Akademie věd České republiky
Ústav teorie informace a automatizace

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Josef Andrýsek

Estimation of Dynamic Probabilistic Mixtures

No. 2150 November 29, 2005

GA ČR 102/03/0049
AV ČR S1075351, 1ET 100 750 401

ÚTIA AVČR, P.O.Box 18, 182 08 Prague,
Czech Republic
Fax: (+420)286890378, http://www.utia.cas.cz, E-mail: utia@utia.cas.cz

This report constitutes a non-referred software description. Opinions and conclusions expressed in this report are those of the author(s) and do not necessarily represent the views of the Institute.

Acknowledgement: This work was supported by

- GA ČR 102/03/0049
- AV ČR S1075351
- AV ČR 1ET 100 750 401.

Contents

Symbols and Notations

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

$\mathring{x}$ *denotes the number of entries in the vector* x.

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

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

$x_{i;t}$ *is an* i*-th entry of the vector* x_t. The semicolon in the subscript indicates that the symbol following it is the time index.

$x_{k\_l;t}$ *is a subvector of the vector* x_t. $x_{k\_l;t} = (x_{k;t}, \cdots, x_{l;t})$.

$x(k\_l) \equiv x_k, \ldots, x_l$.

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

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

d *is data array,* d_t *is data record at time* t *(vector with entries* $(d_{1;t}, \cdots, d_{\mathring{d};t}$ *)).*

$\mathring{t}$ *is finite time horizon, see Section 2.1.*

ϕ_{t-1} *is state vector, see Section 2.1 .*

ψ_t *is regression vector, see (4.7) .*

Ψ_t *is data vector, see Agreement 4.*

Θ is *unknown parameter, finite-dimensional array.*

f, π, ρ *are the letters reserved for probability density functions (pdf).*

- $f(d_t|d(t-1), \Theta)$ *means parameterized model of the system.*
- $f_c(d_t|d(t-1), \Theta_c)$ *is parameterized component of the mixture.*
- $\pi_0(\Theta)$ *denotes prior density of the unknown parameter* Θ.
- $\pi_t(\Theta|d(t)) \equiv \pi_t(\Theta|\mathcal{G}_t)$ *means (approximate) posterior density of the parameter* Θ *determined by the sufficient statistic* $\mathcal{G}_t$.
- $\rho(\Omega|\mathcal{H}_{t-1})$ *means (approximate) posterior density of the parameter* Ω *determined by the statistic* $\mathcal{H}_{t-1}$.

$\mathcal{G}_t, \mathcal{S}_{ic;t}, \mathcal{H}_t$ *are general statistics of (approximate) posterior pdf.*

$\propto$ *is the proportion sign,* $h \propto g$ *means that function* h *equals to the function* g *up to the normalization. i.e.* $\frac{h}{\int h} = \frac{g}{\int g}$.

∂ *is the model order.*

$\mathcal{D}(\cdot||\cdot)$ *means the Kullback-Leibler divergence [1].* $\mathcal{D}\left(f \;\middle||\; g\right) = \int f \ln\left(\frac{f}{g}\right)$. *It is also referred to as the KL divergence. See Section C.2.*

$\mathcal{K}(\cdot||\cdot)$ *means the Kerridge divergence [2].* $\mathcal{K}\left(f \;\middle||\; g\right) = -\int f \ln(g)$. *See Section C.4.*

$\Gamma(x)$ *means gamma function,* $\Gamma(x) = \int_0^{+\infty} t^{x-1}\exp(-t)dt$.

$\bullet$ *is used as a placeholder when specifying submatrix of a matrix. See Agreement 2.*

$\psi_0(x), \psi_1(x)$ *are digamma and trigamma functions,* $\psi_0(x) = \frac{\partial \ln\Gamma(x)}{\partial x}$, $\psi_1(x) = \frac{\partial \psi_0(x)}{\partial x}$.

δ *denotes identity matrix. i.e.* $\delta_{ij} = 1$ *iff* $i = j$, *otherwise* $\delta_{ij} = 0$.

$\otimes$ *denotes the Kronecker product of two matrices*

GiW *denotes Gauss-inverse-Wishart (GiW) pdf, see Section C.5.2.*

V *is statistic of GiW pdf, symmetric, positive definite matrix, see Section C.5.2.*

$\lfloor^{\psi}V$, $\lfloor^{d\psi}V$, $\lfloor^{d}V$ *denote submatrices of matrix* V, *see (C.15).*

ν *is statistic of GiW pdf, positive scalar, see Section C.5.2.*

L *is part of* $L'DL$ *decomposition, lower triangular matrix with units on diagonal, see Section C.5.2.*

$\lfloor^{\psi}L$, $\lfloor^{d\psi}L$ *denote submatrices of the matrix* L, *see (C.16).*

D *is part of* $L'DL$ *decomposition, diagonal matrix with positive diagonal, see Section C.5.2.*

$\lfloor^{\psi}D$, $\lfloor^{d}D$ *denotes submatrices of the matrix* D, *see (C.16).*

C, $\hat{\theta}$ *are alternative statistics of GiW pdf, see (C.18) and (C.17).*

$\mathcal{N}$ *denotes Gaussian pdf, see Section C.5.2.*

M *is statistic of Gaussian pdf, finite dimensional vector, see (C.6).*

R *is statistic of Gaussian pdf, symmetric, positive definite matrix, see (C.6).*

κ *is statistic of Dirichlet pdf, vector with positive elements, see Section C.2.*

α *is component weighting function, see Section 4.1.*

Ω *is parameter of component weighting function, see Section 4.1.*

$'$ *denotes transposition of a matrix.*

Agreement 1 (Generalization of matrix) *Within this text, we index general mathematical objects in the same manner as matrices. For example* $\Theta_{1,1}$ *is first element of "generalized matrix"* Θ *and can be arbitrary mathematical object. This notation is analogical to cell matrices in MATLAB.*

Agreement 2 (Indexing of (generalized) matrices) *For* M *being a (generalized) matrix of type* m, n *the following notation is used:*

M_{ij} *is* ij*-th entry of* M.

$M_{\bullet j}$ *is (generalized) matrix* $\begin{pmatrix} M_{1j} \\ \vdots \\ M_{mj} \end{pmatrix}$.

$M_{i\bullet}$ *is (generalized) matrix* $(M_{i1}, \cdots, M_{in})$.

$M_{\bullet\bullet}$ *means the same as* M*.We use this notation when we want to stress that* M *is a (generalized) matrix.*

Agreement 3 (Other matrix notations) *Let* M *be a matrix of type* m, n *and* c *some scalar. Let us define the following operations:*

$M \pm c$ *is matrix of type* m, n, $(M \pm c)_{ij} = M_{ij} \pm c$.

$\exp(M)$ *is matrix of type* m, n, $(\exp(M))_{ij} = \exp(M_{ij})$.

$\max M$ *is scalar with maximal value of* M.

$|M|$ *is determinant of matrix* M.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

This work has its origin in the EU grant ProDaCTool, which stands for Probabilistic Data Clustering Tool. The aim of the project was to develop an advisory system for operators of complex systems. Typically, an operator observes many variables indicating state of the system. His task is to manage the system, i.e. perform necessary actions based on the observations. Experienced operator is trained to detect abnormal behavior of the system and react appropriately. However, his experience can not be expressed by simple rules. Therefore it is not easy to share this knowledge with the new unexperienced operators.

The main assumption of the ProDaCTool project is that the experience of the operators is reflected in the historical data. If this assumption is true, then it is possible to create an advisory system, which will guide an unexperienced operator by suggesting solutions that were successful in the past. Moreover, the current data will also be incorporated into the advisory system to improve quality of advising in the future. This is known as adaptivity.

The advising problem can be formalized as a task of optimized dynamic decision making. The challenge is to process huge amount of historical data in such a way that reveals the operator's experience. This could be achieved by a detailed analysis of the specific application domain using as much expert knowledge as possible. Such analysis can be time consuming and expensive task, moreover its results cannot be used in other application domains. Therefore, this approach is suitable only for large companies, where benefits of the analysis will pay off. However, in many application domains this approach is too expensive or risky.

The aim of the ProDaCTool project was to prepare a general theoretical and software background, that will be applicable to many various application domains. The project was successfully finished in 2003 and the approach was applied in industry (operating cold rolling mill [3]), medicine (treatment of thyroid glance cancer [4]), traffic control (prediction of traffic flow [5]) and society (modelling of a fair governing in connection with e-democracy [6]).

In order to make the solution domain-independent, detailed physical modelling of the problem is not possible, hence the system is modelled by a black-box model. A general parametric model is chosen and its parameters are estimated to match the observed data as close as possible. The choice of the parametric model is essential for success of the approach. Too simple parametric model has a low descriptive power and too complex parameterized model is not analytically tractable. Hence, we seek a compromise between descriptive power of the model and its analytical tractability.

The probabilistic models were chosen as a base class of parametric models. The advantage of probabilistic models is the availability of compact theoretical solution of all tasks related to model learning, which is known as the Bayesian theory [7]. This theory allows finding the compromise mentioned above [8]. Moreover, Bayesian recursive learning of model parameters provides the desired adaptivity of the advisory system.

1.2 Problem Formulation

The basic model used in the ProDaCTool project is a probabilistic mixture. It was chosen for the following reasons: i) it provides a universal approximation of almost any probability density function [9], ii) the tasks of control and decision making with mixture models are computationally tractable [10].

The mixture model is a convex combination of simpler models called components, the coefficients of the convex combination are called component weights. If the components model the temporal dependency of data samples, we speak about dynamic components, otherwise, we speak about static components. Similarly, if the component weights depends on historical data, they are called dynamic, otherwise, they are called static.

In the ProDaCTool project, mixtures with dynamic components and static weights were used. Exact Bayesian inference of their parameters is not tractable and some approximations of Bayesian learning has to be used. The quasi-Bayes approximation [11] was exploited to solve this task. This approach was successfully used in many application domains [3, 4, 5, 6], however, for some data sets this approach does not provide an acceptable solution. This can be due to two reasons: i) the quasi-Bayes approximation is too coarse, or ii) the descriptive power of the model is not sufficient. The aim of this work is to address these two issues as follows: i) to develop a better approximation for inference of parameters of mixtures with static weights, ii) to find a richer model than mixtures with static weights and to develop an adequate approximate inference method.

1.3 State of the Art

1.3.1 Inference of Mixture-model Parameters

Rich literature on inference of probabilistic mixtures with static components and static weights is available [9, 12, 13, 14, 15, 16]. These models are appropriate for sequences of independent observation [9]. They are related to clustering [17], neural networks [18] or principal component analysis (PCA) [19]. However, the static mixtures are not sufficiently rich for the considered advisory system.

Inference of probabilistic mixtures with dynamic components and static weights is more demanding task and only a little work was published in this area [20]. The quasi-Bayes algorithm [21] developed for static probabilistic mixtures has been generalized to cope with dynamic components [10]. Theoretical justification of the quasi-Bayes modification is missing.

Particle filters [22] can be efficiently used for estimation of parameters of arbitrary probability density function. They are based on Monte-Carlo techniques and their use is limited to low dimensional cases only. Another general approach is based on mean field methods [23], which provide promising approximation techniques. Especially, the variational Bayes (VB) approach [24] provides a systematic and applicable solution. It is based on minimization of Kullback-Leibler divergence [1]. Since this divergence is not symmetric, the result of optimization depends on the selected argument order of this divergence. Theoretical analysis [25, 26] suggest that one argument order provides a better approximation. However, the VB approach uses the opposite order of arguments, which allows to find an analytical solution [27].

