$ decomposition of extended information matrix	L with I replaced by D	Section 8.1.2 Section 8.1.3
L	triangular part of $L'DL$ decomposition	L	Section 8.1.3

name	**description**	**counterpart**	**reference**
MAPstr	MAP estimate of the factor structure		Section 8.6
Mix	mixture estimate	$f(\Theta)$	Section 8.5
Psi	data vector	Ψ	(8.2)
Sim	mixture simulator	sharp $f(\Theta_{ic})$	
TIME	processing time	t	Chapter 2
V	extended information matrix	V	Section 8.1.3
belief	belief on a guess of maximum richest structure		Section 6.6
chn	channel (data row)	pointer to $d_{i;t}$	Agreement 5.2.1
com	component	list of factors	Agreement 5.2.1
comlls	component log-v-likelihood log(f(dt | com))	$\ln(f(d(t)\|c))$	Algorithm 6.5.2
coms	array of components		
comwgs	component weights	$w_{c;t}$	Algorithm 8.5.1
cove	point estimate of noise variance	$\mathcal{E}[r\|d(t)]$	(8.23)
dfcs	vector of degrees of freedom of the components	$\kappa_{c;t}$	Algorithm 6.5.2
dfm	degrees of freedom of factor	$\nu_{ic;t} - 2$	Proposition 8.1.7
fac	position of a factor in an array of factors	i in $d_{i;t}$	
faclds	virtual factor predictions log(f(dt | fac))	$\ln(f(d_{i;t}\|\psi_{ic;t}))$	(8.40)
facwgs	factor weights	$w_{ic;t}$	(8.74)
frg0	default forgetting rate	λ	Proposition 8.1.11
frg	forgetting rate	λ	Proposition 8.1.11
iter	iteration counter	counter n	Section 6.1.3
kld	KL distance	$\mathcal{D}(f\|\|\bar{f})$	(2.26) Propositions 8.1.9, 12.1.2
ll	log-v-likelihood	$\ln(f(d(\mathring{t})\|v))$	Section 6.1.2
mapping	components where factor is used	description of the sets c_i^*	see formula (6.87)
maxFac	richest factor		Section 8.6
maxMix	richest mixture		Section 8.6
maxstr	guess of the richest structure		Section 8.6
maxtd	maximum time delay of factors in a mixture	$\max_{c \in c^*} \partial_c$	Agreement 5.2.1
mixll	log-v-likelihood log(f(dt | mix))	$\ln(f(d(\mathring{t}))$	Section 6.1.2
nFacs	number of factors		
nPsi	length of data vector	$\mathring{\Psi}$	Agreement 5.2.1
nbest	number of "best" MAP structures stored		Section 6.6
nchn	number of modelled channels	$\mathring{d}$	Chapter 2
ncom	number of components	$\mathring{c}$	Agreement 5.2.1
ndat	length of data	$\mathring{t}$	Chapter 2
nfac	number of active factors		
niter	number of iterations	$\mathring{n}$	Section 6.1.3
npsi	length of regression vector	$\mathring{\psi}$	Chapter 2
nrep	number of random starts		Section 6.6
nychn	number of modelled channels	$\mathring{d}$	Chapter 2

name	description	counterpart	reference
pdf	probability density function	$f(\cdot\|\cdot)$	Chapter 2
psi	regression vector	ψ	(5.11)
std	standard deviation		
str	factor structure	phase form	Agreement 5.2.1
ychn	modelled channel	$\mathring{\Delta}$	
ychns	modelled channels in component		

14.1.3 Mixtools tasks and functions

Here, individual Mixtools functions are related to the basic theory. It should help in understanding their role and use.

Mixtools tasks

We list the basic tasks that are gradually covered by Mixtools for all considered models. All of them are implemented for normal mixtures. The tasks are

Data preprocessing that includes data transformation (scaling shifting, reducing), noise and outlier reduction, check for physical boundaries, replacement of missing data etc., Sections 6.2, 8.2, 10.2.

Prior knowledge processing that means mainly quantification of expert knowledge, Sections 6.3, 8.3, 10.3.

Structure estimation that selects the best structure of the mixture model used, Sections 6.6, 8.6, 10.6.

Initialization that selects iteratively a suitable prior pdf on all parameters of the mixture, Sections 6.4, 8.4, 10.4.

Mixture estimation that provides a “final” off line estimate of the mixture on preprocessed data and for the structure chosen, Sections 6.5, 8.5, 10.5.

Design of the advisory system that has to analyze dangerous components, propose advising strategy, including the communication with user, Chapters 7, 9, 11.

Design verification that judges the achieved quality of the proposed advisory system before using it practice.

Export of the advisory system that provides, in the form acceptable by “industrial” software systems, the data files describing the estimated and designed advisory mixtures as well as on line advising algorithms describing the tuned advisory system, Section 14.2.

Obviously, the whole processing is expected to iterative. The logic of iterations is far from being trivial. Thus, a standard processing style is gradually created with the aim to let the user handle exceptions only. It gradually distinguishes *Mixtools user's functions* and from *Mixtools design base*. The list of functions below gives the overall impression on the supported tasks. For a detailed description, see the guide [101] and the helps of functions as well as available examples of their usage.

Constructors	
`Fac = facarx(ychn, str)`	build ARX factor
`Fac = facarxls(ychn, str)`	build ARX LS factor
`Com = comarxls(ychns, str)`	build matrix ARX LS component
`Com = comarx(ychns, str)`	build matrix ARX component
`Mix = mixconst(Facs, coms, dfcs)`	build ARX or ARX LS mixture
`Mix = mixconst(Coms, dfcs)`	build mixture of arbitrary type

Initialization of estimation	
`Mix = ... mixinit(Mix0,frg,ndat,niter,opt,belief)`	initialization of mixture estimation[a]
`Mix = comdel(Mix, com)`	cancel specified component
`Mix = commerge(Mix, Mix0, com)`	merge mixture components
`Mix0 = genmixe(ncom,ychns,str,ndat)`	generate initial mixture

[a]estimation options: 'q', 'b', 'f', 'm' 'n'+ number of iteration steps; belief expresses user's belief into the regressor specified

Estimation operations	
`Mix = ... mixest(Mix0, frg, niter, opt)`	iterative mixture estimation[b]
`Mix = mixestim(Mix0, frg, ndat)`	quasi-Bayes mixture estimation
`Mix = mixestim(Mix0, frg)`	recursive quasi-Bayes mixture estimation
`Mix0 = mixflat(Mix)`	mixture flattening
`Mix = mixstats(Mix, ndat)`	compute estimation statistics for a fixed Mix
`Mix = mixstats(Mix)`	compute statistics recursively for a fixed Mix
`Mix0 = genmixe(ncom, ychns, str)`	generate initial mixture for estimation

[b]opt - options: 'q', 'b', 'f','m' for quasi-Bayes, batch Bayes, forgetting branching and estimation with fixed covariances

Prediction operations	
`pMix = mix2mixm(Mix, pchns)`	build marginal predictor
`pMix = mix2pro(Mix, pchns, cchns)`	build/re-build mixture projector
`pMix = profix(pMix, psi0, pre)`	build mixture prediction from projector
`pMix = ... mixpro(Mix0,pchns,cchns,psi0,pre)`	build mixture projection[a]
`[pMix, weights] = ... profixn(pMix, psi0, pre, nstep)`	prediction n-steps ahead [b]

[a]defaults: pchns - [1,2], cchns - no, psi0 - substituted from DATA, no data scaling
[b]the weights are data dependent even for static mixtures

Visualization [a]

[a]defaults: see Prediction operations. The functions allow definition of grid densities and ranges

`mixplot (Mix,pchns,cchns,psi0,pre)`	mixture plot (shaded)
`mixplotc(Mix,pchns,cchns,psi0,pre)`	mixture plot (contours, components)
`[x,y,z] =... mixgrid(Mix,pchns,cchns,psi0,pre)`	coordinates for mixture plot
`[x,y,z] = datagrid(Mix)`	coordinates for data plot
`datascan(chns)`	scan data for 2 dim clusters
`mixmesh(Mix,pchns,cchns,psi0,pre)`	mixture mesh plot
`mixscan(Mix,chns,pre)`	scan mixture for 2 dim. clusters
`setaxis(list, ax)`	set global axis in subplots[a]
`sigscan(chns)`	scan signal

[a]list is list of subplots, ax is a scaling see axis function

Interactive visualization	
`mixshow(Mix)`	interactive plot of mixture
`mixbrow(Mix)`	interactive display of mixture attributes
`setdbg('function')`	interactive setting of "dbstop"

Data preprocessing	
`pre = preproc(pre)`	preprocess data
`pre = preinit(pre)`	initialize preprocessing
`pre = prestep(pre)`	preprocessing step

Structure estimation	
`Mix = ...`	
`mixstrid(Mix,Mix0,belief,nruns)`	estimate mixture structure
`MAPstr = ...`	estimate structure of a factor
`facstrid(Fac,Fac0,belief,nbest,nruns)`	

Mixture simulation	
`mixsimul(Sim, ndat)`	batch mixture simulation
`mixsimul(Sim)`	recursive mixture simulation
`Sim = statsim((ndat, ncom, cove)`	create static mixture with components on unit circle

Basic conversion functions	
`LD   = ltdl(V)`	decompose positive definite matrix to L'DL
`Mix  = mix2mix(Mix, form)`	convert mixture to a specified form[a]
`Com  = com2com(Com, form)`	convert component into a specified form
`X    = arx2arx(X)`	convert between ARX and ARX LS representations

[a] "form" is a coding summarized in "Codes"

Design of advisory system	
`[aMix, aMixu] = ...`	
`inisyn(Mix,Mixu,pochn,uchn)`	initialize advisory design for normal mixture
`[aMix, aMixu] = ...`	
`inisyn(Mix,Mixu,Chns)`	call with channel descriptions
`aMix  = ...`	
`aloptim(aMix,aMixu,ufc,nstep,chis)`	perform academic design for normal mixture
`ufc   = ufcgen(aMix, aMixu)`	generate normalized vector qualified components
`aMix  = ...`	
`soptim(aMix,aMixu,ufc,nstep,chis)`	perform +simultaneous advisory design for normal mixture
`aMix  = algen(aMix,aMixu,ufc)`	compute of probabilistic weights for advisory design
`[Mixu, ychns] = target(Chns)`	create user's target mixture
`Mix   = mixcopy(Mix1, Mix2)`	copy of ARX or ARX LS statistics

Channel descriptions	
`Chns = chnconst(chns)`	build channel descriptions
`Chns = chnset(Chns,chns,fld, val)`	set channel descriptions field
`val  = chnget(Chns,chns,fld)`	get values of channel descriptions fields`Chns  =...`
`scaleDescription(Chns,pre)`	scale channel descriptions according to preprocessing chosen

General purpose functions	
`prodini`	standard Mixtools session beginning
`prt(X)`	debugging prints
`is = equal(X1,X2, eps)`	test of equivalence up to a small difference
`str = genstr(order, nchn, td)`	generation of model structure of given order
`is = streq(str1, str2)`	compare two structures
`is = isstatic(Mix)`	test whether mixture is static
`is = isdimeq(X1,X2)`	test of equality of dimensions
`is = streq(str1,str2)`	test of equality of structures
`mversion`	display current Mixtools version

14.1.4 Off line GUI

The graphical user interface (GUI) is implementation dependent. Examples of the GUI used in the project pilot applications are in Chapter 15, Figures 15.4, 15.6.

A general experimental GUI framework has been designed that covers the expected applications. An example of its visual aspect is in the Fig. 14.1. It consists of two parts: the left-hand side serves for imitating the operator actions, the right-hand side is the one-dimensional version of the generic GUI provided to users, cf. left-hand side of Figure 15.4. Its two-dimensional version seen there on right-hand side can also be used in off line mode.

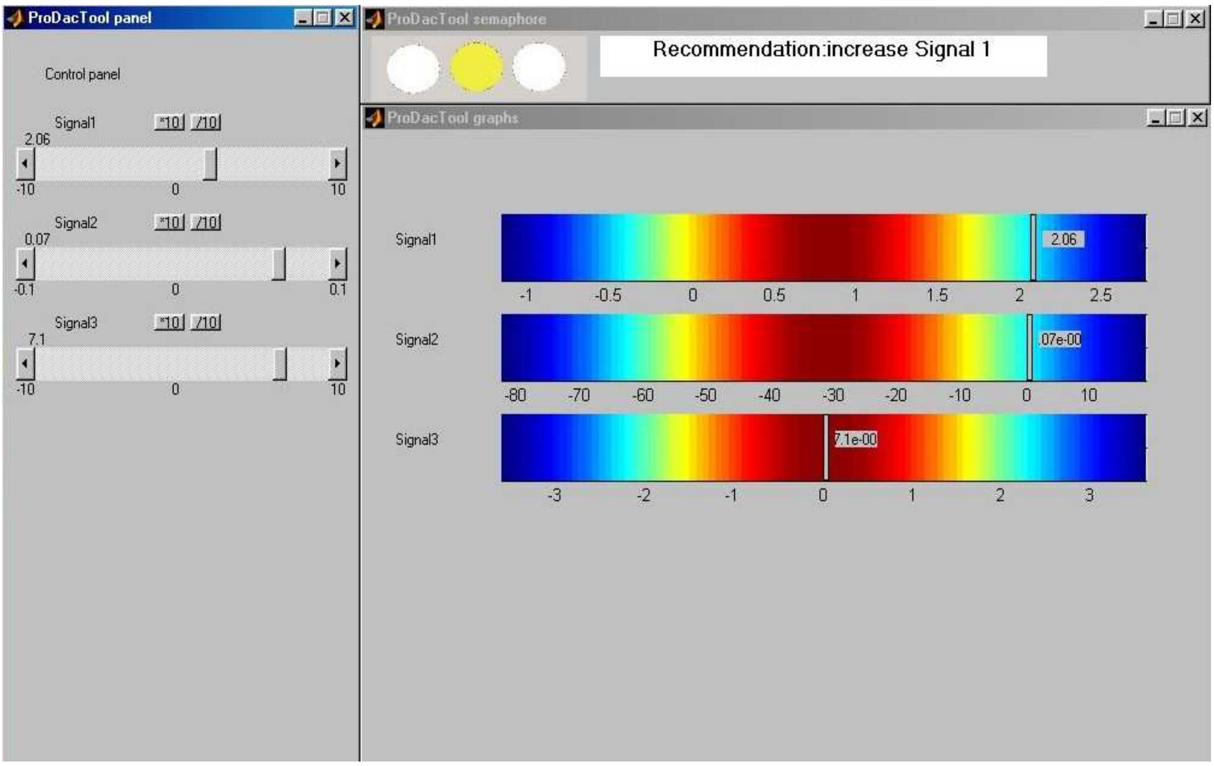


Figure 14.1: Screen-shot of GUI used for off line experiments. Left-hand part is active and right-hand one passive part of GUI, respectively.