The theoretical analysis [25, 26] motivates our search for an approximation minimizing the KL divergence with recommended argument order. It may not be possible to derive such general results as the VB approach, but it may be possible to derive inference algorithms for special but important classes of pdfs. This approach will be used to address the tasks of this work.

1.3.2 Other Classes of Models

Naturally, there are competitive ways of modelling the dependency of data samples. For example dynamic versions of PCA [28, 29, 30] or neural networks [18]. PCA provides probabilistic model of the system, but it can represent unimodal pdfs only. Hence it can not be used instead of probabilistic mixtures. It can be used as a mixture component, but the problem with static component weights remains.

Neural networks (NN) serve as universal approximations of multivariate, generally non-linear mappings [31]. As such, they provide non-linear black-box dynamic models used in various decision-supporting modules, for instance, as standards in fault detection or as predictors [32]. They are extensively used so that their advantages and limitations can be studied on real cases [33]. Unfortunately, NN does not provide probabilistic description of the system and thus can not be exploited to solve our task.

Other important approaches to probabilistic models are based on nonparametric Bayesian estimation [34, 35]. They are mostly used for simple static cases. Important representant of nonparametric classes are gaussian priors and mixtures of them [36]. These models look very promising, but still, the complexity of systems, which this approach is tractable for, is limited.

To our best knowledge, none of the existing system models is equivalent with probabilistic mixtures in the sense of tractability, description power and suitability for subsequent control or decision-making tasks. This forces us to stay within the class of probabilistic mixture models. It was proven [10] that probabilistic mixtures with dynamic component and static weights describes all dynamic probability distributions only asymptotically. There were also attempts [10] to estimate dynamic mixtures with specific types of dynamic weights, but a general framework is missing. This leads to the need for introducing general probabilistic mixtures with both dynamic components and dynamic weights and developing an appropriate estimating algorithm.

1.4 Aims of the Work

The two main problems addressed within this text were already mentioned. Firstly, we need to improve estimation of dynamic probabilistic mixtures with static weights. Secondly, we need to improve the mixture model to work with data-dependent component weights.

As the static-weights mixtures are a special case of dynamic-weights mixtures, both these tasks can be solved within a single general framework. Estimation algorithm for static-weights mixtures will be then obtained by specialization of the general algorithm. The specific tasks of the work are:

- to define dynamic probabilistic mixture model with dynamic weights as a generalization of the current dynamic mixture with static weights,
- to elaborate a general algorithm for recursive estimation of the generalized model,
- to apply the algorithm to specific types of components and component weighting functions,
- to specialize the algorithm for mixtures with static weights,
- to implement all algorithms in MATLAB,
- to implement algorithms for static-weights mixtures in C and integrate them into MATLAB toolbox Mixtools,
- to compare quality of the new algorithm with the current quasi-Bayes algorithm on a large set of examples dealing with estimation of a static-weights mixture,
- to test reliability of algorithms for dynamic-weights mixtures on simple examples.

1.5 Thesis Layout

Chapter 1 summarizes the aims of this work. Also, the means used to achieve this aims are presented here.

The underlaying Bayesian estimation is discussed in Chapter 2.

In Chapter 3, general useful propositions about projection into two important classes are proved.

Chapters 4, 5, 6 and 7 form the core of the work. The two main problems of the work are discussed and solved here.

Chapter 4 provides a specific problem formulation.

General techniques describing solution of the formulated problem form the content of Chapter 5. The problem is split into two subproblems: (i) optimizing of factors statistics and (ii) optimizing of statistics determining the component weighting functions.

Chapters 6 and 7 solves the mentioned subproblems (i) and (ii).

Content of Chapter 8 is formed by experiments demonstrating and verifying the theoretical results.

Chapter 9 concludes the work by summarizing the status of the research achieved and lists some problems to be addressed in future.

Appendix A recalls the quasi-Bayes estimation algorithm, which serves as a reference for quality comparison. Also the algorithm `mixinit` for initialization of mixture estimation is briefly described there.

General mathematical tools used, auxiliary propositions and properties of polygamma functions are summarized in Appendix B.

Appendix C summarizes basic properties and propositions of probabilistic calculus, The Kullback-Leibler divergence, the Kerridge divergence and their properties as well as important probability density functions and their properties.

Appendix D describes normal autoregressive factors and their Bayesian estimation.

In order to provide compact text, majority of propositions, definitions and pdf properties are placed in the appendices. Inside the main text, references to them are made. This style of presentation may be little bit confusing for the readers who are beginners in area of probabilistic modelling. To minimize this confusion, Section 1.6 briefly summarizes most of the terms, which will be referred within the main text.

1.6 Means and Tools Used for the Work

Here, the main tools and means used for the work are summarized.

- Basic properties of probability density functions (Appendix C)
  - Conditioning (Proposition 19)
  - Jensen inequality (Proposition 20)
  - Mean value transformation (Proposition 21)
  - Marginalization (Proposition 19)
  - Chain rule (Proposition 19)
- Properties of known pdfs
  - Gaussian pdf (Section C.6)
  - Dirichlet pdf (Section C.2)
  - Gauss-inverse Wishart pdf (Section C.5)
- Bayesian estimation (Chapter 2)
  - Prior, posterior pdf (Section 2.1)

  - Bayes rule, Bayesian updating (Section 2.2)
  - Conjugate prior, conjugate posterior (Section 2.3)

- Proximity measures (Appendix C)
  - Kullback-Leibler divergence (Section C.1.1)
  - Kerridge divergence (Section C.1.2)

- General mathematical tools (Appendix B)
  - Matrix differential calculus (Section B.1)
  - Extremes of multivariate functions (Proposition 10)
  - Monte-Carlo integral evaluation (Section 7.2.5)
  - Polygamma functions (Appendix B.3)

This text doesn't have ambitions to provide exact mathematical description of the presented propositions and their proofs. Instead, it tries to present them as simply as possible in the form close to their software implementation.

Chapter 2

Bayesian Estimation

This chapter starts with a description of Bayesian estimation. Then its feasibility is discussed. Finally, general mechanism for achieving feasibility of Bayesian recursive estimation is proposed.

2.1 General Description of Bayesian Estimation

Let us have some process with $\mathring{d}$ scalar sensors called here data channels. Current values on all data channels at time t form a $\mathring{d}$-dimensional data vector $d_t \equiv [d_{1;t}, \cdots, d_{\mathring{d};t}]$. We measured values on all data channels for $\mathring{t}$ times and got data $d(\mathring{t}) \equiv (d_1, \cdots, d_{\mathring{t}})$.

Probabilistic modelling relies on assumption that $d(t)$ is a random quantity. Then the task of estimation is defined as finding the probability density function (pdf) of this random quantity. It means that our task is to find pdf $f(d(\mathring{t}))$. Because this task is enormously difficult, we usually assume that $f(d(\mathring{t}))$ belongs to some known class of pdfs determined by finite dimensional parameter Θ, $f(d(\mathring{t})) \equiv f(d(\mathring{t})|\Theta)$. Then the task reduces to estimating the parameter Θ.

According to the chain rule (Proposition 19), we can factorize the pdf $f(d(\mathring{t})|\Theta)$ as follows:

$$f(d(\mathring{t})|\Theta) = \prod_{t=1}^{\mathring{t}} f_t(d_t|d(t-1), \Theta).$$

It is reasonable to expect that d_t does not depend on all historical values $d(t-1)$, but just on a subselection ϕ_{t-1} forming state vector, i.e.

$$f_t(d_t|d(t-1), \Theta) \equiv f_t(d_t|\phi_{t-1}, \Theta).$$

The state vector ϕ_{t-1} can be even empty. In such a case no dependence on past is considered and the model is called static. Otherwise, the model is called dynamic.

Next, it is often reasonable to expect that all pdfs f_t have the same functional form:

$$f_t(d_t|d(t-1), \Theta) \equiv f(d_t|\phi_{t-1}, \Theta).$$

The pdf $f(d_t|\phi_{t-1}, \Theta)$ is called parameterized model of the system and it of course fully determines the pdf $f(d(\mathring{t})|\Theta)$ considering the previous assumptions.

The basic principle of Bayesian decision making [7] states that uncertainty should be modelled by randomness. This means that unknown parameter Θ should be treated as a random quantity. If Θ is a random quantity it makes sense to speak about its pdf. The main interest of Bayesian analysis lies on studying the pdf of Θ conditioned by all known data $d(\mathring{t})$. This is so called posterior pdf $\pi(\Theta|d(\mathring{t}))$. This pdf is the main outcome of Bayesian estimation as it provides full information about the unknown parameter Θ. From practical reasons, we consider the posterior pdf $\pi(\Theta|d(\mathring{t}))$ to be determined by statistic $\mathcal{G}_{\mathring{t}}$ instead of all $d(\mathring{t})$. This assumption is very weak, because we do not assume the finiteness of

$\mathcal{G}_{\mathring{t}}$ yet. Hence $\pi(\Theta|d(\mathring{t})) \equiv \pi(\Theta|\mathcal{G}_{\mathring{t}})$. Important object of Bayesian estimation needed for evaluation of $\pi(\Theta|d(\mathring{t}))$ is also so called prior pdf $\pi(\Theta) \equiv \pi(\Theta|\mathcal{G}_0)$ reflecting our knowledge about the system before the estimation. This pdf can be constructed using information of some experts. The expert information must be of course translated into probabilistic terms [37].

According to previous considerations, we can formulate the task of Bayesian parameter estimation:

Provide the posterior pdf $\pi(\Theta|\mathcal{G}_{\mathring{t}})$, using the knowledge of:

- $\mathring{t}$ data records (realizations) $d(\mathring{t})$,
- the prior pdf $\pi(\Theta)$,
- the parameterized model $f(d_t|\phi_{t-1}, \Theta)$.

In practical application we often need to update the posterior pdf $\pi(\Theta|\mathcal{G}_{t-1})$ with each new data record d_t. This task can be formulated as follows:

Provide the posterior pdf $\pi_t(\Theta|\mathcal{G}_t)$, using the knowledge of:

- state vector ϕ_{t-1},
- new data record d_t,
- the parameterized model $f(d_t|\phi_{t-1}, \Theta)$,
- old posterior pdf $\pi_{t-1}(\Theta|\mathcal{G}_{t-1})$.

This task is called Bayesian recursive estimation and is the key problem addressed within this text. It is simple to observe, that non-recursive version of estimation can be obtained by repetitive use of the recursive version.

The following example illustrates some terms defined in previous paragraph.

Example 1 (Bayesian estimation)

$$\begin{array}{rcll} \mathring{d} & = & 1 & \textit{(scalar data)} \\ \phi_{t-1} & \equiv & (d_{t-1}, d_{t-2}) & \textit{(state of the model)} \\ \Theta & \equiv & (a, b, c) & \textit{(unknown parameter)} \\ f(d_t|\phi_{t-1}, \Theta) & = & \mathcal{N}_{d_t}\left(ad_{t-1} + bd_{t-2} + c, 1\right) & \textit{(normal parameterized model)} \\ \pi_0(\Theta|\mathcal{G}_0) & \equiv & U_a(0,2)U_b(1,3)U_c(-1,1) & \textit{(uniform prior pdf)} \end{array}$$

We have some scalar system. Data record d_t at time t depends on two historical values d_{t-1} and d_{t-2}. We measured $\mathring{t}$ data records $d_1, \cdots, d_{\mathring{t}}$. We do not know the values of parameters a, b, c, but the prior pdf says that $a \in (0,2)$, $b \in (1,3)$, $c \in (-1,1)$. We need to know more about them. The posterior pdf $\pi_{\mathring{t}}(a, b, c|\mathcal{G}_{\mathring{t}})$ will give us better information.

2.2 Solution of Bayesian Recursive Estimation

Bayesian recursive estimation has a simple solution:

$$\pi_t(\Theta|\mathcal{G}_t) = \frac{f\big(d_t|\phi_{t-1}, \Theta\big)\pi_{t-1}(\Theta|\mathcal{G}_{t-1})}{\int f\big(d_t|\phi_{t-1}, \Theta\big)\pi_{t-1}(\Theta|\mathcal{G}_{t-1})d\Theta}. \tag{2.1}$$

The pdf $f\big(d_t|\phi_{t-1}, \Theta\big)$ is taken as a function of Θ. The data record d_t and state vector ϕ_{t-1} must be known. The following example demonstrates use of this relation.

Example 2 (Bayesian recursive estimation)

$$\begin{array}{rcll} \mathring{d} & = & 1 & \textit{(scalar data)} \\ \phi_{t-1} & \equiv & \emptyset & \textit{(static model)} \\ \Theta & & & \textit{(unknown scalar parameter)} \\ f(d_t|\phi_{t-1},\Theta) & \equiv & \mathcal{N}_{d_t}(\Theta,1) & \textit{(normal parameterized model)} \\ \pi_{t-1}(\Theta|\mathcal{G}_{t-1}) & \equiv & \mathcal{N}_{\Theta}(M_{t-1},R_{t-1}) & \textit{(Gaussian old posterior pdf)} \end{array}$$

According to the relation (2.1), the new posterior pdf is:

$$\pi_t(\Theta|M_t,R_t) = \frac{\mathcal{N}_{d_t}(\Theta,1)\,\mathcal{N}_{\Theta}(M_{t-1},R_{t-1})}{\int \mathcal{N}_{d_t}(\Theta,1)\,\mathcal{N}_{\Theta}(M_{t-1},R_{t-1})\,d\Theta}.$$

With a simple computation, we obtain the result:

$$\pi_t(\Theta|M_t,R_t) = \mathcal{N}_{\Theta}\left(\frac{M_{t-1}+R_{t-1}d_t}{R_{t-1}+1}, \frac{R_{t-1}}{1+R_{t-1}}\right).$$

The new posterior pdf $\pi_t(\Theta|M_t,R_t)$ has the same functional form as the old one. This fact is very important, because Bayesian update reduces here to updating of statistics M_t, R_t, i.e. it consists of the mapping $(M_{t-1},R_{t-1},d_t) \to (M_t,R_t)$ defined as follows:

$$M_t = \frac{M_{t-1}+R_{t-1}d_t}{R_{t-1}+1}, \quad R_t = \frac{R_{t-1}}{1+R_{t-1}}.$$

Now let us assume we have the prior pdf $\pi_0(\Theta|M_0,R_0) \equiv \mathcal{N}_{\Theta}(M_0,R_0)$ and apply the rule repeatedly in $\mathring{t}$ time steps. We get the posterior pdf $\pi_{\mathring{t}}(\Theta|M_{\mathring{t}},R_{\mathring{t}}) \equiv \mathcal{N}_{\Theta}(M_{\mathring{t}},R_{\mathring{t}})$.

Let us simulate $\mathring{t} \equiv 10$ data records with $\Theta_{true} \equiv 2.0000$. The result of simulation is displayed in Table 2.1.

t	1	2	3	4	5
d_t	1.7271	0.9745	3.0329	1.0502	1.4423
t	6	7	8	9	10
d_t	1.4322	1.2444	2.7505	1.7242	4.9642

Table 2.1: Simulated data

If we select relatively flat prior pdf given by

$$M_0 = 0.0000,\ R_0 = 5.0000,$$

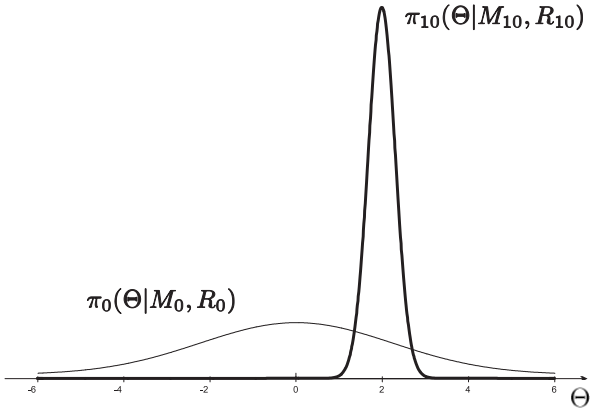
we obtain the result

$$M_{10} = 1.9944,\ R_{10} = 0.0980.$$

The prior pdf $\pi_0(\Theta|M_0,R_0)$ and posterior pdf $\pi_{10}(\Theta|M_{10},R_{10})$ as well as values of M_t and R_t during the estimation are depicted on Figure 2.1. Note that the posterior pdf is concentrated near the true value 2.0000.

2.3 Feasibility of Bayesian Estimation

In Example 2, the Bayesian estimation leads to simple recursion on statistics M_t and R_t. Unfortunately, this happens only in a very limited number of cases, when the new posterior pdf $\pi_t(\Theta|\mathcal{G}_t)$ after one step of estimation preserves the same functional form as the previous posterior pdf $\pi_{t-1}(\Theta|\mathcal{G}_{t-1})$. Then we can omit the time subscript in the pdf, i.e. $\pi_t(\Theta|\mathcal{G}_t) \equiv \pi(\Theta|\mathcal{G}_t)\forall t$. When updating from $\pi(\Theta|\mathcal{G}_{t-1})$



t	M_t	R_t
0	0.0000	5.0000
1	1.4392	0.8333
2	1.2280	0.4545
3	1.7920	0.3125
4	1.6154	0.2381
5	1.5821	0.1923
6	1.5579	0.1613
7	1.5144	0.1389
8	1.6651	0.1220
9	1.6715	0.1087
10	1.9944	0.0980

Figure 2.1: Example of Bayesian estimation
The figure shows the prior pdf $\pi_0(\Theta|M_0, R_0)$ and resulting posterior pdf $\pi_{10}(\Theta|M_{10}, R_{10})$ after processing 10 data records. It can be seen that the posterior pdf concentrated near the true value 2.0000. The table in the right part of the figure shows evolution of the posterior statistics during time.

to $\pi(\Theta|\mathcal{G}_t)$, it suffices to update the statistic $\mathcal{G}_t$. The prior pdf, which leads to this behavior is called *conjugate*[38] .

In this text, we will orient on the case when the conjugate pdf does not exist. Then we have to face to two major problems.

- The normalizing integral in (2.1) need not be analytically solvable.
- Repetitive use of this rule would lead to very complex forms of the posterior pdf.

The first problem can be solved by approximation of the integral or using numeric integration. Solution of the second problem is much more difficult. We will demonstrate this problem on a simple example.

Example 3 (Not suitable Bayesian estimation)

$$\begin{aligned}
\mathring{d} &= 1 && \textit{(scalar data)}\\
\phi_{t-1} &\equiv \emptyset && \textit{(static model)}\\
\Theta &\equiv (a,b) && \textit{(unknown parameter)}\\
f(d_t|\phi_{t-1},\Theta) &\equiv 0.5\mathcal{N}_{d_t}(a,1) + 0.5\mathcal{N}_{d_t}(b,1) && \textit{(parameterized model)}\\
\pi_0(a,b|\mathcal{G}_0) &\equiv \mathcal{N}_{(a,b)}(M_0,R_0) && \textit{(Gaussian prior pdf)}
\end{aligned}$$

According the Bayes rule (2.1):

$$\pi_1(a,b|\mathcal{G}_1) \propto 0.5\mathcal{N}_{d_1}(a,1)\,\mathcal{N}_{(a,b)}(M_0,R_0) + 0.5\mathcal{N}_{d_1}(b,1)\,\mathcal{N}_{(a,b)}(M_0,R_0)\,.$$

With a little simple computation:

$$\pi_1(a,b|\mathcal{G}_1) = w_1\mathcal{N}_{(a,b)}\left(M_1^{(1)},R_1^{(1)}\right) + (1-w_1)\mathcal{N}_{(a,b)}\left(M_1^{(2)},R_0^{(2)}\right),$$

where $w_1, M_1^{(1)}, M_1^{(2)}, R_1^{(1)}, R_1^{(2)}$ are evaluated somehow. Details are not important now. Important is that π_1 is a weighted sum of two pdfs of the same type as π_0. It is simple to observe, that π_2 would be a weighted sum of two pdfs of the same type as π_1, i.e. it will be a sum of 4 pdfs of the type π_0. Generally, π_t would consist of 2^t weighted terms. It is clear that we are not able to store the statistics of these terms in computer even for a relatively small t.

In the previous example, we were able to perform analytically one estimation step, but we were not able to use its result in the next estimation steps. A simple way out of this situation is to approximate the new posterior pdf to obey the same form as the old posterior pdf. Now let us formalize the Bayesian estimation using this trick.

Off-line phase

- Choose sufficiently rich class of posterior pdfs, element of this class is determined by finite statistic $\mathcal{G}_t$. $\pi(\Theta|\mathcal{G}_t)$
- Set $\mathcal{G}_0$ so that $\pi(\Theta|\mathcal{G}_0)$ reflects the prior information.

On-line phase

- Evaluate one step of the Bayesian estimation (2.1), getting

$$\hat{\pi}_t(\Theta) = \frac{f(d_t|\phi_{t-1},\Theta)\pi(\Theta|\mathcal{G}_{t-1})}{\int f(d_t|\phi_{t-1},\Theta)\pi(\Theta|\mathcal{G}_{t-1})d\Theta}. \tag{2.2}$$

  This pdf will be referred to as correct update and it is usually out of our class.
- Find $\mathcal{G}_t$ so that $\pi(\Theta|\mathcal{G}_t)$ is the best projection of the obtained pdf $\hat{\pi}_t$ into our class of posteriors.

The term "best projection" is a little bit vague. Within this text, under this term we will consider exclusively minimizer of Kullback-Leibler divergence [1]. So the task can be more precisely formulated as follows:

Find $\mathcal{G}_t$ so that

$$\mathcal{D}\left(\hat{\pi}_t(\Theta) \,\middle|\middle|\, \pi(\Theta|\mathcal{G}_t)\right)$$

is minimal.

Note that this divergence is not symmetric in order of its arguments. There exist approaches minimizing the other argument order[39], because more or less analytical solution can be found [40]. We choose KL divergence and this argument order, because it is compatible with the Bayesian methodology [25, 26]. The feasible solution is not guaranteed at general level, but it is possible to find the minimizer for special cases. Because the specified approach finds the best projection into specific classes, it is called projection based approach.