14.2 Bridge Matlab – ANSI C

Matlab is used as the development environment but applications must not rely on it – an ANSI C code should be available at the implementation level.

The Matlab functions can be compiled by the Matlab tool "Matlab Compiler". This solution is expensive and platform dependent (MathWork supplied libraries) and does not cover platforms that are expected.

The solution developed has three steps. It is referred to as *Matlab – ANSI C bridge*.

- The Mixtools processing is fully converted into C-encoded mex versions.
- A library of functions is designed that makes the same processing as a subset of *Matlab API* [106] Application Program Interface provided by MathWorks.
- Appropriate data conversions between Matlab and ANSI C and vice versa are implemented.

It is possible to compile the MEX source files using the standard ANSI C compiler without rewriting them. We have at least two reasons for it. The run-time library independent of Matlab has to be available and the PVM system, Section 14.3, used for parallelization, supports C, C++, or Fortran languages. Because a target platform is unknown we decided to use ANSI C language to implement final applications. It guarantees software and data portability.

The data conversion between Matlab and ANSI C on different computer platforms is an additional problem. Routines in Matlab and in ANSI C have been prepared to this purpose.

The current version of the ANSI C support for the *Mixtools Toolbox* requires Matlab version 5.x Release 11 (Win32 or Unix) or higher and the corresponding ANSI C compiler.

14.2.1 Structure *mxArray*

The Matlab language works with a single object type, the *Matlab array.* All Matlab variables, including scalars, vectors, matrices, strings, cell arrays, structures, and objects are stored as Matlab arrays. In C, the Matlab array is declared to be of type `mxArray`. We have designed our own specific – Matlab compatible – structure `mxArray`.

The `mxArray` structure should carry the following information.

- type of array
- dimensions
- the data associated with this array
- distinction whether the array is real or complex
- indices and maximum number non-zero elements if the array is sparse
- the number of fields and field names if structure or object is represented.

Under the current agreement of the development software group about the allowed data types and structures, we could omit some of these requirements.

Matrices in the Mixtools are of type *double* (double precision floating-point number) and have dimensions m-by-n, where m is the number of rows and n is the number of columns. Only a simple Matlab structures (1-by-n) and cell vectors (1-by-n) are used. All MATLAB data are stored column-wise similarly as Fortran matrices.

It corresponds the following definitions.

```
/* mxType defines available types of data contained in mxArray */
typedef enum {
  mxREAL = 1,              /* real matrix */
  mxCOMPLEX = 2,           /* complex matrix - reserved */
  mxCHAR = 5,              /* char string - reserved */
  mxCELL = 11,             /* Matlab CELL structure */
  mxSTRUCT = 12            /* Matlab STRUCT structure */
} mxType;
```

```
/* structure mxArray can hold compatible types of Matlab data;     */
/* this structure should not be accessed directly, use mxFunctions */
/* to access data contained in this structure;                     */
/* the following data types are currently supported:               */
/*   - m by n real matrices                                        */
/*   - 1-dimensional cell arrays                                   */
/*   - 1-dimensional structure arrays                              */
/*     (cell and structure array are equivalent)                   */
typedef struct mxarray
{  int     M;                     /* number of rows */
   int     N;                     /* number of columns */
   int     type;                  /* mxArray type */
   double  *pr;                   /* pointer to data if numeric */
   struct  mxarray **fields;      /* fields or cells if not numeric */
}  mxArray;
```

The complex matrices and character array or string are not supported at present. The structures and routines are prepared for the later extensions.

We have designed and implemented basic `mxArray` dependent *mxFunctions*, listed here:

```
/*     === API supported equivalents ===     */

#define mxCreateDoubleMatrix(M, N, T)      (_alloc_array (M, N, mxDOUBLE))
#define mxCreateCellArray(M, N)            (_alloc_array (1, N, mxCELL))
#define mxCreateCellMatrix(M, N)           (_alloc_array (1, N, mxCELL))
#define mxCreateStructMatrix(M, N, NF, X)  (_alloc_array (1, NF, mxSTRUCT))

mxArray *mxCreateScalarDouble (double value);

#define mxIsDouble(X)  ((X) ? ((X->type == mxDOUBLE)  ? 1:0):0)
#define mxIsComplex(X) ((X) ? ((X->type == mxCOMPLEX) ? 1:0):0)
#define mxIsNumeric(X) ((X) ? ((X->type == mxDOUBLE)  ? 1:0):0)
#define mxIsCell(X)    ((X) ? ((X->type == mxCELL)    ? 1:0):0)
#define mxIsStruct(X)  ((X) ? ((X->type == mxSTRUCT)  ? 1:0):0)

int mxGetM (const mxArray *array_ptr);
int mxGetN (const mxArray *array_ptr);
void mxSetM (mxArray *array_ptr, int m);
void mxSetN (mxArray *array_ptr, int n);

double *mxGetPr (const mxArray *array_ptr);
#define mxGetPi(P)                         (mxGetPr (P))
void mxSetPr (mxArray *array_ptr, double *pr);
#define mxSetPi(P, X)    (mxSetPr (P, X))

double mxGetScalar (const mxArray *array_ptr);

mxArray *mxGetCell (const mxArray *array_ptr, int index);
void mxSetCell (mxArray *array_ptr, int index, mxArray *value);

#define mxGetFieldByNumber(X, I, F)        (mxGetCell (X, F))
#define mxSetFieldByNumber(X, I, F, V)     (mxSetCell (X, F, V))

int mxGetNumberOfElements (const mxArray *array_ptr);
```

```
void mexErrMsgTxt (const char *error_msg);

void mexFunction (int no, mxArray **out, int ni, const mxArray **in);


/*     === API functions that manipulate memory ===     */

void *mxCalloc (size_t n, size_t size);
void mxFree(void *ptr);

#define mxDuplicateArray(X)               (_dupl_array (X))
#define mxDestroyArray(X)                 (_free_array (X))


/*     === MixTools additional functions ===     */
/*      not defined in original Matlab API       */

#define mxCreateCell(N)                   (_alloc_array (1, N, mxCELL))
#define mxCreateStruct(N)                 (_alloc_array (1, N, mxSTRUCT))
#define mxGetType(X)                      ((X) ? (X->type) : 0)
#define mxGetFieldN(X, i)                 (mxGetCell (X, i))
#define mxGetFieldPr(X, i)                (mxGetPr (mxGetCell (X, i)))
#define mxSetFieldN(X, i, V)              (mxSetCell (X, i, V))

void mxSetScalar (mxArray *array_ptr, const double value);
void mxUpdtFieldN (mxArray *Xt, int field, const mxArray *Xs);


/*     === MixTools data handling functions ===     */

mxArray *mixload (const char *filename);
int mixsave (const mxArray *mix, const char *filename);

mxArray *dataload (const char* filename);
int datasave (const mxArray *data, const char *filename);

/*     === auxiliary debug functions ===     */

#define mixprint(X) (_array_print (X))
void _array_print (const mxArray *array_ptr);

void prt (const char *msg, const mxArray *X);
void prtmx (const mxArray *X);

/*     === auxiliary non-standard declarations ===     */

#define LEN(X)                            ((X) ? (X->M * X->N) : 0)

void error (const char *msg);
```

The most of function definitions correspond the standard Matlab API with the restrictions given in the previous paragraph. For complete description of these functions see *Matlab Documentation* [106]. Some non-standard functions are required by MixTools. For the information about implementation and description of these functions in the context of MixTools, see the ANSI C source files (`mex.h` and `mex.c`).

14.2.2 Data conversion

If we talk about data in connection with MixTools we have to consider two kinds of data, namely, mixture *structures* and *real data*.

We denote data *structures* as data used during computation that describe structure of the probabilities, mixtures, factors, etc. Data *structures* correspond the `mxArray` data types. Data collected at an application site that are used as a system input we call simply *data*. We have to assure portability both for structures and for real data.

Data formats

Each element of the binary file is represented as *double* format including dimensions. The necessary conversions are made inside conversion functions.

The structures are stored in the *Mixture Definition Format* (MDF) (see Fig. 14.2) in the binary file. In fact, the mixture structure consists of *cells* and *matrices* that have to be saved in the binary file. The individual parts of the mixture structure in the binary file are represented by its type, dimensions, and matrix if the type contains data. The type can be one of the `mxType` values defined in the file `mex.h`. The dimensions are lengths of the cell array, $M = N$, if the type is cell. In fact, dimensions are 1-by-N but we also set M to the value of N to achieve compatibility with some routines. In the case of matrix, M is the number of rows and N it the number of columns of the matrix. If the data type is matrix, the matrix values follow the cells with dimensions. They are represented as linear array.

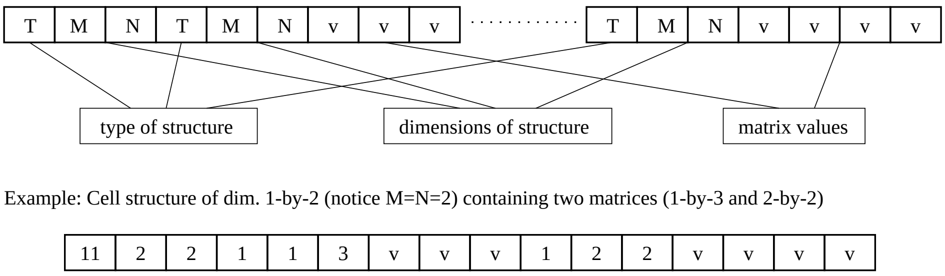


Figure 14.2: Mixture Definition Format

The data are stored in the *Data Definition Format* (DDF) (see Fig. 14.3) in the binary file. The first two elements of the file represent dimensions of the data matrix. The first is M, which is number of rows (channels) and the latter is N, which is number of columns. The remaining part is data matrix itself stored as linear array organized column-wise.

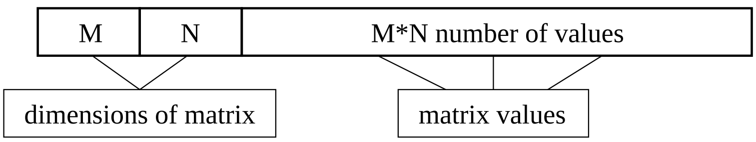


Figure 14.3: Data Definition Format

Matrices both in MDF and in DDF files are stored column-wise as common in Matlab environment. Be careful if you are using these matrices in C language.

Matlab functions

The following functions extends standard *MixTools Matlab Toolbox* [12] and bring us possibility to work with the mixtures in the plain C environment. These functions are used to load or save a fragment of the Matlab workspace from or to the hard drive as a binary file. Using these functions, it is possible to load or save any `mxArray` structure from or to the binary file.

The *mixload()* function reads the content of the binary file and load the data from MDF-file to the Matlab mixture structure

```
Mix = mixload(filename, machineformat, prec)

      filename      - name of the input binary file
      machineformat - target hardware platform
      prec          - number precision
```

The parameter `machineformat` specifies in what type of binary representation data are stored in the file. The default value is `native`, see Matlab documentation [106] for information about available formats. The parameter `prec` should always be `double`, it is default.

The information of variable names *(field-names)*, as it is common in Matlab, is not stored in the MDF-file. This information is superfluous during C-processing but not during Matlab processing. If the mixture structure is loaded from the file, the individual fields of the structure do not hold this kind of information. Thus, it is necessary to provide this structure its field-names. The function *rename()* is used for this purpose

```
Mix = rename(source, template)

      source   - mixture to be renamed
      template - mixture template
```

The parameter `source` should be the mixture returned by *mixload()* function. The parameter `template` should be the original Matlab mixture used as a template. It is recommended to save the original mixture to the MAT-file and to use it as a template.

The *mixsave()* function writes a mixture structure to the binary MDF-file

```
mixsave(Mix, filename, machineformat, prec)

      Mix           - Matlab mixture structure
      filename      - name of the output binary file
      machineformat - target hardware platform
      prec          - number precision
```

The parameter `Mix` specifies the Matlab mixture structure to be saved to the binary file. For explanation of other parameters see *mixload()* function above.

The *dataload()* function reads data from the binary DDF-file

```
data = dataload(filename, machineformat, prec)

      filename      - name of the output binary file
      machineformat - target hardware platform
      prec          - number precision
```

For explanation of the parameters, see *mixload()* function above.

The *datasave()* function writes data to the binary DDF-file

```
datasave(data, filename, machineformat, prec)

      data          - real DATA
      filename      - name of the output binary file
      machineformat - target hardware platform
      prec          - number precision
```

The parameter `data` specifies real data used by MixTools to be saved to the binary DDF-file. For explanation of other parameters, see *mixload()* function above.

ANSI C functions

The function *mixload()* loads the mixture structure from the binary MDF-file

```
mxArray *mixload (const char *filename);

       filename - name of the input binary file
```

The function returns a pointer to the loaded `mxArray` if it is successful, otherwise, it returns NULL pointer.

The function *mixsave()* saves the mixture structure to the binary MDF-file

```
int mixsave (const mxArray *mix, const char *filename);

       mix      - structure which points to the mixture to be saved
       filename - name of the output binary file
```

The function returns 0 (zero) if it is successful, otherwise, it returns -1.