Example 4 (Projection based approach) *Let us have the same parameterized model and prior pdf as in Example 3. Now we will force the posterior pdf to stay within the class of 2-dimensional Gaussian distributions.*

$$\begin{array}{rcll} \mathring{d} & = & 1 & \textit{(scalar data)} \\ \phi_{t-1} & \equiv & \emptyset & \textit{(static model)} \\ \Theta & \equiv & (a,b) & \textit{(unknown parameter)} \\ f(d_t|\phi_{t-1},\Theta) & \equiv & 0.5\mathcal{N}_{d_t}(a,1) + 0.5\mathcal{N}_{d_t}(b,1) & \textit{(parameterized model)} \\ \pi(a,b|\mathcal{G}_{t-1}) & \equiv & \mathcal{N}_{(a,b)}(M_{t-1},R_{t-1}) & \textit{(Gaussian class of posteriors)} \end{array}$$

Similarly as in the previous example, we came to the relation

$$\hat{\pi}_t(a,b) = w_t\mathcal{N}_{(a,b)}\left(M_t^{(1)},R_t^{(1)}\right) + (1-w_t)\mathcal{N}_{(a,b)}\left(M_t^{(2)},R_t^{(2)}\right).$$

Now we have to find $\mathcal{G}_t \equiv (M_t,R_t)$ determining the best projection of this pdf to class of 2-dimensional Gaussian pdfs. Using Propositions 2 and 22 it can be found that

$$\begin{array}{rcl} M_t & = & w_tM_t^{(1)} + (1-w_t)M_t^{(2)} \\ R_t & = & w_tR_t^{(1)} + (1-w_t)R_t^{(2)} + w_t(1-w_t)(M_t^{(2)}-M_t^{(1)})(M_t^{(2)}-M_t^{(1)})', \\ & & \textit{where } ' \textit{ denotes transposition.} \end{array}$$

Chapter 3

Basic Tool

This chapter contains two important propositions exploited extensively during the work. They converts very complex problem of minimization the KL divergence into a simpler task of integration for two important classes.

Proposition 1 (Best projection into GiW class) *Let $f(\theta, r)$ be arbitrary joint pdf on vector θ and positive scalar r fulfilling following conditions.(The assumptions are not very restrictive and in obvious situations they are fulfilled.)*

$$\begin{aligned} p &\equiv \int \frac{f(\theta,r)}{r} d\theta dr \text{ is finite.} \\ s &\equiv \int \ln(r)\, f(\theta,r) d\theta dr \text{ is finite.} \\ h(\theta,r) &\equiv \frac{f(\theta,r)}{rp} \text{ has finite positive definite covariance matrix } \mathbf{cov}\left[\theta\right]_h. \end{aligned}$$

Then the statistics $(C, {}^{\lfloor d}D, \hat{\theta}, \nu)$ minimizing the KL divergence $\mathcal{D}\left(f(\theta,r) \,\middle\|\, GiW_{\theta,r}(C, {}^{\lfloor d}D, \hat{\theta}, \nu)\right)$ fulfill:

$$\begin{aligned} C &= p\ \mathbf{cov}\left[\theta\right]_h \\ \hat{\theta} &= \mathcal{E}\left[\theta\right]_h \\ \ln(0.5\nu) - \psi_0(0.5\nu) &= \ln(p) + s \\ {}^{\lfloor d}D &= \frac{\nu}{p} \end{aligned}$$

Proof: We will show that the specified statistics minimize the Kerridge divergence

$$\mathcal{K}\left(f(\theta,r) \,\middle\|\, GiW_{\theta,r}(C, {}^{\lfloor d}D, \hat{\theta}, \nu)\right) = -\int f(\theta,r) \ln\left(GiW_{\theta,r}(C, {}^{\lfloor d}D, \hat{\theta}, \nu)\right), \tag{3.1}$$

which, according to Proposition 23, directly implies the statement of the proposition.
GiW pdf has the following form:

$$\begin{aligned} GiW_{\theta,r}(C, {}^{\lfloor d}D, \hat{\theta}, \nu) &= \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(C, {}^{\lfloor d}D, \nu)} \exp\left\{-\frac{1}{2r}\left[(\theta-\hat{\theta})' C^{-1} (\theta-\hat{\theta}) + {}^{\lfloor d}D\right]\right\}, \text{where} \\ \mathcal{I}(C, {}^{\lfloor d}D, \nu) &= \Gamma(0.5\nu)\, {}^{\lfloor d}D^{-0.5\nu} \left|C^{-1}\right|^{-0.5} 2^{0.5\nu} (2\pi)^{0.5\mathring{\psi}}. \end{aligned}$$

We need to evaluate logarithm of GiW pdf:

$$\begin{aligned}\ln\left(GiW_{\theta,r}(C,\ {}^{\lfloor d}D,\hat{\theta},\nu)\right) &= -0.5(\nu+\mathring{\psi}+2)\ln(r)-\ln\left(\Gamma\left(0.5\nu\right)\right)+ \\ &+ 0.5\nu\ln\left({}^{\lfloor d}D\right)-0.5\mathring{\psi}\ln(2\pi)-0.5\nu\ln(2)+ \\ &+ 0.5\ln\left(|C^{-1}|\right)-\frac{1}{2r}\left[(\theta-\hat{\theta})'C^{-1}(\theta-\hat{\theta})+\ {}^{\lfloor d}D\right].\end{aligned} \tag{3.2}$$

Before substituting (3.2) into (3.1), we split it into two parts. The first part depends only on ν, ${}^{\lfloor d}D$, the second part depends on $\hat{\theta}, C$. The part, which does not depend on any of ν, ${}^{\lfloor d}D, \hat{\theta}, C$ is omitted. With this separation, the Kerridge divergence we are evaluating splits into

$$\mathcal{K}\left(f(\theta,r)\ \middle|\middle|\ GiW_{\theta,r}(C,\ {}^{\lfloor d}D,\hat{\theta},\nu)\right)=G(\nu,\ {}^{\lfloor d}D)+W(\hat{\theta},C)+const,\ \text{where}$$

$$\begin{aligned}&G(\nu,\ {}^{\lfloor d}D)= \\ &= -\int f(\theta,r)\left[-0.5\nu\ln(r)-\ln\left(\Gamma\left(0.5\nu\right)\right)+0.5\nu\ln\left(0.5\ {}^{\lfloor d}D\right)-\frac{1}{2r}\ {}^{\lfloor d}D\right]d\theta dr= \\ &= 0.5\nu\underbrace{\int f(\theta,r)\ln(r)\,d\theta dr}_{\equiv s}+\ln\left(\Gamma\left(0.5\nu\right)\right)-0.5\nu\ln\left(0.5\ {}^{\lfloor d}D\right)+0.5\ {}^{\lfloor d}D\underbrace{\int\frac{1}{r}f(\theta,r)d\theta dr}_{\equiv p}= \\ &= 0.5\nu s+\ln\left(\Gamma\left(0.5\nu\right)\right)-0.5\nu\ln\left(0.5\ {}^{\lfloor d}D\right)+0.5\ {}^{\lfloor d}Dp\end{aligned}$$

$$\begin{aligned}&W(\hat{\theta},C)= \\ &= -\int f(\theta,r)\left[0.5\ln\left(|C^{-1}|\right)-\frac{1}{2r}[(\theta-\hat{\theta})'C^{-1}(\theta-\hat{\theta})]\right]d\theta dr= \\ &= -0.5\ln\left(|C^{-1}|\right)+\int f(\theta,r)\frac{1}{2r}[\theta'C^{-1}\theta-2\hat{\theta}'C^{-1}\theta+\hat{\theta}'C^{-1}\hat{\theta}]d\theta dr\overbrace{=}^{Prop.7} \\ &= -0.5\ln\left(|C^{-1}|\right)+0.5\mathbf{tr}\left(C^{-1}\underbrace{\int\frac{\theta\theta'}{r}f(\theta,r)d\theta dr}_{\equiv Q}\right)-\hat{\theta}'C^{-1}\underbrace{\int\frac{\theta}{r}f(\theta,r)d\theta dr}_{\equiv U}+0.5p\ \hat{\theta}'C^{-1}\hat{\theta}= \\ &= -0.5\ln\left(|C^{-1}|\right)+0.5\mathbf{tr}\left(C^{-1}Q\right)-\hat{\theta}'C^{-1}U+0.5p\ \hat{\theta}'C^{-1}\hat{\theta}\end{aligned}$$

It is clear, that we can split the minimization task into two independent parts. The first part is searching for the optimal scalars ν, ${}^{\lfloor d}D$ and the second part is searching for the optimal matrix C and vector $\hat{\theta}$. The minimization will use the standard differential approach summarized in Proposition 10. Let us start with the first 2-dimensional minimization.

First we evaluate partial derivatives of G.

$$\frac{\partial G}{\partial\ {}^{\lfloor d}D}=-0.5\frac{\nu}{{}^{\lfloor d}D}+0.5p \tag{3.3}$$

$$\frac{\partial G}{\partial\nu}=0.5s+0.5\psi_0\left(0.5\nu\right)-0.5\ln\left(0.5\ {}^{\lfloor d}D\right) \tag{3.4}$$

$$\frac{\partial^2 G}{\partial\ {}^{\lfloor d}D^2}=0.5\frac{\nu}{{}^{\lfloor d}D^2} \tag{3.5}$$

$$\frac{\partial^2 G}{\partial\nu^2}=0.25\psi_1\left(0.5\nu\right) \tag{3.6}$$

$$\frac{\partial^2 G}{\partial \nu \partial \, \lfloor^d D} = -\frac{0.5}{\lfloor^d D} \tag{3.7}$$

$$\tag{3.8}$$

By zeroing the first derivatives, we obtain the equations for the optimal values:

$$\lfloor^d D = \frac{\nu}{p} \tag{3.9}$$

$$\ln(0.5\nu) - \psi_0(0.5\nu) = \ln(p) + s \tag{3.10}$$

According to Proposition 17, the equation (3.10) is known to have unique positive solution iff $\ln(p)+s > 0$. It holds:

$$\ln(p) + s = \ln\left(\int \frac{1}{r} f(\theta, r) d\theta dr\right) - \int \ln\left(\frac{1}{r}\right) f(\theta, r) d\theta dr = \tag{3.11}$$

$$= \ln\left(\mathcal{E}\left[\frac{1}{r}\right]\right) - \mathcal{E}\left[\ln\left(\frac{1}{r}\right)\right] \tag{3.12}$$

Applying the Proposition 21, Jensen inequality (Proposition 20) and assumptions of the current proposition on (3.12) gives that $\ln(p) + s > 0$.

We found unique stationary point. Lets investigate the definiteness of the Hessian.

$$H = \begin{pmatrix} 0.5\frac{\nu}{\lfloor^d D^2} & -\frac{0.5}{\lfloor^d D} \\ -\frac{0.5}{\lfloor^d D} & 0.25\psi_1(0.5\nu) \end{pmatrix}$$

According to the Proposition 12, we need to show that

$$0.5\frac{\nu}{\lfloor^d D^2} > 0 \tag{3.13}$$

$$\begin{vmatrix} 0.5\frac{\nu}{\lfloor^d D^2} & -\frac{0.5}{\lfloor^d D} \\ -\frac{0.5}{\lfloor^d D} & 0.25\psi_1(0.5\nu) \end{vmatrix} > 0. \tag{3.14}$$

The inequality (3.13) holds, because $\nu > 0$. The determinant is equal to: $\frac{0.25}{\lfloor^d D^2}(0.5\nu\psi_1(0.5\nu) - 1)$ which is positive, because the function $\nu\psi_1(\nu) > 1, \ \forall \nu > 0$ (Proposition 18).

We proved that there is unique local minima. Because the minimization was performed without constraints, we need tho show, that the resulting $\lfloor^d D$ and ν are positive. We already showed that ν is positive. Because p is positive, $\lfloor^d D = \frac{\nu}{p}$ is positive too. Because the function G is continuous and has unique local extreme, this extreme is global extreme.