The function *dataload()* loads the data from the binary DDF-file

```
mxArray *dataload (const char* filename);

       filename - name of the input binary file
```

The function returns a pointer to the loaded `mxArray` if it is successful, otherwise, it returns NULL pointer.

```
int datasave (const mxArray *data, const char *filename);

       data     - structure which points to the data to be saved
       filename - name of the output binary file
```

The function returns 0 (zero) if it is successful, otherwise, it returns -1.

It is useful to see source files `mex.h` and `data.c` for the specifics of implementation and examples.

14.3 Off line parallel processing

Dimensionality of the problem calls for an efficient use of computing power especially in the off line mode where many alternatives on large data sets should be processed. For this reason, parallel processing have been considered as a way of increasing the computer power used. Parallel Virtual Machine (PVM) was selected as a system tool allowing to combine rather heterogeneous computational hardware. The current section just gives a flavor on the problem solved. Details on PVM and parallelization can be found in [103, 107].

Efficiency of approximate estimation algorithms, Section 6.5, decides on efficiency of overall off line processing, at least its learning part. Section 14.3.1 indicates how its parallelization can be performed.

14.3.1 Computation analysis

Forming the mixture, we consider units or tens of components and tens or hundreds of factors.

Looking at the estimation algorithms, we find that the critical steps of the algorithm – from the computational viewpoint – are computations of factor weights (predictions) $F_i = f(d_{ic;t+1}|d(t), c)$ and the weighted updating of parameter estimates $f(\Theta_{ic}|d(t)) \rightarrow f(\Theta_{ic}|d(t+1))$.

Both these steps coincide with recursive least squares when dealing with normal mixtures, Chapter 8, requiring $\mathcal{O}(\mathring{\Psi}^2)$ operations for each factor. At the same time, both predictions and updating can be performed independently if they have access to the relevant data vectors $\Psi_{ic;t}$ and get/send from/to other factor computations just the values of F_i.

The other steps of the estimation algorithms are much less complex and can be performed sequentially.

14.3.2 Communication model

The master-slave model, see Fig. 14.4, of virtual parallel machine architecture has been chosen. A separate "control" program termed the master is responsible for process spawning, initialization, collection of results, and timing of functions. The slave programs perform the computation allocated to them by the master. Thus, the master is responsible for load balancing.

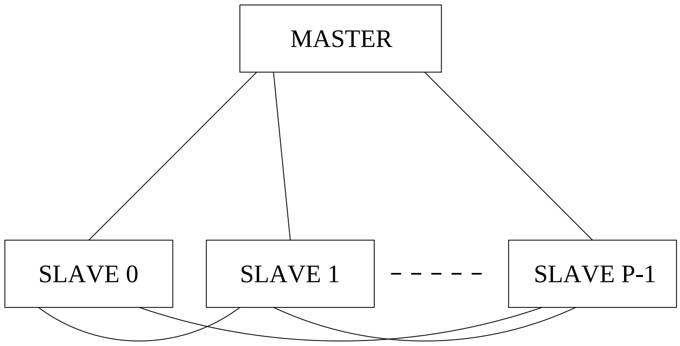


Figure 14.4: The Virtual Machine Communication Model

The complete graph inter-connection network is used. During computation, individual slaves are able to communicate both with the master and with each other. Because of the facilities of the PVM, it is relatively straightforward to implement this network inter-connection.

Before the slave computations are performed, the master has to do a work that can be termed "workload allocation". The master establishes connections between individual nodes and performs data decomposition or partitioning. This is done statically, i.e. each process knows *a priori* its share of workload.

14.3.3 Data decomposition

The master process estimates parameter and handles all operations over a mixture, i.e. it follows the sequential algorithm and it works as a common sequential program. Once this program reaches the point where computations can be done consequently for several factors, it becomes a real *master*.

During the time when the slaves are active, the master performs synchronization. Then the master collects and evaluates results. It can follow the sequential algorithm, compute some additional statistics, or send the results to other separated modules of the system. One step of the parallel recursive mixture estimation is depicted in Figure 14.5.

Suppose we use P slaves to compute values of probability for N factors. Then each slave will perform operations on $\lceil \frac{N}{P} \rceil$ or on $\lfloor \frac{N}{P} \rfloor$ factors, i.e. the i-th factor F_i is computed on the slave $p = i \bmod P$, where $i \in i^* = \{0, \ldots, N-1\}$ and $p \in p^* = \{0, \ldots, P-1\}$.

In the case of computing the values of the predictive pdfs F_i for each factor, the "primitive" parallelism can be used. This is because minimal communication is needed between individual slaves. An individual slave process, computing the value of predictive pdf F_i, does not require any additional information. It acquires or receives real *data* record d_{t+1} from memory. Then the slave computes required value and sends the result back to the master.

The situation, in the case of updating Bayesian parameter estimates of factors, $f(\Theta_{ic}|d(t+1))$, is similar to that discussed in the previous paragraph. The slaves have to be provided by information about weights, $w_{ic;t+1}$, for individual factors but information about $w_{\tilde{c};t+1}$ is contained in the initial data. The individual factor weights, $w_{ic;t+1}$, are computed by the master and sent to the slaves. The slaves then update parameters of factors, perform forgetting, and send result to the master.

As mentioned above, when the slaves are active, the master computes some additional statistics and performs synchronization. Then the master collects and evaluates results. It can proceed to new iteration cycle, follow the sequential algorithm, or provide the results to other separated modules of the system.

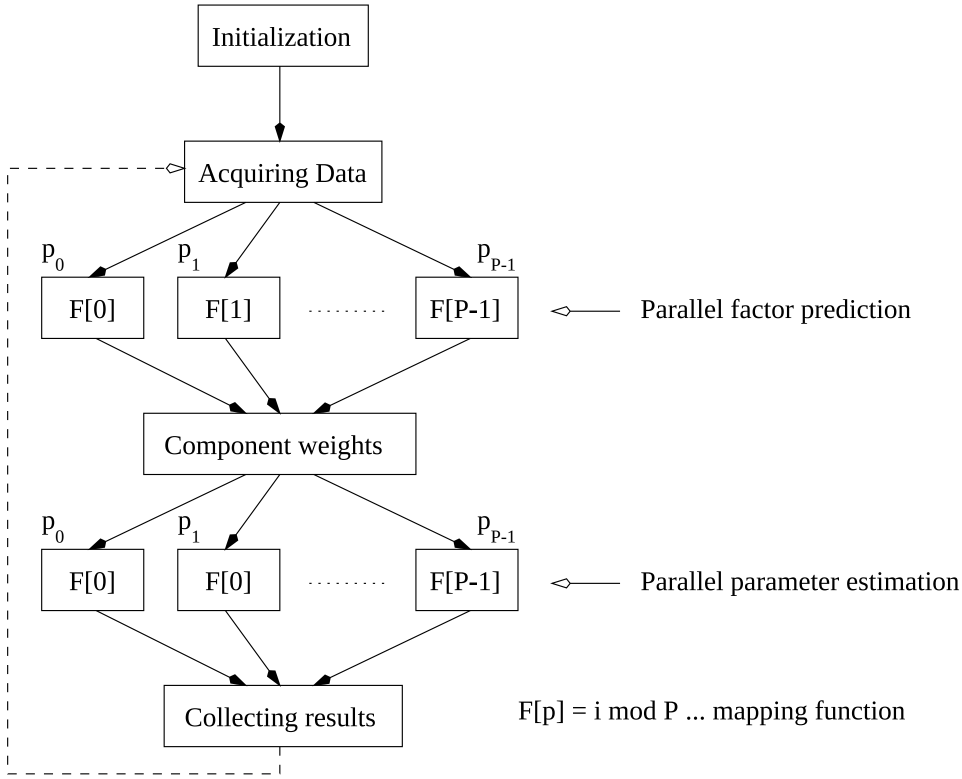


Figure 14.5: One step of the parallel recursive mixture estimation

14.3.4 Parallelization example: mixture estimation

Here, we describe the parallel mixture estimation algorithm. We also outline how communication is running during the computation. The algorithm and communication is depicted in Figure 14.6. The master and one slave are just presented for simplicity.

The symbols used in Figure 14.6 are explained in the sections above or in the guide [101].

After the initialization of processes – starting PVM, spawning slaves, and loading input mixture – the slaves must be provided by information about common variables. They are forgetting factor, start time, stop time, and data file name. Each slave loads data into its data space in this stage of the project. Alternatively, shared data manager can be used to this purpose.

Then, the master sends information about factors to the slaves. In the normal case, they consist of actual factor index, output channel, regression vector structure, degrees of freedom, and information matrix.

After these initialization actions mixture estimation can start. It begins from $TIME = starttime$, given by $maxtd + 1$ (see [12]) until $stoptime$ is reached.

First, factor predictions, $F_i = f(d_{ic;t+1}|d_{i+1\_\mathring{d};t+1}, d(t), c)$, are computed by slaves and results are sent to the master. The master then computes component weights and factor weights, $w_{ic;t+1}$, that are sent back to the slaves. This information about factor weights is required to update factor parameters. The slaves update factor estimates, $f(\Theta_{ic}|d(t))$, as soon as they receive appropriate factor weights. The filtration errors $facpes$ are then sent to the master as the result of factor parameters' update. In the meantime, the master computes some additional statistics used for post-processing of mixtures.

The last step of the mixture estimation iteration cycle is the updating of $\kappa_{c;t}$, mixture forgetting on the side of the master and factor forgetting on the side of the slaves. Then, the next iteration cycle starts.

After the mixture estimation finishes the slaves send updated factors to the master and the resulting mixture is saved to the file. The master stops PVM system and exits.

If we consider that the computational complexity of prediction and estimation steps is $\mathcal{O}(\mathring{\Psi}^2)$, using the parallelization with P processes, we can theoretically achieve linear speedup, $\mathcal{O}(\frac{\Psi^2}{P})$, in solving this

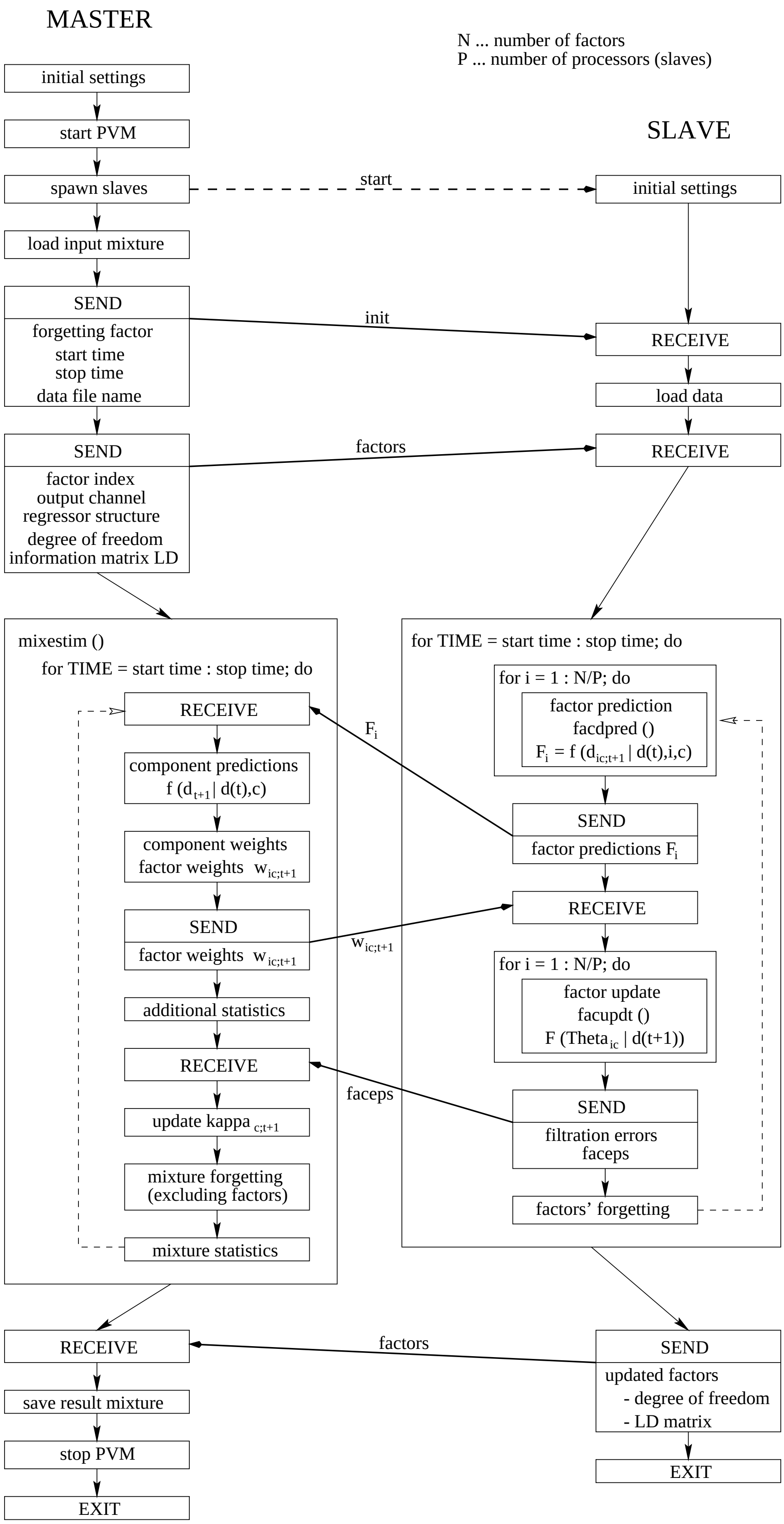


Figure 14.6: Parallel mixture estimation algorithm

part of the problem. The practical speedup is smaller due to the sequential part of the algorithm that is not parallelized and due to overhead caused by communication between the master and the slaves. The slaves do not communicate among each other.

There is a minimal overhead caused by the communication of slaves. But, we have to realize that we have not attempted to parallelize all steps of the algorithm. Thus, we cannot achieve linear speedup of the algorithm as a whole. As implied by Amdahl's law [108], speedup is at most $\frac{1}{h_s}$, where h_s is a relative sequential part of the algorithms.

14.4 On line processing

14.4.1 On line tasks and functions

Except of Matlab graphics, the Mixtools processing has been completely rewritten in C language. Thus, all functions are available for on line use and adaptation of advisory system.

The data management system used in applications is the only application-specific task. The Mixtools are designed and implemented so that the only function that must be changed is the function "*getdvect*". It returns the data vector Ψ at the given processing time and for the considered predicted factor output and the structure of the regression vector, Agreement 5.2.1. Thus, the load related to the connection of the specific data management to Mixtools is negligible comparing to the preparation adequate graphical outputs and other "ordinary" technical problems. Even the GUI part is relatively simply due to the common nature of the presented information (marginal pdfs of the advising mixture, signalling lights).

On line tasks

The tasks that are running on line in an advisory system differ according to the adopted version of the advisory system. For fixed advisory system, it is just

- data management (data sampling, preprocessing),
- evaluation and graphical presentation of low dimensional projections possibly driven by evaluated priority and signaling actions, see Chapters 7, 9, 10,
- evaluation of quality of advising.

For adaptive advisory system, the on line implementation uses, moreover,

- recursive mixture estimation, Sections 6.5. 8.5, 10.5,
- re-design of advisory part, typically, in its IST version, see Chapters 7, 9, 10.

Note that performance of the functions considered for on line use is carefully optimized.

Chapter 15

Competitors and applications of the advisory system

This chapter relates the proposed advisory system to the state of the art and provides samples of applications we are working on.

First, an overall view on the state of the art in operator support is presented, Section 15.1. Then, samples of an extensive set of products and services available on the World market are given. Their scanning has helped us to recognize unique features of our approach and its limitations, too, Section 15.2.

Theory and algorithms presented in this work are believed to be original ones. Naturally, majority of particular building blocks have been used before and many of them are treated in a much deeper way than here. The most common building blocks are commented in Section 15.3. Then, three rather different applications are characterized, Section 15.4. Practical conclusions are made in Section 15.5.

15.1 State of the art

Scanning of extensive data bases reveals that operator support is

- timely topic reflected in hundreds of papers [109];
- predominantly connected with chemical industry and nuclear power plants [110];
- bringing substantial savings [111];
- oriented usually to highly qualified operators [112];
- prepared mostly for very specific applications, where the expensive tuning pays back [113, 114];
- caring significantly about abilities of graphical user interface and software architecture [115] while standard algorithmic toolkits are taken from neural networks [116], fuzzy logic [111], multi-variate statistics [58], filtering [117], and partially from artificial intelligence [118];
- balancing between tendency to centralize data processing and to solve local problems locally [119];
- interpreted as a blend of
  - integrated information systems [120],
  - computerized operator's guides [121],
  - fault detection and isolation modules [122],
  - simple and multi-variate statistical procedures [123],
  - specialized knowledge based system [118],
  - tailored combination of grey-box and black box models,

- multi-variate predictors [116],
- multi-variate advanced controllers [124],
- information filters and transformers allowing intelligent dialogue [125].

15.2 Market competitors

The adopted approach relies on black-box modelling. This orientation on building of universal data-based models is driven by the problem addressed: the modelled processes are so complex that grey-box or white-box modelling would be too expensive. Whenever possible, the grey-box approaches should however be used to complement the advocated approach. They can strengthen it, for instance, by providing prior "physical" information at factor and component levels. The theory presented in [126] and corresponding software MoCaVa serve as an excellent example of grey-box based modelling available.

Black-box modelling and feedback control as well as advising rely on availability of informative data. In this respect, we have to use information systems provided by leading firms specialized on them. Systems of this type care usually not only about sensors quality but also about their testing and checking of data integrity. They employ various signal processing and visualization techniques. Firms CIMT, Cranfield, EMT and Prosig may serve as good examples of this type. Their generality varies but some products integrate the information systems up to the plant level. Products of the firm Yokogawa serve as examples of such solutions.

Advanced multi-variate analysis of data is again supported by many firms and products. For instance, Statistica is a product of a high quality that enables such data analysis. It is not, however, focused on the advisory problem, we are addressing. It could be and should be used by us for inspecting special sub-problems but ready modules and the overall style are definitely prepared for lower-dimensional cases, mostly of a static type. Umetrics products represent multi-variate data analysis oriented more on industrial problems but their aims and tools are less ambitious than ours.

It is not surprising that there are products based on our favorite Bayesian methodology. None of them, however, aims to give advices. Products of the firm MEDC serve as an excellent example in this respect. Huggins is a leading firm in Bayesian networks. Its products serve for probabilistic analysis of cause-consequence relationships and as such they support decision making. They are excellent at analysis level but a systematic design of a decision support is not covered. Moreover, they do not deal with dynamic systems in the control-engineering sense.

Products of the firm Predictive Dynamics are the closest in goals to our approach. Its grey-box oriented product is irrelevant to our problem (see discussion above) but its neural-network based part is highly relevant and represents a real competitor. Here, the strength of the Bayesian approach is believed be to on our side: the whole bunch of theoretical and algorithmic problems is solved within the unified framework (including structure estimation, description of uncertainties, design under uncertainty etc.).

15.3 Common multi-variate techniques used

Here an incomplete list of common techniques is commented.

15.3.1 Principal component analysis

Principal component analysis (PCA) [58] dominates among multi-variate statistical techniques used in the given context. Essentially, eigenvalues and eigenvectors of covariance matrix made of data records are analyzed. It serves both for dimensionality reduction and recognition of non-standard situations (faults). There are many variants that try, for instance,

- to make data processing robust by eliminating outliers that are known to spoil results significantly [127];
- to cope with dynamic character of data [128] by including lagged data into the data records whose covariance matrix is analyzed; moving-window PCA [129] is another remedy offered for the same problem;

- to face the need to cope with multi-scale processes requiring adaptive models [130];
- to combine neural networks (NN) with PCA in order to face non-linear relationships among processed data [131];
- to support a decentralized treatment of local problems [132].

Bayesian versions of PCA are mostly based on Bishop's idea to approximate predictive probability through a minimization of the Kullback-Leibler distance that provides one-sided guaranteed bound on this probability. The idea is also applied in the mixture context [60]. Its numerically stable, fast and dynamic algorithmic solution is still missing.

15.3.2 Progress in classic techniques

The project has origin in clustering area. The NASA supported tool AutoClass [75] is still its prominent example. A care about initialization [133] and structure estimation are the main topic addressed. The achieved results are interesting but their use in high dimensions is still very limited. Non-linear stochastic state-space models and related filtering are applied to specialized supporting systems. It seems that the general "technology" labelled as hidden Markov Chains [134] is the most promising direction in obtaining feasible solution. It is expected to bring widely applicable results after a long development period. Use of local models is another area worth of interest. It searches in extensive data bases of historical records and fits local statistical models to past data similar to the current ones [135].

15.3.3 Neural networks

Neural networks are generally used as universal approximations of non-linear mappings [100]. As such, they serve as non-linear black-box dynamic models in various supporting blocks (standards in fault detection, predictors [116] etc.). They are extensively used so that their advantages and limitations can be studied on real cases [136]. Conclusions are cautiously optimistic, reflecting well known, but rarely reported, instability of results with NN. Specific tasks related to NN are permanently studied. For instance,

- coding of control charts for pattern recognition [121],
- facing reliability and safety issues under a time stress using a blend, NN centered, techniques [137].

15.4 Applications

The presented experience with *applications* we are working with reflects the research stage that corresponds with the last arrow in the verification chain

theory $\rightarrow$ software implementation $\rightarrow$ off line experiments $\rightarrow$ full-scale implementation.

A more matured descriptions of the discussed applications can be found in [138].

15.4.1 Operating of a rolling mill

The rolling mill has already been equipped with a control system enabling a high quality production for properly adjusted manual settings. It is, however, difficult to find optimal settings for all possible working conditions and every type of the material processed.

Problem description

The rolling mill: A fine reversing cold rolling mill with 20-high arrangement of rolls, Fig. 15.1, is considered. About 12 material types – alloys of iron, copper, nickel, zinc, etc. – are processed on it. Contact meters on both sides of the rolling mill provide measurements of strip thickness with $\pm 1\,\mu$m accuracy. The machine is equipped with the two-level control and an information system including the adaptive thickness controller [139]. For properly adjusted settings, the controller is capable to keep deviations of the output strip thickness on the level of the measurement accuracy.

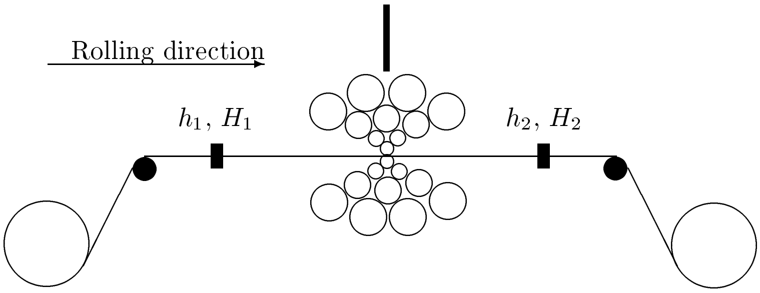


Figure 15.1: Schematic diagram of the 20-high reversing rolling mill. For the selected rolling direction, H_1 and h_1 denote the input strip thickness and deviation from its nominal value, respectively; similarly, H_2 and h_2 for the output thickness.

Process data: The distributed control system of the rolling mill enables to archive every sample of process data. The collection is triggered each 4 centimeters of the strip movement. Usually, 44 signal channels are stored for a further analysis. Thus, for each pass $(\mathring{d}, \mathring{t})$ data matrix is produced where $\mathring{d} = 44$ is the number of channels and the number of samples $\mathring{t}$ varies from 3 000 to 30 000 depending on a particular pass. Ten signal channels have been recognized as the most important for the advising purposes.

Criterion: The deviation of the output strip thickness from its nominal value is the key quality marker.

Problem and its solution

Advisory system is to help to adjust the main process variables in order to maintain the best possible product quality while increasing rolling speed and consequently production. The solution described in this text serves to this purpose. Here, we concentrate on practical implementation aspects.

Pre-processing of data files: Data files corresponding to particular passes were grouped according to the material type. Then three sub-groups were created for the first, even and next odd passes within each group. The corresponding data files within a subgroup were merged into a file having 5×10^5 samples in average. The subsequent mixture estimation provided acceptable results but it was time consuming. Therefore only representative shorter parts from particular data files were merged preserving mutual ratios of data lengths.

As a result, a file containing roughly 30 000 samples of 10 selected signal channels was created for each sub-group. Resulting 36 files were further filtered to suite the mixture estimation. The selection and merging procedures were automated and can be easily repeated for a new set of process data.

Static mixtures: The static mixtures can be considered as a smooth approximation of multi-dimensional histograms representing the number of occurrences of data points in the 10-dimensional (10-D) space. As an input, the data gained during the operation of an experienced operator, yielding a high quality product, are used. The adjustment of the actual working point into a location where data points occurred most often during the good rolling should lead to a high quality product.

Pre-processed data files were used for an iterative estimation in off line mode. Typically, 10 iterations per data file was used. The table shows typical values concerning estimation for a single file with 30 000 records on a 1 GHz machine.

Average number of components	9
Single file processing time for default option	10 min
Single file processing time for “quasi-Bayes” option	160 min

Comparison of the estimated mixture with the empirical pdf constructed from multi-dimensional histograms was used as the main indicator of the estimation success. The estimated mixture seems to be quite fair for some selected 2-D projections, while for another projection with crumbled data clusters it is problematic. Partial improvements can be achieved by utilizing non-default options for identification procedure. At the end, appropriate compromise was achieved considering different importance of signal

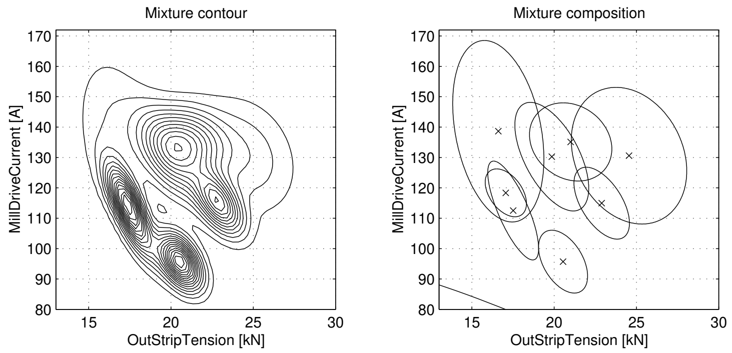


Figure 15.2: Left plot: 2-D marginal projection of a static mixture for two selected data channels. Right plot: composition of the mixture from particular components.

channels from the application point of view. A two-dimensional projection of a typical static mixture is shown on Fig. 15.2. Estimated static mixtures are used as inputs for on line operation according to the type of material and pass number.

Dynamic mixtures: Despite its clear interpretation, the static approach covers only the parts of the data space populated in the past and does not enable predictions for the rest of the data space. Therefore, dynamic modelling was considered. Given the limitations of human perception and reaction times, the term *dynamics* here means transitions among particular stable states of the running machine from the human point of view. Data used as an input for the estimation have to be arranged accordingly. The identification procedure then works with the first order model of the process. Typical processing times are shown in the following table for the same data files.

Average number of components	3
Single file processing time for default option	5 min
Single file processing time for "quasi-Bayes" option	120 min

Location of the resulting mixture and weights of its components become dependent on time and enable to evaluate predictions as a base for proper recommendations. Comparison of predicted and actual values is used for judging of estimation results. In this case, the advisory purpose of the mixture had to be taken into account as well.

As an alternative to a dual static/dynamic estimation, a pseudo-dynamic approach is being investigated, cf. Section 6.4.9. Data matrix is enlarged by 1-step-delayed channels to have double signal channels. Then, the mixture is identified as a static one. The method combines intuitively appealing histogram interpretation while respecting the dynamic nature of dependencies among data records. Time-dependent predictions can be obtained using appropriate conditioning, see Proposition 7.1.2.