Now we have to do the same work for $W(\hat{\theta}, C)$. The used formulas from matrix differential calculus are summarized in Proposition 8.

$$\frac{\partial W}{\partial \hat{\theta}} = -C^{-1}U + pC^{-1}\hat{\theta} \tag{3.15}$$

$$\frac{\partial W}{\partial C^{-1}} = -0.5C + 0.5Q' - \hat{\theta}U' + 0.5p\hat{\theta}\hat{\theta}' \tag{3.16}$$

$$\frac{\partial^2 W}{\partial \hat{\theta}^2} = pC^{-1} \tag{3.17}$$

$$\frac{\partial^2 W}{\partial C^{-2}} = 0.5C \otimes C \tag{3.18}$$

$$\frac{\partial^2 W}{\partial C^{-1} \partial \hat{\theta}} = I \otimes \left(-U + p\hat{\theta}\right), \tag{3.19}$$

where $\otimes$ denotes Kronecker product of matrices.

After a simple manipulation with first derivatives, we get the unique solution (note that Q is symmetric):

$$\hat{\theta} = \frac{U}{p} \tag{3.20}$$

$$C = Q - 2\hat{\theta}U' + p\hat{\theta}\hat{\theta}' = Q - \frac{UU'}{p}. \tag{3.21}$$

We found the stationary point, we need to prove that it is a minimum. For the stationary point it holds that $\left(-U + p\hat{\theta}\right) = 0$, hence the Hessian matrix

$$H = \begin{pmatrix} \frac{\partial^2 W}{\partial \hat{\theta}^2} & \frac{\partial^2 W}{\partial C^{-1} \partial \hat{\theta}}{}' \\ \frac{\partial^2 W}{\partial C^{-1} \partial \hat{\theta}} & \frac{\partial^2 W}{\partial C^{-2}} \end{pmatrix} = \begin{pmatrix} pC^{-1} & 0 \\ 0 & C \otimes C \end{pmatrix}$$

is positive definite, because Kronecker product of positive definite matrices is positive definite.

As we performed minimization without constraints, we need to prove that the obtained C is positive definite.

$$\begin{aligned} C &= Q - \frac{UU'}{p} \\ Q &= \int \frac{\theta\theta'}{r} f(\theta, r) d\theta dr = p\mathcal{E}\left[\theta\theta'\right]_h \\ U &= \int \frac{\theta}{r} f(\theta, r) d\theta dr = p\mathcal{E}\left[\theta\right]_h \\ C &= p\left(\mathcal{E}\left[\theta\theta'\right]_h - \mathcal{E}\left[\theta'\right]_h \mathcal{E}\left[\theta\right]_h'\right) = p\mathbf{cov}\left[\theta\right]_h \end{aligned}$$

Because p is positive and we assume that $\mathbf{cov}\left[\theta\right]_h$ is positive definite, the obtained C is positive definite. The function W is continuous and has unique local extreme, hence this extreme is global extreme.

□

Proposition 2 (Best projection into Gaussian class) *Let $f(\theta)$ be arbitrary joint pdf on vector θ with a finite positive definite covariance matrix $\mathbf{cov}\left[\theta\right]_f$. Then the statistics (M, R) minimizing the KL divergence*

$$\mathcal{D}\left(f(\theta) \,\middle|\middle|\, \mathcal{N}_\theta\left(M, R\right)\right)$$

fulfill:

$$\begin{aligned} R &= \mathbf{cov}\left[\theta\right]_f \\ M &= \mathcal{E}\left[\theta\right]_f \end{aligned}$$

Proof: The proof is omitted here. It is just simplified version of proof of Proposition 1. □

Chapter 4

Problem Formulation

Within this text, we consider the parameterized model of the system in the form of finite probabilistic mixture with data dependent weights. Here, these mixture models are defined and the main estimation task is formulated.

4.1 Dynamic Probabilistic Mixture

We consider the parameterized model of the system in the following form:

$$f(d_t|\phi_{t-1},\Theta) \equiv \sum_{c=1}^{\mathring{c}} \alpha_c(\phi_{t-1}|\Omega) f_c(d_t|\phi_{t-1},\Theta_c), \mathring{c}<\infty, \quad \text{where} \tag{4.1}$$

$$\mathring{c} \equiv \text{number of components} \tag{4.2}$$

$$f_c(d_t|\phi_{t-1},\Theta_c) \equiv \text{c-th component given by the component parameters } \Theta_c$$

$$\alpha_c(\phi_{t-1}|\Omega) \equiv \text{c-th component weighting function (cwf) given by the parameter } \Omega$$

$$\alpha_c(\phi_{t-1}|\Omega) \geq 0, \quad \sum_{c=1}^{\mathring{c}} \alpha_c(\phi_{t-1}|\Omega) = 1, \quad \forall \phi_{t-1}, \ \forall c \tag{4.3}$$

$$\Theta \equiv \{\Theta_1,\cdots,\Theta_{\mathring{c}},\Omega\} \text{ is unknown parameter} \tag{4.4}$$

Verbally: The dynamic probabilistic mixture is a convex combination of several dynamic pdfs called components. The actual weights depends generally on the state vector ϕ_{t-1}. Mixture parameter Θ is formed by the component parameters $\{\Theta_1,\cdots,\Theta_{\mathring{c}}\}$ and by the parameter Ω determining the behavior of component weighting functions. The parameter Θ represents our only uncertainty about the system model, i.e. we assume the know functional form of the components f_c and component weighting functions α_c. The next simple example illustrates all defined terms.

Example 5 (Dynamic probabilistic mixture)

$$\begin{aligned}
\mathring{d} &\equiv 1 && \textit{(data are scalar)}\\
\mathring{c} &\equiv 2 && \textit{(2 components)}\\
\phi_{t-1} &\equiv (d_{t-1},d_{t-2}) && \textit{(state of the model)}\\
\Omega &\equiv (\lambda_1,\lambda_2) && \textit{(parameter of cwfs)}\\
\Theta &\equiv (\lambda_1,\lambda_2,\Theta_1,\Theta_2) && \textit{(mixture parameter)}
\end{aligned}$$

$$\begin{aligned}
\alpha_1(\phi_{t-1}|\Omega) &\equiv \alpha_1(d_{t-1},d_{t-2}|\lambda_1,\lambda_2) = \frac{\lambda_1^2 d_{t-1}^2}{\lambda_1^2 d_{t-1}^2+\lambda_2^2 d_{t-2}^2} && \textit{(1st cwf)}\\
\alpha_2(\phi_{t-1}|\Omega) &\equiv \alpha_2(d_{t-1},d_{t-2}|\lambda_1,\lambda_2) = \frac{\lambda_2^2 d_{t-2}^2}{\lambda_1^2 d_{t-1}^2+\lambda_2^2 d_{t-2}^2} && \textit{(2nd cwf)}\\
f_1(d_t|\phi_{t-1},\Theta_1) &\equiv f_1(d_t|d_{t-1},d_{t-2},\Theta_1) = \frac{1}{(1+(d_t-\Theta_1 d_{t-1})^2)\times\pi} && \textit{(1st component)}\\
f_2(d_t|\phi_{t-1},\Theta_2) &\equiv f_2(d_t|d_{t-1},d_{t-2},\Theta_2) = \frac{1}{(1+(d_t-\Theta_2 d_{t-2})^2)\times\pi} && \textit{(2nd component)}
\end{aligned}$$

The example presents one dimensional dynamic mixture with dynamic weights. It has two components with Cauchy distribution. Note that sum of cwfs is always 1.

Before fixing and refining nomenclature related to the mixture, we split the individual components into so called factors that provide flexibility of the parametric description. Using the chain rule (Proposition 19), the pdfs $f_c(d_t|\phi_{t-1},\Theta_c)$ can be written as a product of pdfs of individual entries of d_t:

$$f_c(d_t|\phi_{t-1},\Theta_c) = \prod_{i=1}^{\mathring{d}} f_{ic}(d_{i;t}|d_{i+1;t},\cdots,d_{\mathring{d};t},\phi_{t-1},\Theta_{ic}). \tag{4.5}$$

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 (4.5).

Before applying the chain rule, entries of d_t can be permuted and some permutations may lead to parameterizations with less parameters. This motivates inclusion of permutations into the model description. Because each component can generally use another permutation, we have to add an additional parameter to the data index, which will determine the component (permutation). More exactly, let $d_{c;t}$ denote the data record after permutation in c-th component. $d_{ic;t}$ is then i-th entry in this permuted data record. Using this notation, the result of the chain rule reads:

$$f_c(d_t|\phi_{t-1},\Theta_c) = \prod_{i=1}^{\mathring{d}} f_{ic}(d_{ic;t}|d_{(i+1)c;t},\cdots,d_{\mathring{d}c;t},\phi_{t-1},\Theta_{ic}) \equiv \prod_{i=1}^{\mathring{d}} f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic}), \tag{4.6}$$

where the regression vector $\psi_{ic;t}$ is generally a sub-vector of the vector

$$[d_{(i+1)c;t},\cdots,d_{\mathring{d}c;t},\phi_{t-1}]'.$$

Often, it is reasonable to include constant 1 into the regression vector $\psi_{ic;t}$. Hence we define $\psi_{ic;t}$ as a sub-vector of the vector

$$[d_{(i+1)c;t},\cdots,d_{\mathring{d}c;t},\phi_{t-1},1]'. \tag{4.7}$$

The next example demonstrates two ways of splitting components into factors.

Example 6 (Parameterized factor)

$$\begin{aligned}
\mathring{d} &\equiv 2 && \textit{(2-dimensional data} \Rightarrow \textit{we have 2 permutations)}\\
\Theta_1 &\equiv (\mu,\rho),\ \rho\in(-1,1) && \textit{(we are dealing with component 1)}\\
\phi_{t-1} &\equiv \emptyset && \textit{(for simplicity, we suppose no dependence on past)}
\end{aligned}$$

$$f_1(d_t|\phi_{t-1},\Theta_1) \equiv \mathcal{N}_{d_t}\left(\begin{pmatrix}0\\ \mu\end{pmatrix},\begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

First Permutation

$$\begin{aligned}
\psi_{11;t}\equiv d_{2;t},\ \Theta_{11} &\equiv (\mu,\rho),\ \psi_{21;t}\equiv\emptyset,\ \Theta_{21}\equiv\mu\\
f_1(d_t|\phi_{t-1},\Theta_1) &\equiv \underbrace{\mathcal{N}_{d_{1;t}}\left(\rho(d_{2;t}-\mu),1-\rho^2\right)}_{f_{11}(d_{11;t}|\psi_{11;t},\Theta_{11})}\ \underbrace{\mathcal{N}_{d_{2;t}}(\mu,1)}_{f_{21}(d_{21;t}|\psi_{21;t},\Theta_{21})}
\end{aligned}$$

Second Permutation

$$\begin{aligned}
\psi_{11;t}\equiv d_{1;t},\ \Theta_{11} &\equiv (\mu,\rho),\ \psi_{21;t}\equiv\emptyset,\ \Theta_{21}\equiv\emptyset\\
f_1(d_t|\phi_{t-1},\Theta_1) &\equiv \underbrace{\mathcal{N}_{d_{2;t}}\left(\mu+\rho d_{1;t},1-\rho^2\right)}_{f_{11}(d_{11;t}|\psi_{11;t},\Theta_{11})}\ \underbrace{\mathcal{N}_{d_{1;t}}(0,1)}_{f_{21}(d_{21;t}|\psi_{21;t},\Theta_{21})}
\end{aligned}$$

The example presents two possible ways of splitting two-dimensional normal pdf into normal factors. Note that the second way of splitting results into empty Θ_{21} whereas the first splitting results into nonempty Θ_{21}. This shows that it makes sense to distinguish the particular permutations.

According to the previous definitions, the parameterized factor $f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic})$ is determined by its parameter Θ_{ic}, by the index of the channel it acts on and by the way how the regression vector $\psi_{ic;t}$ is constructed from $d(t)$.

Now let us summarize the nomenclature related to the mixtures.

Agreement 4 (Nomenclature related to mixtures review)

$\mathring{c}$ *is called number of components.*

$f_c(d_t|\phi_{t-1},\Theta_c)$ *is called parameterized component.*

$\alpha_c(\phi_{t-1}|\Omega)$ *is the component weighting function (cwf) of the c-th parameterized component.*

$f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic})$ *is called parameterized factor.*

$\psi_{ic;t}$ *is regression vector.*

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

4.2 Form of the Prior and the Posterior Pdfs

According to the general rules in Section 2.3, we need to choose the prior and posterior pdf in a form that is well manipulable. This motivates us to select this general form:

Agreement 5 (Considered forms of pdfs on Θ^*) *The prior* $\pi(\Theta)\equiv\pi(\Theta|\mathcal{G}_0)$ *and the posterior* $\pi(\Theta|d(t))\equiv\pi(\Theta|\mathcal{G}_t)$ *are considered to be of the common form:*

$$\begin{aligned}\pi(\Theta|\mathcal{G}_t) &\equiv \rho(\Omega|\mathcal{H}_t)\prod_{i,c=1}^{\mathring{d},\mathring{c}}\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t}),\ t\in\{0\}\cup t^*,\ \textit{where}\\ \rho(\Omega|\mathcal{H}_t) &\quad \textit{is pdf on cwf parameter } \Omega \textit{ determined}\\ &\quad \textit{by the finite-dimensional statistic } \mathcal{H}_t\\ \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t}) &\quad \textit{are pdfs on factor parameters } \Theta_{ic;t} \textit{ determined}\\ &\quad \textit{by the finite-dimensional statistics } \mathcal{S}_{ic;t}\\ \mathcal{G}_t &\equiv (\mathcal{H}_t,\mathcal{S}_{\bullet\bullet;t}).\end{aligned} \tag{4.8}$$

Verbally, parameters $\Theta_{ic},\ i\equiv\{1,\cdots,\mathring{d}\},\ c\in\{1,\cdots,\mathring{c}\}$, *of the individual parameterized factors are considered to be conditionally independent, and also, independent of the parameter* Ω *of component weighting functions. The posterior statistic* $\mathcal{G}_t$ *is formed by the statistic* $\mathcal{H}_t$ *determining the pdf of the parameter of cwfs and by the statistics* $\{\mathcal{S}_{ic;t}\}_{i=1,c=1}^{\mathring{d},\mathring{c}}$ *determining the pdf of parameters of particular factors.*

Remarks 1

1. *The independence of the factor parameters is restrictive, but it is the only way to cope with the high dimensional cases.*
2. *When the conjugate pdf to the particular factor* $f_{ic}(d_i|\psi_{ic;t},\Theta_{ic})$ *exists, it is of course reasonable to select the pdf* $\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})$ *as conjugate one.*

Example 7 (Form of the prior and posterior pdf) *The posterior pdf of the mixture model from Example 5 could look as follows:*

$$\begin{aligned}
\rho(\Omega|\mathcal{H}_t) &\equiv \rho(\lambda_1,\lambda_2|M_t,R_t) = \mathcal{N}_{(\lambda_1,\lambda_2)'}(M_t,R_t)\\
\pi_{11}(\Theta_{11}|\mathcal{S}_{11;t}) &\equiv \pi_{11}(\Theta_1|m_t) = \mathcal{N}_{\Theta_1}(m_t,1)\\
\pi_{12}(\Theta_{12}|\mathcal{S}_{12;t}) &\equiv \pi_{12}(\Theta_2|\mu_t) = \mathcal{N}_{\Theta_2}(\mu_t,1)\\
\mathcal{H}_t &\equiv (M_t,R_t),\ \mathcal{S}_{11;t}\equiv m_t,\ \mathcal{S}_{12;t}\equiv\mu_t\\
\mathcal{G}_t &\equiv (M_t,R_t,m_t,\mu_t)\\
\pi(\Theta|\mathcal{G}_t) &\equiv \pi(\lambda_1,\lambda_2,\Theta_1,\Theta_2|M_t,R_t,m_t,\mu_t) = \mathcal{N}_{[\lambda_1,\lambda_2]'}(M_t,R_t)\mathcal{N}_{\Theta_1}(m_t,1)\mathcal{N}_{\Theta_2}(\mu_t,1)
\end{aligned}$$

4.3 Addressed Problem

Now, it is time to exactly define the addressed problem. We apply the approximation from Section 2.3 to the introduced mixture model (4.1) and get the following problem:

Find the statistic $\mathcal{G}_t$, which minimizes KL divergence $\mathcal{D}\left(\hat{\pi}_t(\Theta)\ \middle|\middle|\ \pi(\Theta|\mathcal{G}_t)\right)$, where

$$\begin{aligned}
\hat{\pi}_t(\Theta) &\equiv \frac{f(d_t|\phi_{t-1},\Theta)\pi(\Theta|\mathcal{G}_{t-1})}{\int f(d_t|\phi_{t-1},\Theta)\pi(\Theta|\mathcal{G}_{t-1})d\Theta} \qquad (4.9)\\
\pi(\Theta|\mathcal{G}_{t-1}) &\equiv \rho(\Omega|\mathcal{H}_{t-1})\prod_{i=1,c=1}^{\mathring{d},\mathring{c}}\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1})\\
f(d_t|\phi_{t-1},\Theta) &\equiv \sum_{c=1}^{\mathring{c}}\alpha_c(\phi_{t-1}|\Omega)\prod_{i=1}^{\mathring{d}}f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic}).
\end{aligned}$$

In other words, we are looking for $\mathcal{G}_t \equiv (\mathcal{H}_t, \mathcal{S}_{\bullet\bullet;t})$ knowing $\mathcal{G}_{t-1} \equiv (\mathcal{H}_{t-1}, \mathcal{S}_{\bullet\bullet;t-1})$ and d_t, ϕ_{t-1}. This optimization task is solved in next chapter.

Chapter 5

General Solution

In this chapter, we will solve the problem formulated in Section 4.3 as generally as possible. First let us investigate the form of correct update $\hat{\pi}_t(\Theta)$ defined in (4.9).

5.1 Form of Correct Update

Proposition 3 (Form of correct update) *The correct update $\hat{\pi}_t(\Theta)$ defined by (4.9) for the mixture model (Section 4.1) has the following form:*

$$\hat{\pi}_t(\Theta) = \sum_{c=1}^{\mathring{c}} w_{c;t}\ \rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \prod_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1}) \prod_{j=1}^{\mathring{d}} \pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U), \tag{5.1}$$

where the following constituents are used:

$$\textit{data weight} \quad w_{c;t} \equiv \frac{\hat{\alpha}_{c;t-1}\beta_{c;t}}{\sum_{c=1}^{\mathring{c}} \hat{\alpha}_{c;t-1}\beta_{c;t}} \tag{5.2}$$

$$\textit{weight estimate} \quad \hat{\alpha}_{c;t-1} \equiv \int \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega \tag{5.3}$$

$$\textit{component prediction} \quad \beta_{c;t} \equiv \prod_{i=1}^{\mathring{d}} \mathcal{I}_{ic;t} \tag{5.4}$$

$$\textit{factor prediction} \quad \mathcal{I}_{ic;t} \equiv \int f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic})\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1})d\Theta_{ic} \tag{5.5}$$

$$\textit{cwf update} \quad \rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \equiv \frac{\alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})}{\hat{\alpha}_{c;t-1}} \tag{5.6}$$

$$\textit{factor update} \quad \pi_{ic}^U(\Theta_{ic}|\mathcal{S}_{ic;t}^U) \equiv \frac{f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic})\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1})}{\mathcal{I}_{ic;t}} \tag{5.7}$$

Proof:

$$\begin{aligned}
& f(d_t|\phi_{t-1},\Theta)\pi(\Theta|\mathcal{G}_{t-1}) = \\
= & \left(\sum_{c=1}^{\mathring{c}} \alpha_c(\phi_{t-1}|\Omega) \prod_{j=1}^{\mathring{d}} f_{jc}(d_{jc;t}|\psi_{jc;t},\Theta_{jc})\right) \times \left(\rho(\Omega|\mathcal{H}_{t-1}) \prod_{i=1,r=1}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1})\right) = \\
= & \sum_{c=1}^{\mathring{c}} \left(\underbrace{\alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})}_{\hat{\alpha}_{c;t-1}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U)} \times \prod_{j=1}^{\mathring{d}} \underbrace{\pi_{jc}(\Theta_{jc}|\mathcal{S}_{jc;t-1})f_{jc}(d_{jc;t}|\psi_{jc;t},\Theta_{jc})}_{\mathcal{I}_{jc;t}\pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U)} \prod_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1}) \right)
\end{aligned}$$

$$= \sum_{c=1}^{\mathring{c}} \hat{\alpha}_{c;t-1}\beta_{c;t} \underbrace{\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \prod_{j=1}^{\mathring{d}} \pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U) \prod_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1})}_{\text{This part is pdf, hence it integrates to 1.}}$$

It is clear that the normalizing integral $\int f(d_t|\phi_{t-1},\Theta)\pi(\Theta|\mathcal{G}_{t-1})d\Theta = \sum_{c=1}^{\mathring{c}} \hat{\alpha}_{c;t-1}\beta_{c;t}$, hence

$$\hat{\pi}(\Theta) = \sum_{c=1}^{\mathring{c}} \underbrace{\frac{\hat{\alpha}_{c;t-1}\beta_{c;t}}{\sum_{\tilde{c}=1}^{\mathring{c}} \hat{\alpha}_{\tilde{c};t-1}\beta_{\tilde{c};t}}}_{\equiv\, w_{c;t}} \rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \prod_{j=1}^{\mathring{d}} \pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U) \prod_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1}).$$

□

Remarks 2 *It is obvious that if $w_{\bullet;t}$ has only one nonzero element, the form of the correct update (5.1) is the same as the form of old posterior density $\pi(\Theta|\mathcal{G}_{t-1})$ (4.8). This means that in this case no approximation is needed and new posterior density $\pi(\Theta|\mathcal{G}_t)$ equals to the correct Bayesian update $\hat{\pi}_t(\Theta)$ (5.1).*

5.2 General Minimization

Proposition 4 (Minimization of the KL divergence) *For $\mathcal{G}_t \equiv \{\mathcal{S}_{\bullet\bullet;t},\ \mathcal{H}_t\}$ minimizing*

$$\mathcal{D}\left(\hat{\pi}_t(\Theta)\ \middle|\middle|\ \pi(\Theta|\mathcal{G}_t)\right), \text{ it holds:}$$

$$\begin{aligned} \mathcal{H}_t &\in \operatorname{Arg\,min}_{\mathcal{H}_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U)\ \middle|\middle|\ \rho(\Omega|\mathcal{H}_t)\right) \\ \mathcal{S}_{ic;t} &\in \operatorname{Arg\,min}_{\mathcal{S}_{ic;t}} \mathcal{D}\left((1-w_{c;t})\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1}) + w_{c;t}\pi_{ic}^U(\Theta_{ic}|\mathcal{S}_{ic;t}^U)\ \middle|\middle|\ \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})\right). \end{aligned} \tag{5.8}$$

Proof:

Instead of working with KL divergence, we will evaluate the Kerridge divergence $\mathcal{K}\left(\hat{\pi}_t(\Theta)\ \middle|\middle|\ \pi(\Theta|\mathcal{G}_t)\right)$. Details about this divergence, its properties and its relation to the KL divergence are discussed in Section C.1.2.

$$\begin{aligned} &\mathcal{K}\left(\sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \prod_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1}) \prod_{j=1}^{\mathring{d}} \pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U)\ \middle|\middle|\ \pi(\Theta|\mathcal{G}_t)\right) \overset{Proposition\ 24}{=} \\ = &\sum_{c=1}^{\mathring{c}} w_{c;t}\mathcal{K}\left(\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \prod_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1}) \prod_{j=1}^{\mathring{d}} \pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U)\ \middle|\middle|\ \pi(\Theta|\mathcal{G}_t)\right) \overset{Proposition\ 26}{=} \\ = &\sum_{c=1}^{\mathring{c}} w_{c;t}\left[\mathcal{K}\left(\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U)\ \middle|\middle|\ \rho(\Omega|\mathcal{H}_t)\right) + \sum_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \mathcal{K}\left(\pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t-1})\ \middle|\middle|\ \pi_{ir}(\Theta_{ir}|\mathcal{S}_{ir;t})\right) + \right. \\ &\left. + \sum_{j=1}^{\mathring{d}} \mathcal{K}\left(\pi_{jc}^U(\Theta_{jc}|\mathcal{S}_{jc;t}^U)\ \middle|\middle|\ \pi_{jc}(\Theta_{jc}|\mathcal{S}_{jc;t})\right)\right]. \end{aligned}$$

Let us temporarily denote

$$\begin{aligned}\mathcal{K}_{ic} &= \mathcal{K}\left(\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1}) \,\middle\|\, \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})\right)\\ \mathcal{K}^U_{jc} &= \mathcal{K}\left(\pi^U_{jc}(\Theta_{jc}|\mathcal{S}^U_{jc;t-1}) \,\middle\|\, \pi_{jc}(\Theta_{jc}|\mathcal{S}_{jc;t})\right).\end{aligned}$$

The minimized function gets the form

$$\sum_{c=1}^{\mathring{c}} w_{c;t}\mathcal{K}\left(\rho^U_c(\Omega|\mathcal{H}^U_{c;t-1}) \,\middle\|\, \rho(\Omega|\mathcal{H}_t)\right) + \sum_{c=1}^{\mathring{c}} w_{c;t} \sum_{\substack{i,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \mathcal{K}_{ir} + \sum_{j,c=1}^{\mathring{d},\mathring{c}} w_{c;t}\mathcal{K}^U_{jc} \overset{Proposition\ 15}{=}$$

$$= \sum_{c=1}^{\mathring{c}} w_{c;t}\mathcal{K}\left(\rho^U_c(\Omega|\mathcal{H}^U_{c;t-1}) \,\middle\|\, \rho(\Omega|\mathcal{H}_t)\right) + \sum_{i,c=1}^{\mathring{d},\mathring{c}} \left[w_{c;t}\mathcal{K}^U_{ic} + (1-w_{c;t})\mathcal{K}_{ic}\right].$$

Now it is clear that minimization of this expression can be done separately.

$$\begin{aligned}\mathcal{H}_t &\in \operatorname{Arg\,min}_{\mathcal{H}_t}\left[\sum_{c=1}^{\mathring{c}} w_{c;t}\mathcal{K}\left(\rho^U_c(\Omega|\mathcal{H}^U_{c;t-1}) \,\middle\|\, \rho(\Omega|\mathcal{H}_t)\right)\right]\\ \mathcal{S}_{ic;t} &\in \operatorname{Arg\,min}_{\mathcal{S}_{ic;t}}\left[(1-w_{c;t})\mathcal{K}_{ic} + w_{c;t}\mathcal{K}^U_{ic}\right] =\\ &= \operatorname{Arg\,min}_{\mathcal{S}_{ic;t}}\Big[(1-w_{c;t})\mathcal{K}\left(\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1}) \,\middle\|\, \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})\right) +\\ &\qquad + w_{c;t}\mathcal{K}\left(\pi^U_{ic}(\Theta_{ic}|\mathcal{S}^U_{ic;t}) \,\middle\|\, \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})\right)\Big]\end{aligned}$$

Now, after applying Propositions 24 and 23, we obtain directly the statement of the proposition.

□

5.3 General Algorithm

Proposition 4 splits the overall problem into two subproblems. The first subproblem is obtaining the statistic $\mathcal{H}_t$ determining the posterior pdf of the parameter Ω of cwfs. The second subproblem is evaluation of statistics $\{\mathcal{S}_{ic;t}\}_{i,c=1}^{\mathring{d},\mathring{c}}$ determining the posterior pdf on parameters Θ_{ic} of particular factors. Important result is that the minimization can be done factor-wise, which simplifies substantially the optimization. The two mentioned subproblems are connected through evaluation of weights $w_{c;t}$, which are needed in both subproblems.

Now we will specify the tasks, which must be done for particular factors and cwfs types.

For all factors

- evaluate factor predictions $\mathcal{I}_{ic;t}$ (5.5)
- evaluate factor updates $\pi^U_{ic}(\Theta_{ic}|\mathcal{S}^U_{ic;t})$ (5.7)
- perform the minimization

$$\mathcal{S}_{ic;t} \in \operatorname{Arg\,min}_{\mathcal{S}_{ic;t}}\left[\mathcal{D}\left((1-w_{c;t})\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1}) + w_{c;t}\pi^U_{ic}(\Theta_{ic}|\mathcal{S}^U_{ic;t}) \,\middle\|\, \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})\right)\right]$$

For all cwfs

- evaluate weight estimates $\hat{\alpha}_{c;t-1}$ (5.3)
- evaluate cwf updates $\rho^U_c(\Omega|\mathcal{H}^U_{c;t-1})$ (5.6)

- perform the minimization
  $\mathcal{H}_t \in \mathrm{Arg\,min}_{\mathcal{H}_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \;\middle\|\; \rho(\Omega|\mathcal{H}_t)\right)$

When we are able to perform all the mentioned steps, we can perform one step of the projection based estimation according to Algorithm 1. Before specifying this algorithm, let us summarize some rules of writing algorithms bellow.

Agreement 6

- *Each algorithm has unique name.*
- *Each algorithm begins with specification of its name, input and output parameters.*
- *Algorithm can contain "calling" of other algorithms, using their name and lists of parameters. Neither the order of inputs nor outputs parameters is significant. The meaning should be clear from the variables names.*
- *In all algorithms, we expect that the state vectors and regression vectors are known. Hence they will not be specified as inputs of algorithms.*
- *In all algorithms, we expect that the functional forms of the parameterized model and posterior pdf are known. Hence they will not be specified as inputs of algorithms.*

Algorithm 1 (General update) $(\mathcal{H}_t,\ \mathcal{S}_{\bullet\bullet;t}) = \mathrm{MIXUPDT}\big(\mathcal{H}_{t-1},\ \mathcal{S}_{\bullet\bullet;t-1}\big)$

1. *For each factor ic, evaluate the factor prediction*
   $\mathcal{I}_{ic;t} = \int f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic})\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1})d\Theta_{ic}$
2. *For each component c, evaluate the weight estimate* $\hat{\alpha}_{c;t-1} = \int \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega$
3. *For each component c, evaluate the data weight* $w_{c;t} = \frac{\hat{\alpha}_{c;t-1}\prod\mathcal{I}_{ic;t}}{\sum\hat{\alpha}_{c;t-1}\prod\mathcal{I}_{ic;t}}$
4. *For each factor ic, evaluate the factor update* $\pi_{ic}^U(\Theta_{ic}|\mathcal{S}_{ic;t}^U) = \frac{\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1})f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic})}{\mathcal{I}_{ic}}$
5. *For each factor ic, evaluate the updated factor statistic*
   $\mathcal{S}_{ic;t} \in \mathrm{Arg\,min}_{\mathcal{S}_{ic;t}}\left[\mathcal{D}\left((1-w_{c;t})\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1}) + w_{c;t}\pi_{ic}^U(\Theta_{ic}|\mathcal{S}_{ic;t}^U) \;\middle\|\; \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})\right)\right]$
6. *For each component c, evaluate the cwf update* $\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) = \frac{\rho(\Omega|\mathcal{H}_{t-1})\alpha_t(\phi_{t-1}|\Omega)}{\hat{\alpha}_{c;t-1}}$
7. *Evaluate the updated cwf statistic* $\mathcal{H}_t \in \mathrm{Arg\,min}_{\mathcal{H}_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \;\middle\|\; \rho(\Omega|\mathcal{H}_t)\right)$

It can be simply seen that the order of steps 1 and 2 can be arbitrary. The steps can be even performed simultaneously. Similarly, the steps 4, 5 can be performed simultaneously with steps 6, 7.

Evaluating of the data weights $w_{c;t}$ in the way specified in the previous algorithm would cause numerical problems, because $\mathcal{I}_{ic;t}$ can be very very small numbers. We need to work with logarithms of them. Let us denote $\mathcal{L}_{ic;t} \equiv \ln(\mathcal{I}_{ic;t})$, $\mathcal{Z}_{c;t-1} \equiv \ln(\hat{\alpha}_{c;t-1})$. Now we will rewrite Algorithm 1 using the mentioned logarithms. Simultaneously, we will replace some steps with "calling" of algorithms, which were not defined yet. This can be taken as a "forward declaration of algorithm" and it specifies the work, which should be done in next chapters.

Algorithm 2 (General update) $(\mathcal{H}_t,\ \mathcal{S}_{\bullet\bullet;t}) = \mathrm{MIXUPDT}\big(\mathcal{H}_{t-1},\ \mathcal{S}_{\bullet\bullet;t-1}\big)$

1. *For each factor ic, evaluate* $\mathcal{L}_{ic;t} = FACNORM(\mathcal{S}_{ic;t})$

2. *Evaluate* $\mathcal{Z}_{\bullet;t-1} = WEIGHTNORM(\mathcal{H}_{t-1})$

3. *Evaluate* $w_{\bullet;t} = \text{EVAL\_WEIGHT}(\mathcal{L}_{\bullet\bullet;t}, \mathcal{Z}_{\bullet;t-1})$

4. *For each factor ic, evaluate the statistic* $\mathcal{S}_{ic;t} = FACUPDT(\mathcal{S}_{ic;t-1}, w_c)$

5. *Evaluate* $\mathcal{H}_t = WEIGHTUPDT(\mathcal{H}_{t-1}, w_\bullet)$

Remarks 3 *The steps 4, 5 (6, 7) of Algorithm 1 were replaced with single step 4(5) in Algorithm 2, because sometimes it is unnecessary to evaluate* $\mathcal{S}^U_{ic;t}$ *explicitly.*

Within Algorithm 2, we formalized all tasks, which have to be solved in the next work. The algorithms *FACNORM, WEIGHTNORM, FACUPDT, WEIGHTUPDT* depend, of course, on the functional form of parameterized factors and cwfs. Hence for each considered variant of factor, we need variant of algorithms *FACNORM* and *FACUPDT* and for each variant of cwf we need variant of algorithms *WEIGHTNORM* and *WEIGHTUPDT*. Important variants of factors and cwfs are proposed in subsequent chapters.

The algorithm EVAL_WEIGHT can be simply written at this general level.