Generating of advices: Based on *off line* results of mixture estimations, the advisory part of the system uses actual data for generating recommendations for operators. The shapes and weights of mixture components are re-calculated for every new data sample to show the probability of the best possible location of the working point.

A heuristic signaling is temporarily implemented. A simple indicator is being evaluated continuously which corresponds to the distance between the actual working point and its most probable best location. When it is found that the setting should be changed the system decides which of recognizable inputs is to be adjusted first.

To allow permanent learning, the estimated mixtures are updated concurrently using relevant new data.

Implementation

The system has been implemented into the distributed control system in the form of a dedicated server and a visualization node equipped with a special graphical user interface (GUI). Specific effort was made to enable the bulk of computation to be executed either under MS Windows or Linux operating systems.

The server: The server executes the following two main applications.

Adviser loads a proper mixture file at the beginning of the pass and then, when appropriate, uses every new data sample for evaluating "advisory" mixture and their marginal projections to be displayed for operators;

Updater uses data samples corresponding to a good rolling for updating of the mixture. Thus, the system can refine the estimates and learn new possible locations of "good" working points.

Both applications use multi-threading technique that allows to optimize system performance. Programs are coded in ANSI C and the Tcl language/interpreter is used for interface to a human supervisor, Fig. 15.3. This together with the appropriate version of the Mixtools library [12] make transition to another software platform easy.

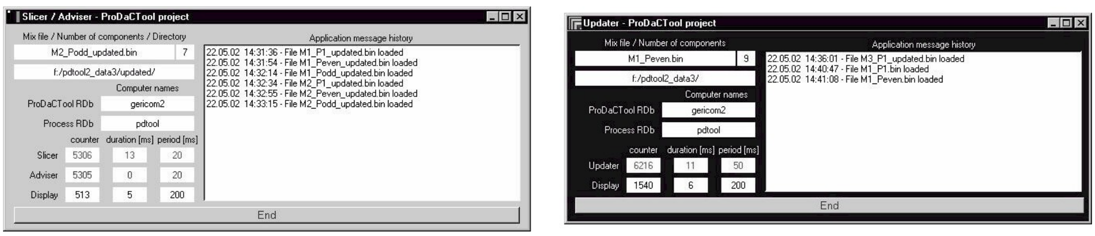


Figure 15.3: Screenshots of Adviser and Updater applications.

Graphical user interface: A new visualization node was introduced for the control system on which the dedicated GUI, Fig. 15.4, for operators is executed. The application can be used in a 1-D and 2-D modes. Its basic properties and appearance can be easily modified through a configuration file. Whatever mode is selected, the GUI displays the overall status of the system in the form of traffic lights where "green" status means "no recommendations". The variable to be changed primarily is emphasized, together with its recommended value. For 2-D mode the operator can select signal channels whose marginal mixture projection should be plot. OpenGL functions are used for smooth on line animations.

Conclusions

The system turns out to stabilize the highest product quality and delimit secure machine settings for the whole variety of working conditions. Obtained results validate prior expectations that the rolling speed can be increased by up to 40 % preserving the highest product quality.

On the other hand, improvements are limited by the amount of information contained in the available data, relatively small and fine mill and low rolling speed. Potential utilization for high-speed steel mills will undoubtedly need more computing power and probably a further development as well.

15.4.2 Treatment of thyroid gland cancer

Another area of application is treatment of thyroid gland tumor using ^{131}I.

The task is to help physicians to decide on administration of individual amount of radioactive ^{131}I for a specific patient using information hidden in retrospective data.

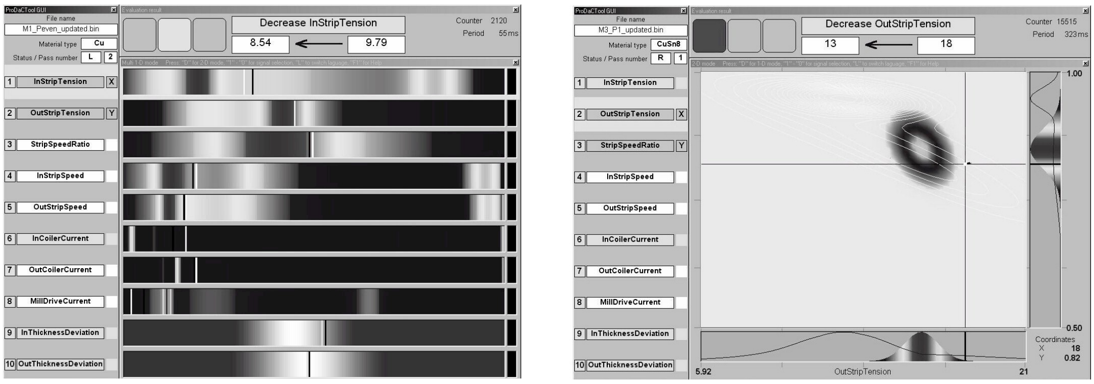


Figure 15.4: Screenshots of the Graphical User Interface for 1-D and 2-D modes respectively.

Problem description

Treatment of thyroid gland cancer is a complex process including endocrinology, histology, surgery, radiology and others. A specific part concerning radiology [140] is described here.

Radio-therapy of thyroid cancer: The thyroid gland accumulates iodine from blood and creates thyroid hormones. Radioactive ^{131}I emits ionizing particles β and γ. Majority of γ-particles (high energy photons) is not absorbed in the tissue and is detectable outside the body. Almost all β-particles (electrons) pass their energy to the tissue by absorption.

Thyroid tumor is usually removed by surgery. What remains or re-appears is destroyed by radiotherapy. The radio-therapy has diagnostic and therapeutic stages.

In the *diagnostic* stage, ^{131}I of a low diagnostic (*tracer*) activity (about 70 MBq) is administered orally to a patient. Then several measurements are performed, detecting γ-radiation from the accumulating tissues, and some quantities describing the kinetic of ^{131}I are evaluated. This information is used in the next stage.

In the *therapeutic* stage, the patient is administered solution with ^{131}I of a high activity (3–10 GBq). The dose (absorbed energy) of β-radiation caused by ^{131}I accumulated in the thyroid destroys it.

About 3–6 months later, the *diagnostic* stage is repeated to examine the therapy result.

After the administration, the thyroid activity rapidly increases, reaches its maximum within hours and then slowly decreases within days or weeks. The decrease can be described by *effective half-life* [140] reflecting combined mechanisms of physical decay and biological elimination. The absorbed dose is proportional to *residence time* [141] for which the administered activity effectively "resided" in thyroid with the same irradiation effect. All *relative activities* are related to the administered activity decreased by physical decay at the given time.

Data: The data used for the mixture analysis were collected from 1992 on more than 6 000 patients. After reorganizing and pre-processing, the final data file contains about 1 100 complete records of 12 biophysical variables directly measured or estimated using Bayesian methodology [142, 143].

The variables describe: (i) general patient's information (gender, age, number of therapies previously undergone), (ii) biophysical information obtained from a diagnostic stage preceding therapy (administered activity of ^{131}I, number of tumor sites accumulating ^{131}I, the effective half-life and residence time, relative activity diluted first and second day after the administration, maximum relative thyroid activity), (iii) activity administered for therapy and (iv) maximum relative thyroid activity in the diagnostic administration *after* therapy.

Criterion: Therapeutic administered activity A_0 decided by the physician is the key quantity to influence the absorbed dose and the treatment success. A_0 must be high enough to destroy the affected tissue but low enough to meet irradiation limits and to minimize secondary radiation risks.

The maximum relative activity A_{rm} (*uptake*) in the diagnostic administration of ^{131}I *after* the therapy is the success indicator. Ideally, after a successful therapy, $A_{rm} = 0$. Practically, $A_{rm} < 0.18\,\%$ of the administered activity represents a successful therapy.

Problem and its solution

The aim is to recommend such a value of activity A_0 for therapy so that 3–6 months later the uptake A_{rm} is less than $0.18\,\%$. A_0 should be chosen as low as possible.

The solution consists of *off line* estimation of a static mixture on the data and *on line* generation of recommendation on A_0.

Pre-processing of data files: Records with missing data were excluded. This caused the significant reduction of available records. For the complete preserved records, the data were tested for physically and medically meaningful ranges.

Mixture estimation: The 12-dimensional data space was examined for occurrence of clusters. Two approaches based on Bayesian methodology were used:

BMTB algorithm: It uses Bayesian extension of the mean-tracking algorithm [144], see Chapter 12. It is computationally very cheap, but, unfortunately, it failed as the data space is sparsely populated.

Mixinit & quasi-Bayes algorithm supported by AutoClass: An older version of initialization [145] and quasi-Bayes algorithm as implemented in Mixtools [12] were used.

The results were improved by using AutoClass [75]. This freely available software [75] detects clusters using the EM algorithm, see Chapter 8, with random starts. Repetitive searches increase probability of finding a better description of clusters but the computational cost increases, too. Over 700 000 random starts have been tried. Low dimensions of the data set have allowed it.

The best result was used as prior information for Mixtools.

Simple advising based on $\mathcal{E}\,[A_{rm}|A_0,\cdot]$: The function $\hat{A}_{rm} = \mathcal{E}[A_{rm}|A_0, \text{patient data}]$ was found for each patient using marginalization of the mixture [146] estimated in learning mode. The value of A_0 (recognizable input) for which the expectation $\hat{A}_{rm} = 0.18\,\%$ is recommended to physicians. The example is shown in the Fig. 15.5.

Industrial design: The *industrial design*, see Chapter 9, was applied as the component weights are understood as objective values reflecting characteristics of the patient population. The normal user ideal pdf ${}^{\lfloor U}f(\cdot)$ was selected to have practical support on the target interval $A_{rm} \in (0, 0.18)$. Similarly, the independent marginal of the administered activity was specified to have support in the usual range of applied activities. The remaining variables were assigned rather flat pdfs as they are out of interest in this design. The ideal conditional pdf was generated for ${}^{\lfloor I}f(A_{rm}, A_0|\text{patient data})$. Its maximum indicates the recommended value of A_0. The example with one-dimensional marginal pdfs of both variables is shown in the Fig. 15.6.

Implementation

The initial versions of advisory modules were programmed in C++ and integrated into the system *Iodine III*, [147]. The system, written in FoxPro, cares about data management and evaluation of a range of Bayesian estimates of various bio-physical quantities. The presented GUI is a part of this system.

Graphical presentations of advices are shown in Figures 15.5 and 15.6. The interactive color probability density map allows physicians to try, for the given patient, various planned administrations and examine the consequences modelled by the advisory modules.

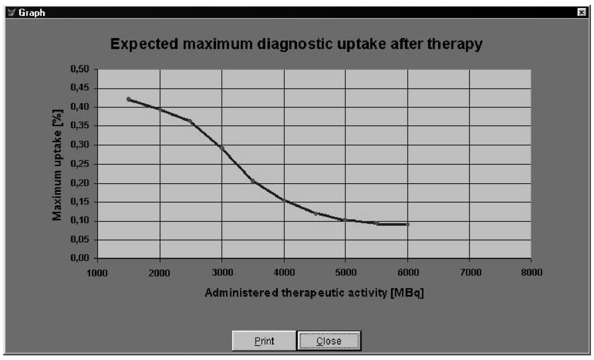


Figure 15.5: Screenshot of *Iodine III* with a deterministic dependence between A_0 and A_{rm}

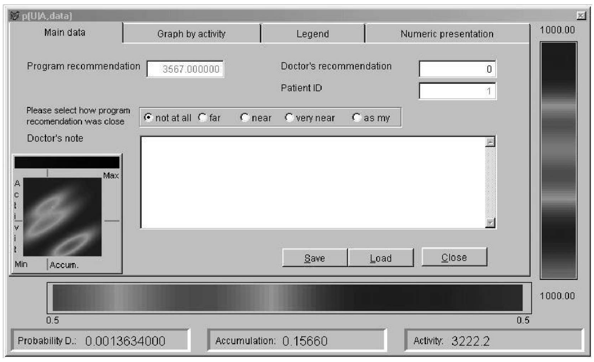


Figure 15.6: Screenshot of *Iodine III* with interactive probability density map

Conclusions

The task is specific by a small amount of the data available, therefore the results of the learning stage depend heavily on prior information. The AutoClass initialization improves performance of the current version of the Mixtools. Even with the prior obtained on the basis of 700 000 random starts, the model designed for the simple version of advising has acceptable physical interpretation only in about half of the cases. The advices are sensitive to mixture estimation. The key question, whether the requested information *is* contained in the available data set, is still not answered definitely. For the final answer, a further research is needed with an active cooperation of medical doctors.

15.4.3 Prediction of traffic

With an increasing number of cars, the transportation problems, especially urban traffic, become urgent [148], [149]. At the same time, their solutions are hard as they have to respect the existing urban conditions.

Traffic control is the application domain where mixture modelling seems to be suitable. Daily and seasonal traffic changes indicate it clearly. This section confirms it by showing possible contributions to an increase of the permeability of city cross-roads by a better utilization of the available traffic lights.

Problem description

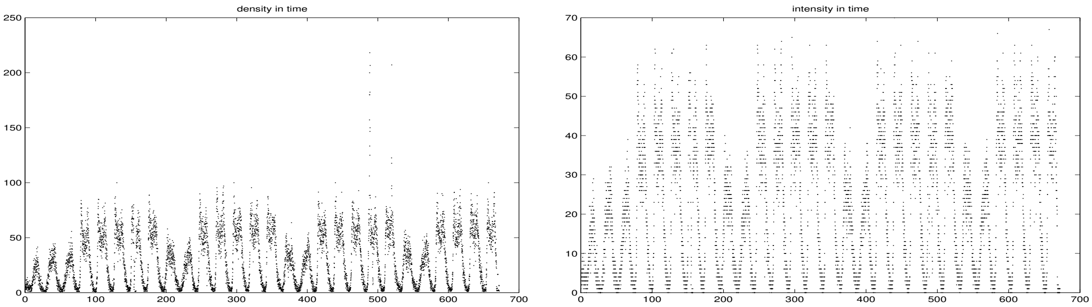
Controlled subjects are cross-roads with adjoining roads. The data are measured by detectors under the road. Each detector gives information about time periods for which it has been occupied by cars and those for which it was free. Using an average length of cars, intensity q (a number of cars per minute) and density ρ (a number of cars per kilometer) of the traffic flow can be computed. These variables fully describe the traffic state at the detector position.

Data description: A time course of the variables on each detector is similar and it reflects daily and weekly periodicity. A typical time course for about 4 weeks is shown in the Fig. 15.7. For experiments, 20-dimensional data records measured along two traffic lanes of the Strahov tunnel in Prague were used. The data sample was split in two parts: 8 500 items (4 weeks) used for learning and 300 items (1 day) for testing.

Criterion: Quality of models used as predictors is judged using by the relative standard deviation R_i of prediction error prediction errors $\hat{e}_{i;t} \equiv d_{i;t} - \hat{d}_{i;t}$ of i-th data entry $d_{i;t}$ (i-th channel)

$$R_i \equiv \sqrt{\frac{\sum_{t \in t^*} \hat{e}^2_{i;t}}{\sum_{t \in t^*} (d_{i;t} - \bar{d}_i)^2}}, \quad i = 1, \ldots, \mathring{d} = 20, \ \bar{d}_i \equiv \text{sample mean of } d_{i;t} \ t \in t^*. \tag{15.1}$$

In (15.1), the point prediction $\hat{d}_{i;t}$ is an approximate conditional expectation of $d_{i;t}$.



Time course of density ρ | Time course of intensity q

Figure 15.7: Traffic variables from the middle of the tunnel

Problem and its solution

Cross-roads in cities are controlled by setting a proper green/red proportion of the lights. Mostly, these signals are set according to a working plan, which changes during the day according to a fixed schedule. Automatic feedback in selection of these plans can improve quality of the traffic significantly.

For the control, knowledge of actual value of incoming traffic flow in the near future is necessary. Reliable and possibly multi-step prediction of the traffic flow can decide about practical success of such a feedback control.

Pre-processing of data: Outlier filtration and normalization to zero mean and standard deviation equal to one were applied on the raw data.

Model for prediction: The traffic system switches among several different states. A mixture model has been chosen as a proper tool for modelling.

The model is built in the learning *off line* phase and used for prediction in *on line* phase. In the process of learning, the joint pdf of the modelled data is estimated. Then, the pdf predicting selected channel(s) conditioned on other measured data is created. It is used for point or interval predictions of the respective channels.

Estimation of mixture models: An automatic initialization determining the number of components of the mixture model was used [145]. Mostly from two to six components were automatically determined and used. In all cases, at least three of these components had non-negligible weights. Some components had almost zero weights but it is interesting that their omission decreased the prediction quality visibly.

Experiments

The experiments verified the ability of the estimated mixture model to predict characteristics of the traffic flow at the northern exit of the tunnel using the data along the whole tunnel. A comparison of prediction quality with mixture (MIX) model and auto-regression (AR) model is presented.

Two basic types of model components were used. Static ones, which are simpler and commonly used in mixture modelling and dynamic, which are of our main interest. The results, comparing quality of prediction expressed by R_i (15.1) in dependence on the number of modelled channels $\mathring{d}$ and the number of prediction steps, are presented in the tables.

Static mixtures: Static components depending only on a constant term were used here. This experiment shows need for dynamic models and superiority of the mixture model comparing to the AR model. Intensity and density at the tunnel exit (1^{st} and 2^{nd} channel) were modelled only.

Table 15.1 shows prediction quality expressed by the criterion R_i (15.1) for two-dimensional modelled channels and the prediction horizon ranging from 1 to 18 steps.

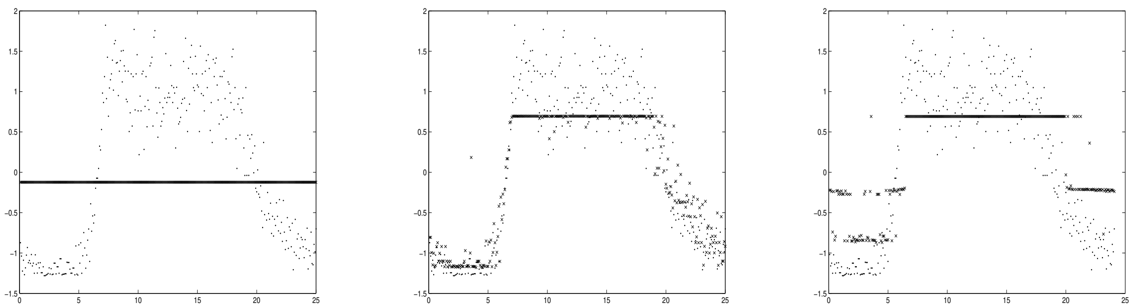
Steps ahead	R_1: MIX	R_1: AR	R_2: MIX	R_2: AR
1	0.418	1.021	0.422	1.020
6	0.512	1.025	0.516	1.024
12	0.634	1.031	0.638	1.030
18	0.756	1.028	0.760	1.037

Table 15.1: Multi-step ahead predictions with static two-dimensional model. R_i is the criterion (15.1) for channels $i = 1, 2$. Results related to the mixture and auto-regression models are marked "MIX" and "AR", respectively.

	One step ahead, 5 [min], prediction				Six steps ahead, 30 [min], prediction			
$\mathring{d}$	R_1: MIX	R_1: AR	R_2: MIX	R_2: AR	R_1: MIX	R_1: AR	R_2: MIX	R_2: AR
2	0.391	0.376	0.392	0.378	0.487	0.526	0.483	0.521
20	0.328	0.330	0.329	0.331	0.421	0.444	0.416	0.436
	Twelve step ahead, 60 [min], prediction				Eighteen steps ahead, 90 [min], prediction			
2	0.614	0.711	0.615	0.706	0.758	0.840	0.766	0.836
20	0.524	0.576	0.526	0.566	0.654	0.697	0.644	0.688

Table 15.2: Multi-step ahead predictions with dynamic models. $\mathring{d}$ – the number of modelled channels, R_i the prediction criterion (15.1) for channels $i = 1, 2$. Results related to mixture and auto-regression model are marked "MIX" and "AR", respectively.

Figure 15.8 illustrates typical behavior of the compared auto-regression "AR" and mixture "MIX" models on one step and eighteen steps ahead predictions.



Static "AR" prediction ≡ average   1 step ahead "MIX" prediction   18 step ahead "MIX" prediction

Figure 15.8: Prediction with static auto-regression "AR" and mixture "MIX" models. Dots denote measured data, $\times$ predictions.

Dynamic mixtures: First order normal components have been used.

Table 15.2 shows a comparison of the criterion R_i, (15.1), for the "AR" and "MIX" models used as multiple-step ahead predictors, covering the range from 5 to 90 minutes. Two and twenty channels are modelled but only first two channels are shown.

Results

Comparison of regression and mixture models: It can be seen, that the results with mixture model are almost always better, even though the signal lights are preset into the mode preventing traffic jam inside the Strahov tunnel and limit other traffic modes. The mixture model can manage long multi-variate data samples, it can also utilize better the information from the additional entries of data records.

Static models: The good property of mixtures is clearly seen on Fig. 15.8. The static AR model is able to predict only the mean value of the whole data sample. The mixture, due to the data dependent weights of individual components, is able to follow even the ascending and descending trends of the data, see Section 7.1.2. With the increasing length of prediction the ability to follow data declines but the switching among components is clearly visible.

Dynamic models: They describe and predict data of this kind much better. For short-term predictions, the difference between the AR and mixture models is rather small. It is about a level 1 to 5 % in terms of R_i. On multi-step predictions, the contribution of mixture modelling is much more pronounced. Also, advantages of the use of high dimensional models for construction of low-dimensional is demonstrated.

Conclusions

In summary, mixtures are worth to be considered for modelling and prediction of high-dimensional transportation data. Their ability to make reasonable multi-step ahead predictions indicates their ability to grasp a significant portion of physical properties of the modelled systems. This is a necessary property of a model used for the control design.

15.5 Conclusions

Inspection of the state of the art leads to the following conclusions.

- The advisory system may serve well as a "clever" node in a more complex hierarchical support operator or their group. It cannot cover all tasks taken traditionally as operator support and should rely on information system available as well as on the signals generated by other specialized modules.
- The advisory system has found a specific niche on the market by providing relatively quickly an advanced tailored addition to existing information systems. It will benefit from the expected speed of its implementation and its adaptive abilities, Chapter 9.
- The advisory system has to be extended so that the results of grey-box modelling can be exploited using available models both as prior-pdf-shaping knowledge and as an alternative model in stabilized forgetting, Chapters 6, 8.
- The theoretical and algorithmic outcomes of the research may well complement tool set for building specialized modules like fault-detectors.
- Abilities of the advisory system to reduce information overflow and focus operator attention on significant variables are more important than expected. Thus, efficient and fast solution of the presentation task is quite important, see Chapter 7, 9.
- The advisory system is coming in a proper time as users are gradually willing to work with sophisticated multi-variate data processing.
- The advisory system has no viable competitor that would be able to give optimized advices using dynamic, black-box, high-dimensional models. Its generic nature makes the resulting product adaptable to a surprisingly wide range of problems.

The described applications confirm the last item above as well as usefulness of the developed theory and algorithms. At the same time, the applications proved to be significant for developing both the theory and algorithms. They helped us to discover errors, inconsistencies, drawbacks and bugs and stimulated solutions of many sub-problems that arisen in attempts to implement the theoretical results. The process is still unfinished but the discussed interaction can be unanimously claimed to be useful.

Lessons learnt

The applicability of mixture models has been tested on three real processes: rolling of metal strips, treatment of thyroid gland tumor and prediction of transportation data. Common feature of all these processes is their multi-modal nature which prevents to rely on ordinary linear regression models for their good description.

The mixture models, in spite of necessary approximation in their estimation, proved to be a good tool for description of multi-modal systems. They are able to model several different system modes simultaneously. High dimensions of the modelled variable bring long processing times but they guarantee a good estimation quality.

A specific trait of nuclear medicine application is insufficiency of data given by the technical conditions of the treatment and data measurement. It makes the most visible the key application problem: the amount of information on the system carried by the data has to be sufficient. According to the *GIGO principle* (garbage in, garbage out), the quality of advices depends on how much information in the data represents the system behavior in its entirety. It occurs that lack of information in the data is crucial for successful estimation and advising.

Other application areas: The described applications are just samples from an extreme application domain of the described theory and algorithms. During the development of the advisory system we met, for instance, the following tasks that can efficiently use the research outcomes

- prediction of non-linear phenomena like load in energy networks (electricity, gas, heat),
- modelling, prediction and evaluation of life-saving signals of prematurely born children,
- use of mixtures for speculative short-term monetary operations,
- support of operators of large furnaces for producing glass or metal,
- call for creating operation model to be used for training of operators of complex processes,
- attempt to use mixture models for fault detection and isolation [150],
- effort to use mixture models in complex social phenomena like the project of European Science Foundation "Towards Electronic Democracy",
- • •

Chapter 16

Conclusions

This work has arisen during solution of project IST-1999-12058 as a theoretical and algorithmic basis of the project solution. The project itself has been quite ambitious both with respect to its content, planned short period between elaboration of the theoretical solutions and the full scale use. The result is necessarily marked by the fact that we needed to collect or design theoretical solutions that can be converted in implementable algorithms. As such, it is unbalanced in its level, it is counter-pedagogical, contains many repetitions and surely some inconsistencies, various solutions have an ad hoc character etc.

These problems may hide achievements we are proud of. First of all, our formulation of the design of the advisory systems is novel and represents an extension of the classical decision theory for dynamic uncertain system. Consequently, it has the generic nature and an extreme applicability width. The following list underlines some points within this framework we feel as crucial achievements.

1. Academic, industrial and simultaneous advising were formulated and solved using approximate probabilistic optimization of advising.

2. Dynamic mixture modelling and prediction provide a novel and powerful decision-supporting tool.

3. Problem of the mixture estimation with mixed dynamic components has not been solved before to the extent presented here.

4. The designed estimation algorithms admit adaptive on-line clustering with an extreme potential for a permanent improvements of managing.

5. The basic initialization of the mixture estimation seems to be a novel tool applicable in high dimensional data spaces.

   It also solves – to a substantial extent – the structure estimation problem that is known to be crucial in the mixture estimation.

6. All estimation and design algorithms for the basic normal factors use of efficient, numerically robust, factorized algorithms.

7. Claims above are supported by the diversity of the successful applications we have managed implement within the project span, see Chapter 15.

We are aware that a lot of things is unfinished and some of them missing. The crucial steps we should do are as follows.

1. Off line prediction of closed loop performance of the advisory system should be done, in the vein of DESIGNER [70].

2. Filtering counterpart of the learning should be made. This would allow to exploit process knowledge more deeply and to express management aims more directly.

3. The set of employed dynamic factors can be and should be significantly extended.
4. The models of the "rational" form – ratios of a mixture pair – should be addressed in order to cope with the dynamic case more rigorously.
5. The number of ad hoc solutions should be decreased.
6. Improvements of various aspects of the proposed solutions should be considered.
7. Experience with real applications should be accumulated further on and reflected in the tool set we described here ...

Bibliography

[1] G. F. Luger and W. A. Stubblefield, *Artificial Intelligence and the Design of Expert Systems*, chapter 8, pp. 296–297, Benjamin/Cummings Publisher Company Inc., 1989.

[2] B. S. Everitt and D. J. Hand, *Applied Multivariate Data Analysis*, Edward Arnold, London, 1991.

[3] J. Girst, "The practical application of rule and knowledge-based system techniques to industrial processes", in *Rule-Based Systems for Real-Time Planning and Control, IEE Colloquium*, October 1991, number 1991/160.

[4] S. R. Schmidt and R. G. Launsby, *Understanding Industrial Designed Experiments*, Air Academy Press, Colorado, 1992.

[5] B. S. Everitt and D. J. Hand, *Finite Mixture Distributions*, Chapman and Hall, 1981.

[6] D. W. Scott, *Multivariate Density Estimation*, Wiley, New York, 1992.

[7] B. W. Silverman, *Density Estimation*, Chapman and Hall, 1991.

[8] J. Coste, A. Spira, P. Ducimetiere, and J. B. Paolaggi, "Clinical and psychological diversity of non-specific low-back pain. a new approach towards the classification of clinical subgroups", *Journal of Clinical Epidemiol*, vol. 44, pp. 1233–1245, 1991.

[9] R. Huth, I. Nemesova, and N. Klimperova, "Weather categorization based on the average linkage clustering technique: An application to European Midlatitudes", *International Journal of Climatology*, vol. 13, 1993.

[10] N. Jardine and R. Sibson, "The construction of hierarchic and non-hierarchic classification", *Computer Journal*, vol. 11, pp. 177–184, 1968.

[11] E.L. Sutanto, *Mean-tracking algorithm for multivariable cluster analysis in manufacturing processes*, PhD thesis, Reading, UK, 1996.

[12] P. Nedoma, M. Kárný, I. Nagy, and M. Valečková, "Mixtools. MATLAB Toolbox for Mixtures", Tech. Rep. 1995, ÚTIA AV ČR, Praha, 2000.

[13] M. Kárný, A. Halousková, J. Böhm, R. Kulhavý, and P. Nedoma, "Design of linear quadratic adaptive control: Theory and algorithms for practice", *Kybernetika*, vol. 21, 1985, Supplement to No. 3, 4 ,5, 6.

[14] V. Peterka, "Bayesian system identification", in *Trends and Progress in System Identification*, P. Eykhoff, Ed., pp. 239–304. Pergamon Press, Oxford, 1981.

[15] M. Kárný, I. Nagy, and J. Novovičová, "Quasi-Bayes Approach to Multi-Model Fault Detection and Isolation", Tech. Rep. 1974, ÚTIA AV ČR, Praha, 1999.

[16] R.L. Keeny and H. Raiffa, *Decisions with multiple objectives: Preferences and value tradeoffs*, J. Wiley and Sons, New York, 1978.

[17] M.M. Rao, *Measure Theory and Integration*, John Wiley and Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1987, Pure and Applied Mathematics, Wiley-Interscience Series of Texts, Monographs and Tracts.

[18] V. Jarník, *Integral calculus II*, Academia, Prague, 1984, in Czech.

[19] I. Vajda, *Information theory and stadistical decision making*, Alfa, Bratislava, 1982, in Slovak.

[20] A.A. Feldbaum, "Theory of dual control", *Autom. Remote Control*, vol. 21, no. 9, 1960.

[21] A.A. Feldbaum, "Theory of dual control", *Autom. Remote Control*, vol. 22, no. 2, 1961.

[22] R. Bellman, *Introduction to the Mathematical Theory of Control Processes*, Academic Press, New York, 1967.

[23] S. Kullback and R. Leibler, "On information and sufficiency", *Annals of Mathematical Statistics*, vol. 22, pp. 79–87, 1951.

[24] M. Kárný, "Towards fully probabilistic control design", *Automatica*, vol. 32, no. 12, pp. 1719–1722, 1996.

[25] J. Šindelář and M. Kárný, "Dynamic decision making under uncertainty allows explicit solution", Tech. Rep. 1916, ÚTIA AVČR, POB 18, 18208 Prague 8, CR, 1997.

[26] H. Kushner, *Introduction to stochastic control*, Holt, Rinehart and Winston, New York, San Francisco, London, 1971.

[27] M. Loeve, *Probability theory*, D. van Nostrand Company Inc., Princeton, New Jersey, 1962, Russian translation, Moscow 1962.

[28] L. Berec and M. Kárný, "Identification of reality in Bayesian context", in *Computer-Intensive Methods in Control and Signal Processing: Curse of Dimensionality*, K. Warwick and M. Kárný, Eds. Birkhauser, 1997.

[29] I.N. Sanov, "On probability of large deviations of random variables", *Matematičeskij Sbornik*, vol. 42, pp. 11–44, 1957, in Russian, translation in Selected Translations mathematical statistics and probability, I, 1961, 213–244.

[30] R. Kulhavý, *Recursive Nonlinear Estimation: A Geometric Approach*, vol. 216 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, London, 1996.

[31] R. Kulhavý and M. B. Zarrop, "On general concept of forgetting", *International Journal of Control*, vol. 58, no. 4, pp. 905–924, 1993.

[32] R.K. Mehra and D.G. Lainiotis (Eds.), *System identification – Advances and case studies*, Pergamon Press, New York, 1976.

[33] R. Koopman, "On distributions admitting a sufficient statistic", *Transactions of American Mathematical Society*, vol. 39, pp. 399, 1936.

[34] M. Kárný, "Probabilistic identification and self-reproducing pdfs with application to order estimation", Tech. Rep., ÚTIA ČSAV, Prague, 1976, in Czech, CSc thesis.

[35] E.F. Daum, "New exact nonlinear filters", in *Bayesian Analysis of Time Series and Dynamic Models*, J.C. Spall, Ed. Marcel Dekker, New York, 1988.

[36] L. Berec, *Model Structure Identification: Global and Local Views. Bayesian Solution*, Ph.D. Thesis, Czech Technical University, Faculty of Nuclear Sciences and Physical Engineering, Prague, Czech Republic, 1998.

[37] R. Kulhavý, "A Bayes-closed approximation of recursive non-linear estimation", *International Journal Adaptive Control and Signal Processing*, vol. 4, pp. 271–285, 1990.

[38] R. Kulhavý, "Recursive Bayesian estimation under memory limitations", *Kybernetika*, vol. 26, pp. 1–20, 1990.

[39] R. Taylor, *Introduction to functional analysis*, Academia, Prague, 1973, Czech translation.

[40] V.S. Vladimirov, *Generalized functions in mathematical physics*, Mir Publishers, Moscow, 1979.

[41] M. Kárný, "Adaptive systems: Local approximators?", in *Preprints of the IFAC Workshop on Adaptive Systems in Control and Signal Processing*, Glasgow, August 1998, pp. 129–134, IFAC.

[42] V. Peterka, "Adaptive digital regulation of noisy systems", in *Preprints of the 2nd IFAC Symposium on Identification and Process Parameter Estimation*, p. paper 6.2. Prague, 1970.

[43] V. Peterka, "Predictor-based self-tuning control", *Automatica*, vol. 20, no. 1, pp. 39–50, 1984, reprinted in : Adaptive Methods for Control System Design, Editor M.M. Gupta, IEEE Press, 1986.

[44] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized predictive control", *Automatica*, vol. 23, no. 2, pp. 137–160, 1987.

[45] O.L.R. Jacobs and J.W. Patchell, "Caution and probing in stochastic control", *International Journal of Control*, vol. 16, pp. 189–199, 1972.

[46] A. Wald, *Statistical Decision Functions*, John Wiley & Sons, New York, London, 1950.

[47] J. Rojíček and M. Kárný, "A sequential stopping rule for extensive simulations", in *Preprints of the 3rd European IEEE Workshop on Computer-Intensive Methods in Control and Data Processing*, J. Rojíček, M. Valečková, M. Kárný, and K. Warwick, Eds., Praha, September 1998, pp. 145–150, ÚTIA AV ČR.

[48] M. Kárný, E. Sutanto, P. Ettler, J. Kadlec, A. Quinn, and L. Tesař, "Decision support tool for complex industrial processes based on probabilistic data clustering", Tech. Rep., ÚTIA AVČR, 1998, deliverable of Esprit project ProDaCTools, No. 25729.

[49] D.M. Titterington, A.F.M. Smith, and U.E. Makov, *Statistical Analysis of Finite Mixtures*, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1985, ISBN 0 471 90763 4.

[50] R. Horst and H. Tuy, *Global Optimization*, Springer, Berlin, 1996, 727 pp.

[51] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, 1996, 387 pp.

[52] Tao Chen, Kai-Kuang Ma, and Li-Hui Chen, "Tri-state median filter for image denoising", *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1834–1838, December 1999.

[53] T. Cipra, "Dynamic credibility with outliers", *Applications of Mathematics*, vol. 41, no. 2, pp. 149–159, 1996.

[54] M. Tanaka and T. Katayama, "A robust identification of a linear system with missing observations and outliers by the EM algorithm", *Transactions of the Institute of Systems, Control and Information Engineering*, vol. 1, no. 4, pp. 117–129, September 1988.

[55] L. Tesař and A. Quinn, "Detection and removal of outliers from multidimensional AR processes", in *Proceedings of Irish Signal and Systems Conference*, Maynooth, Ireland, August 2001.

[56] L. Tesař and A. Quinn, "Method for artefact detection and supressing using alpha-stable distributions", in *Proceedings of ICANNGA Conference*, Prague, Czech Republic, March 2001.

[57] A.V. Oppenheim and A.S. Wilsky, *Signals and systems*, Prentice Hall, New Jersye, 1983.

[58] I.T. Jolliffe, *Principal Component Analysis*, Springer Verlag, 1986.

[59] M.E. Tipping and C.M. Bishop, "Probabilistic principal component analysis", *Journal of the Royal Society Series B – Statisticl Methodology*, vol. 61, pp. 611–622, 1999.

[60] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers", *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[61] S.J. Godsill, *The Restoration of Degraded Audio Signals*, PhD thesis, University of Cambridge, Department of Engineering, December 1993.

[62] Kárný M., "Local filter robust with respect to outlying data", Tech. Rep. 1644, UTIA CSAV, Praha, 1990.

[63] Raghuver M. Rao and Ajit S. Bopardikar, *Wavelet transforms: Introduction to theory and applications*, Addison Wesley Longman, July 1998.

[64] He Li and M. Kárný, "Estimation and prediction with ARMMAX model: a mixture of ARMAX models with common ARX part", *International Journal of Adaptive Control and Signal Processing*, 2003, in print.

[65] V. Peterka, "Real-time parameter estimation and output prediction for ARMA-type system models", *Kybernetika*, vol. 17, pp. 526–533, 1981.

[66] V. Peterka, "Control of uncertain processes: applied theory and algorithms", *Kybernetika*, vol. 22, 1986, Supplement to No. 3, 4 ,5, 6.

[67] V. Šmídl, T.V. Guy, M. Kárný, and A. Quinn, "Estimation of arx model with parallelly filtered regression vectors", *System Control Letters*, 2003, submitted.

[68] M. Kárný, "Quantification of prior knowledge about global characteristics of linear normal model", *Kybernetika*, vol. 20, no. 5, pp. 376–385, 1984.

[69] J. Bůcha, M. Kárný, P. Nedoma, J. Böhm, and J. Rojíček, "Designer 2000 project", in *International Conference on Control '98*, London, September 1998, pp. 1450–1455, IEE.

[70] N. Khaylova, "Exploitation of prior knowledge in adaptive control design", Tech. Rep., FAV ZČU, University of West Bohemia, Pilsen, Czech Republic, 2001, PhD Thesis.

[71] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, London, 1997, 403 pp.

[72] B. Harris and G. Heinel, "The relation between statistical decision theory and approximation theory", in *Optimizing Methods in Statistics*, J.G. Rustagi, Ed., pp. 263–272. Academic Press, New York, San Francisco, London, 1979.

[73] M. Kárný, J. Kadlec, and E. L. Sutanto, "Quasi-Bayes estimation applied to normal mixture", in *Preprints of the 3rd European IEEE Workshop on Computer-Intensive Methods in Control and Data Processing*, J. Rojíček, M. Valečková, M. Kárný, and K. Warwick, Eds., Praha, September 1998, pp. 77–82, ÚTIA AV ČR.

[74] S. Richardson and P.J. Green, "On Bayesian analysis of mixtures with an unknown number of components, with discussion", *Journal of the Royal Statistical Society, Series B*, vol. 59, no. 4, pp. 731–792, 1997.

[75] J. Stutz and P. Cheeseman, "AutoClass - a Bayesian approach to classification", in *Maximum Entropy and Bayesian Methods*, J. Skilling and S. Sibisi, Eds. Kluwer Academic Publishers, Dordrecht, 1995.

[76] M. Kárný and R. Kulhavý, "Structure determination of regression-type models for adaptive prediction and control", in *Bayesian Analysis of Time Series and Dynamic Models*, J.C. Spall, Ed. Marcel Dekker, New York, 1988, chapter 12.

[77] M. Valečková, M. Kárný, and E. L. Sutanto, "Bayesian M-T clustering for reduced parametrisation of Markov chains used for non-linear adaptive elements", in *Preprints of the IFAC Workshop on Adaptive Systems in Control and Signal Processing*, Glasgow, August 1998, pp. 381–386, IFAC.

[78] V. Peterka, "Adaptation for LQG control design to engineering needs", in *Lecture Notes: Advanced Methods in Adaptive Control for Industrial Application; Joint UK-CS seminar*, K. Warwick, M. Kárný, and A. Halousková, Eds., vol. 158. Springer Verlag, 1991, held in May 1990, Prague.

[79] G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1977.

[80] M. Kárný, "Identification for adaptive control", Tech. Rep., ÚTIA AV ČR, POBox 18, 182 08 Prague 8, Czech Republic, 1990, DrSc thesis, in Czech.

[81] M. Abramowitz and I.A. Stegun, *Handbook of mathematical functions*, Dover Publications, Inc., New York, 1972.

[82] M. Kárný, "Parametrization of multi-output multi-input autoregressive-regressive models for self-tuning control", *Kybernetika*, vol. 28, no. 5, pp. 402–412, 1992.

[83] M. Kárný, "Local filter insensitive to outliers", in *9th National conference ASRTP'90*, Nízké Tatry, Tále, 1990, pp. 141–148, in Czech.

[84] M. Kárný, "Algorithms for determining the model structure of a controlled system", *Kybernetika*, vol. 19, no. 2, pp. 164–178, 1983.

[85] M. Kárný, J. Bohm, T.V. Guy, and P. Nedoma, "Mixture-based adaptive probabilistic control", *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 2, pp. 119–132, 2003.

[86] I. Nagy and M. Kárný, "A view on filtering of continuous data signals", *Kybernetika*, vol. 28, no. 6, pp. 494–505, 1992.

[87] T. V. Guy, M. Kárný, and J. Böhm, "Linear adaptive controller based on smoothing noisy data algorithm", in *European Control Conference. ECC '99. (CD-ROM)*, Karlsruhe, August 1999, VDI/VDE GMA.

[88] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, London, 1987.

[89] A. Renyi, *Probability theory*, Academia, Prague, 1972, in Czech.

[90] R. Murray-Smith and T.A. Johansen, *Multiple Model Approaches to Modelling and Control*, Taylor & Francis, London, 1997.

[91] M. Valečková and M. Kárný, "Estimation of Markov chains with reduced parametrisation", in *Preprints of the 2nd European IEEE Workshop on Computer Intensive Methods in Control and Signal Processing, CMP'96*, L. Berec, J. Rojíček, M. Kárný, and K. Warwick, Eds., pp. 135–140. ÚTIA AV ČR, Prague, 1996.

[92] T.S. Fergusson, "A Bayesian analysis of some nonparametric problems", *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.

[93] R. Jiroušek, "Introduction into Bayesian network theory", Tech. Rep. LISp-94-04, LISp, Department of Information and Knowledge Engineering, Prague University of Economics, 1994, in Czech.

[94] M. Valečková, M. Kárný, and E. L. Sutanto, "Bayesian M-T clustering for reduced parameterisation of Markov chains used for non-linear adaptive elements", *Automatica*, vol. 37, no. 6, pp. 1071–1078, 2001.

[95] H. Mine and S. Osaki, *Markovian Decision Processes*, Elsevier, New York, 1970.

[96] M. Hendrikx, J. van Nunen, and J. Wessels, "On iterative optimization of structured markov decision processes with discounted rewards", *Math. Operationsforsch. und Statistics, ser. Optimization*, vol. 15, no. 3, pp. 439–459, 1984.

[97] J. Šindelář, M. Kárný, and P. Nedoma, "Algorithms selecting several best "diagonal" variants", Tech. Rep., ÚTIA AV ČR, 1998.

[98] Šindelář J., Kárný M., and Nedoma P., "Algorithm for selecting the best variant in the shadow cancelling problem", *Computational Statistics and Data Analysis*, submitted.

[99] Rabe-Hesketh S. and Skrondal A., "Parameterization of multivariate effects models for categorical data", *Biometrics*, vol. 57, no. 4, pp. 1256–1263, 2001.

[100] S. Haykin, *Neural networks: A comprehensive foundation*, Macmillan College Publishing Company, New York, 1994.

[101] P. Nedoma, J. Böhm, T. V. Guy, L. Jirsa, M. Kárný, I. Nagy, L. Tesař, and J. Andrýsek, "Mixtools: User's Guide", Tech. Rep. 2060, ÚTIA AV ČR, Praha, 2002.

[102] P. Nedoma, M. Kárný, and J. Böhm, "ABET99. Adaptive Bayesian Estimation Toolbox for MATLAB", Tech. Rep. 1963, ÚTIA AV ČR, Praha, 1999.

[103] M. Tichý and B. Kovář, "Parallel factorised algorithms for mixture estimation", in *Artificial neural networks and genetic algorithms*, V. Kůrková, N.C. Steel, R. Neruda, and M. Kárný, Eds., pp. 410–413. Springer-Verlag, Austria, 2001.

[104] P. Ettler, M. Kárný, and I. Nagy, "Employing information hidden in industrial process data", in *Proceedings of the Symposium Intelligent Systems for Industry SOCO/ISFI*, Paisley, June 2001, pp. 1814–1817, Academic Press.

[105] P. Ettler, M. Valečková, M. Kárný, and I. Puchr, "Towards a knowledge-based control of a complex industrial process", in *Proceedings of the American Control Conference*, Chicago, June 2000, pp. 2063–2066, IEEE.

[106] The Matlab Inc., *Matlab documentation*, vol. MA 01760-2098, Natick, 2000.

[107] A. Geists and et al, *PVM: Parallel Virtual Machine. User's guide and tutorial for networked parallel computing*, The MIT Press, Cambridge, Massachusetts, 1994.

[108] P. Tvrdík, *Parallel systems and algorithms*, Publishing House of the Czech Technical University, Prague, 1996, Lecture notes in Czech.

[109] T.H.Lee, C.C.Hang, and K.K.Tan, "Special issue in intelligent control for industrial applications", *Journal of Adaptive Control and Signal Processing*, vol. 15, no. 8, 2001.

[110] J.T.Kim, K.C. Kwon, I.K. Hwang, D.Y. Lee, W.M. Park, J.S. Kim, and S. Lee, "Development of advanced I & C in nuclear power plants: ADIOS and ASICS", *Nuc. Eng. Des.*, vol. 207, no. 1, pp. 105–119, 2001.

[111] M. Setnes and R. Babuška, "Fuzzy decision support for the control of deteregent production", *Int. J. of Adaptive Control and Signal Processing*, vol. 15, no. 8, pp. 769–785, 2001.

[112] C. Lindheim and K.M. Lien, "Operator support systems for new kinds of process operation work", *Comput. Chem. Eng.*, vol. 21, pp. S113–S118, 1997.

[113] I. Ivanov, "Multivariate techniques for process monitoring and optimization at Tembec", *Pulp Pap. – Can.*, vol. 102, no. 7, pp. 23–25, 2001.

[114] B. Lennox, G.A. Montague, H.G. Hiden, G. Kornfeld, and P.R. Goulding, "Process monitoring of an industrial fed-batch fermentation", *Biotechnol. Bioeng.*, vol. 74, no. 2, pp. 125–135, 2001.

[115] A. Mjaavatten and B.A. Foss, "A modular system for estimation and diagnosis", *Comput. Chem. Eng.*, vol. 21, no. 11, pp. 1203–1218, 1997.

[116] Y.D. Pan, S.W. Sung, and J.H. Lee, "Data-based construction of feedback-corrected nonlinear prediction model using feedback neural networks", *Control Engineering Practice*, vol. 9, no. 8, pp. 859–867, 2001.

[117] A. Alessandri, T. Parisini, and R. Zoppoli, "Sliding-window neural state estimation in a power plant heater line", *Int. Journal of Adaptice Control and Signal Processing*, vol. 15, no. 8, pp. 815–836, 2001.

[118] M. Malek, M.P. Toitgans J.L. Wybo, and M. Vincent, "An operator support system based on case-based reasoning for the plastic moulding injection process", in *Lecture note in artificial intelligence*, vol. 1488, pp. 402–413. 2001.

[119] A. Bonastre, R. Ors, and M. Peris, "Distributed expert systems as a new tool in analytical chemistry", *Trac–Trends Anal. Chem.*, vol. 20, no. 5, pp. 263–271, 2001.

[120] G.A. Sundstrom, "Designing support contexts: Helping operators to generate and use knowledge", *Control Eng. Practice*, vol. 5, no. 3, pp. 375–381, 1997.

[121] M.B. Perry, J.K. Spoerre, and T. Velasco, "Control chart pattern recognition using back propagation artificial neural networks", *Int. J. Prod. Res.*, vol. 39, no. 15, pp. 3399–3418, 2001.

[122] C.W. Lu and M.R. Reynolds, "Cusum charts for monitoring an autocorrelated process", *J. Qual. Technol.*, vol. 33, no. 3, pp. 316–334, 2001.

[123] F. Doymaz, J. Chen, J.A. Romagnoli, and A. Palazoglu, "A robust strategy for real-time process monitoring", *J. Process Control*, vol. 11, no. 4, pp. 343–359, 2001.

[124] A. Fink, O. Nelles, M. Fisher, and R. Iserman, "Non-linear adaptive control of a heater exchanger", *Int. Journal of Adaptice Control and Signal Processing*, vol. 15, no. 8, pp. 883–906, 2001.

[125] B. Kuijpers and K. Dockx, "An intelligent man-machine dialogue system based on ai planning", *J. Appl. Intell.*, vol. 8, no. 3, pp. 235–245, 1998.

[126] T. Bohlin, *Interactive System Identification: Prospects and Pitfalls*, Springer-Verlag, Berlin, Heidelberg, New York, 1991.

[127] K.A. Hoo, K.J. Tvarlapati, M.J. Piovoso, and R. Hajare, "A method of robust multivariate outlier replacement", *Comput. Chem. Eng.*, vol. 26, no. 1, pp. 17–39, 2002.

[128] J.H. Chen and J.L. Liu, "Derivation of function space analysis based PCA control charts for batch process", *Chemical Engineering Science*, vol. 56, no. 10, pp. 3289–3304, May 2001.

[129] M. Kano, S. Hasebe, I Hashimoto, and H. Ohno, "A new multivariate process monitoring method using principal component analysis", *Computers and Chemical Engineering*, vol. 25, no. 7-8, pp. 1103–1113, August 2001.

[130] C. Rosen and J.A. Lennox, "Multivariate and multiscale monitoring of wastewater treatment operation", *Water Res.*, vol. 35, no. 14, pp. 3402–3410, 2001.

[131] J.H. Chen and K.C. Liu, "On-line batch process monitoring using dynamic pca and dynamic pls models", *Chem. Eng. Sci.*, vol. 57, no. 1, pp. 63–75, 2002.

[132] S.J. Qin, S. Valle, and M.J. Piovoso, "On unifying multiblock analysis with application to decentralized process monitoring", *J. Cheometrics*, vol. 15, no. 9, pp. 715–742, 2001.

[133] P. Paclík, J. Novovičová, P. Pudil, and P. Somol, "Road sign classification using Laplace kernel classifier", *Pattern recognition letters*, vol. 21, no. 13/14, pp. 1165–1174, 2000.

[134] R.H. Elliot, L. Assoun, and J.B. Moore, *Hidden Markov Models*, Springer, New York, 1995.

[135] R. Kulhavý and P.I. Ivanova, "Memory-based prediction in controland optimization", in *Proceedings of IFAC World Congress*, pp. 469–485. Pergamon, Oxford, 1999.

[136] B. Lennox, G.A. Montague, A.M. Frith, C. Gent, and V. Bevan, "Industrial application of neural networks - an investigation", *J. of Process Control*, vol. 11, no. 5, pp. 497–507, 2001.

[137] P.V. Varde, S. Sankar, and A.K. Verma, "An operator support system for research reactor operations and fault diagnosis through a connectionist framework and psa based knowledge based systems", *Reliabl. Eng. Syst. Safety*, vol. 60, no. 1, pp. 53–69, 1998.

[138] A. Quinn, P. Ettler, L. Jirsa, I. Nagy, and P. Nedoma, "Probabilistic advisory systems for data-intensive applications", *Inf. Journal of Adaptive Control and Signal Processing*, 2003, in print.

[139] P. Ettler and F. Jirkovský, "Digital controllers for škoda rolling mills", in *Lecture Notes: Advanced Methods in Adaptive Control for Industrial Application (Joint UK-CS seminar)*, K. Warwick, M. Kárný, and A. Halousková, Eds., vol. 158, pp. 31–35. Springer Verlag, 1991.

[140] J. Harbert, W. Eckelman, and R. Neumann, *Nuclear Medicine. Diagnosis and Therapy*, Thieme Medical Publishers, Inc, New York, 1996.

[141] G.A. Siegel, S.R. Thomas, J.B. Stubbs, M.G. Stabin, and M.T. Hay, "MIRD pamphlet No.16, Technics for quantitative radiopharmaceutical bio-distribution data aquisition and analysis", *Journal of Nuclear Medicine*, vol. 40, no. 2, 1999, Supplement.

[142] J. Heřmanská, K. Vošmiková, L. Jirsa, M. Kárný, and M. Šámal, "Biophysical inputs into the software MIRDose", *Sborník lékařský*, vol. 99, no. 4, pp. 521–527, 1998.

[143] J. Heřmanská, M. Kárný, J. Zimák, L. Jirsa, and M. Šámal, "Improved prediction of therapeutic absorbed doses of radioiodine in the treatment of thyroid carcinoma", *The Journal of Nuclear Medicine*, vol. 42, no. 7, pp. 1084–1090, 2001.

[144] E. L. Sutanto, J. D. Mason, and K. Warwick, "Mean-tracking clustering algorithm for radial basis function centre selection", *International journal of Control*, vol. 67, no. 6, pp. 961–977, August 1997.

[145] M. Kárný, P. Nedoma, I. Nagy, and M. Valečková, "Initial description of multi-modal dynamic models", in *Artificial Neural Nets and Genetic Algorithms. Proceedings*, V. Kůrková, R. Neruda, M. Kárný, and N. C. Steele, Eds., Wien, April 2001, pp. 398–401, Springer.

[146] L. Jirsa and A.Quinn, "Mixture analysis of nuclear medicine data: Medical decision support", in *Irish Signals and Systems Conference 2001, Proceedings*, R. Shorten, T. Ward, and T. Lysaght, Eds., Maynooth, 2001, pp. 393–398, NUI Maynooth.

[147] J. Zimák, J. Heřmanská, N. Medić, L. Jirsa, M. Kárný, M. Šámal, and L. Kárná, "Guide to *Iodine III*", Tech. Rep., Clinic of Nuclear Medicine, Motol Hospital, Praha, 2001, 28 pp.

[148] D. Helbing, "Traffic data and their implications for consistent traffic flow modelling", in *Transportation Systems*. 1997, pp. 809–814, Chania.

[149] F. McLeod, N. Hounsell, and Saeed Ishtia, "Traffic data and their implications for consistent traffic flow modelling", in *Transportation Systems*. 1997, pp. 911–916, Chania.

[150] Andrej Rakar, Petr Nedoma Tatiana V.G̃uy, Miroslav Kárný, and Dani Juričić, "Advisory system prodactool: Case study on gas conditioning unit", *Journal of Process Control*, 2003, submitted.

Index

Root file = pror.tex, May 26, 2003, responsible = MK