Algorithm 3 (Evaluation of data weight) $(w_{\bullet;t}) = \text{EVAL\_WEIGHT}\left(\mathcal{L}_{\bullet\bullet;t}, \mathcal{Z}_{\bullet;t-1}\right)$

1. *For each component c evaluate* $Q_{c;t} = \mathcal{Z}_{c;t-1} + \sum_{i=1}^{\mathring{d}} \mathcal{L}_{ic;t}$
2. $\bar{Q}_{\bullet;t} \equiv Q_{\bullet;t} - \max Q_{\bullet;t}$
3. $w_{\bullet;t} = \frac{\exp(\bar{Q}_{\bullet;t})}{\sum_c \exp(\bar{Q}_{\bullet;t})}$

Proposition 5 (Correctness of algorithm 3) *Algorithm 3 is correct.*
Proof:

$$\begin{aligned} w_{c;t} &= \frac{\exp\left(Q_{c;t} - \max Q_{\bullet;t}\right)}{\sum_c \left(\exp\left(Q_{c;t} - \max Q_{\bullet;t}\right)\right)} = \frac{-\exp\left(Q_{c;t}\right)\exp\left(\max Q_{\bullet;t}\right)}{-\exp\left(\max Q_{\bullet;t}\right)\sum_c \exp\left(Q_{c;t}\right)} = \\ &= \frac{\exp\left(Q_{c;t}\right)}{\sum_c \exp\left(Q_{c;t}\right)} = \frac{\hat{\alpha}_{c;t-1}\prod \mathcal{I}_{ic;t}}{\sum_c \hat{\alpha}_{c;t-1}\prod \mathcal{I}_{ic;t}}. \end{aligned}$$

□

Chapter 6

Optimization of Statistics for Normal Factors

In this chapter, we will solve the factor-related problems sketched in Section 5.3 for dynamic normal models with unknown and known variance. The outcome is design of algorithms *FACNORM* and *FACUPDT* used in Algorithm 2 for each factor type.

Because this chapter deals with only one factor $f_{ic}(d_{ic;t}|\psi_{ic;t}, \Theta_{ic})$ and with corresponding part of posterior pdf $\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t})$, we can omit the indexes i and c, i.e. $f_{ic}(d_{ic;t}|\psi_{ic;t}, \Theta_{ic}) \rightarrow f(d_t|\psi_t, \Theta)$, $\pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t}) \rightarrow \pi(\Theta|\mathcal{S}_t)$.

6.1 Normal Factors with Unknown Variance

In this section, we assume that the parameterized factor is dynamic normal pdf with parameters $\Theta \equiv (\theta, r)$, where θ is vector of regression coefficients and r is noise variance of the factor.

$$f(d_t|\psi_t, \Theta) = \mathcal{N}_{d_t}(\theta'\psi_t, r) = \frac{1}{\sqrt{2\pi r}} \exp\left(-\frac{(d_t - \theta'\psi_t)^2}{2r}\right) \tag{6.1}$$

We do not need to introduce a shift in the mean value, because the regression vector can contain entry equal to 1 (see (4.7)). The shifting constant is then placed to the corresponding place of the vector of regression coefficients. Details about Bayesian estimation of normal factors can be found in Appendix D.

Example 8 (Normal factor)

ψ_t	$\equiv$	(1)	*(regression vector)*
Θ	$\equiv$	(θ, r)	*(unknown factor parameter consists of two scalars)*
$f(d_t\|\psi_t, \Theta)$	$\equiv$	$\mathcal{N}_{d_t}(\theta, r)$	*(normal static factor)*

The factor is one-dimensional pdf, which can be simply plot when its parameters are known. Figure 6.1 shows this pdf for $\theta = 2$ and $r = 2$.

6.1.1 Form of the Posterior Pdf

The parameterized factor (6.1) has conjugated prior pdf [10]. Hence it is advantageous to use this pdf, when specifying the form of the posterior pdf. (See Remarks 1.) The mentioned conjugated pdf to this model is the Gauss inverse Wishart pdf with statistics $\mathcal{S}_t \equiv (\nu_t, V_t)$ [10], where ν_t is scalar number of degrees of freedom and V_t is so called extended information matrix (square, symmetric, positive definite matrix with $\mathring{\Psi}_t$ rows).

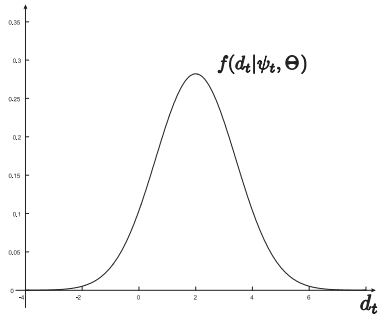


Figure 6.1: Normal factor with known parameters
The figure shows pdf $\mathcal{N}_{d_t}(\theta, r)$ *for known parameters* $\theta = 2,\ r = 2$.

$$\pi(\Theta|\mathcal{S}_t) = GiW_{\theta,r}(V_t, \nu_t) \propto r^{-0.5(\nu_t+\mathring{\psi}_t+2)} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(V_t[-1,\theta']'[-1,\theta']\right)\right\}$$

Example 9 (GiW factor) *The posterior pdf related to the factor specified in Example 8 would be:*

$$\begin{array}{rcll} \Theta & \equiv & (\theta, r) & \textit{(unknown factor parameter consists of two scalars)} \\ \mathcal{S}_t & \equiv & (\nu_t, V_t) & \textit{(statistics of the posterior pdf, scalar and 2x2 matrix)} \\ \pi(\Theta|\mathcal{S}_t) & = & GiW_{\theta,r}(V_t, \nu_t) & \textit{(GiW posterior)} \end{array}$$

Because the factor parameter θ *was scalar in this case, we can plot the pdf* $GiW_{\theta,r}(V_t, \nu_t)$ *for given statistics. Figure 6.2 displays this pdf for some given statistics.*

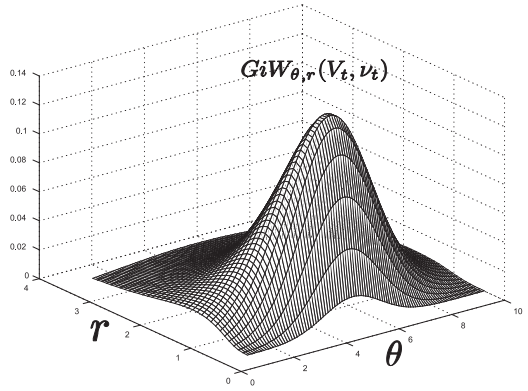


Figure 6.2: GiW factor with known statistics
The figure shows pdf $GiW_{\theta,r}(V_t, \nu_t)$ *for known statistics* $\nu_t = 6$ *and* $V_t = \begin{pmatrix} 16.3333 & 1.6667 \\ 1.6667 & 0.3333 \end{pmatrix}$.

The details and important properties of this pdf are summarized in Appendix D. Note that the matrix V_t can be equivalently manipulated through its $L'DL$ decomposition. i.e. with lower triangular matrix L_t with unit diagonal and positive diagonal matrix D_t, which fulfills the relation $V_t = L_t' D_t L_t$. Next, the matrices L_t and D_t can be equivalently expressed via positive definite matrix C_t, vector $\hat{\theta}_t$ and scalar $\lfloor^d D_t$. This representation determines well-known least squares (LS) statistics. ($\hat{\theta}_t \equiv$ LS estimate of θ, $\lfloor^d D_t \equiv$ LS remainder, $\frac{\lfloor^d D_t}{\nu_t - 2} C_t \equiv$ covariance of LS estimates).The relations between individual representations can be found in Section C.5.2.

Agreement 7 *Because all three representations described above are equivalent, we will not formally distinguish them. If V_t is a statistic of GiW factor, the variables $L_t, D_t, \hat{\theta}_t, C_t, {}^{\lfloor d}D_t$ automatically mean the parts of corresponding representation of the matrix V_t.*

Example 10 (Different representations of matrix V)

The matrix $V_t = \begin{pmatrix} 16.3333 & 1.6667 \\ 1.6667 & 0.3333 \end{pmatrix}$ *from Example 9 has following alternative representations:*

$$ {}^{\lfloor d}D_t = 8,\ \hat{\theta}_t = 5,\ C_t = 3$$

or

$$L_t = \begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix},\ D_t = \begin{pmatrix} 8 & 0 \\ 0 & \frac{1}{3} \end{pmatrix}.$$

6.1.2 Factor Prediction

The factor prediction $\mathcal{I}_t$ (5.5) is defined as

$$\mathcal{I}_t = \int f(d_t|\psi_t, \Theta)\pi(\Theta|\mathcal{S}_{t-1})d\Theta = \int \mathcal{N}_{d_t}(\theta'\psi_t, r)GiW_{\theta,r}(V_{t-1}, \nu_{t-1})d\theta dr.$$

According to Proposition 38, for normal factors and conjugate prior $\mathcal{I}_t$ is evaluated as:

$$\mathcal{I}_t = \frac{\Gamma(0.5(\nu_{t-1}+1))\left[{}^{\lfloor d}D_{t-1}(1+\zeta_t)\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu_{t-1})\left(1+\frac{\hat{e}_t^2}{{}^{\lfloor d}D_{t-1}(1+\zeta_t)}\right)^{0.5(\nu_{t-1}+1)}},\ \text{where} \tag{6.2}$$

$$\begin{aligned} \hat{e}_t &\equiv d_t - \hat{\theta}'_{t-1}\psi_t \equiv \text{prediction error} \\ \zeta_t &\equiv \psi'_t C_{t-1}\psi_t \end{aligned}$$

Remarks 4 *We need to evaluate $\mathcal{L}_t = \ln \mathcal{I}_t$. It can be done efficiently using the product form of (6.2). The following algorithm summarizes this task.*

Algorithm 4 (Factor prediction) $(\mathcal{L}_t) = \text{FACNORM}\left(C_{t-1}, \hat{\theta}_{t-1}, {}^{\lfloor d}D_{t-1}, \nu_{t-1}\right)$

1. *Evaluate* $\zeta_t = \psi'_t C_{t-1}\psi_t$
2. *Evaluate* $\hat{e}_t \equiv d_t - \hat{\theta}'_{t-1}\psi_t$
3. *Evaluate*

$$\begin{aligned} \mathcal{L}_t = \ln \mathcal{I}_t &= \ln\left(\Gamma\left(0.5(\nu_{t-1}+1)\right)\right) - \ln\left(\Gamma\left(0.5\nu_{t-1}\right)\right) - 0.5\ln\left({}^{\lfloor d}D_{t-1}\right) - 0.5\ln\left(1+\zeta_t\right) - \\ &\quad -0.5(\nu_{t-1}+1)\ln\left(1+\frac{\hat{e}_t^2}{{}^{\lfloor d}D_{t-1}(1+\zeta_t)}\right) - 0.5\ln(\pi) \end{aligned}$$

Remarks 5 *Function $\ln(\Gamma(x))$ can be efficiently evaluated without computing $\Gamma(x)$ first [41].*

Example 11 (Factor prediction)

$$\begin{aligned} \psi_t &\equiv (1) && \textit{(regression vector)} \\ \Theta &\equiv (\theta, r) && \textit{(unknown factor parameter consists of two scalars)} \\ f(d_t|\psi_t, \Theta) &\equiv \mathcal{N}_{d_t}(\theta, r) && \textit{(normal static factor)} \\ \pi(\Theta|\mathcal{S}_{t-1}) &= GiW_{\theta,r}(V_{t-1}, \nu_{t-1}) && \textit{(GiW posterior)} \end{aligned}$$

The Figure 6.3 displays $\mathcal{I}_t$ taken as a function of d_t for given values of statistics V_{t-1} and ν_{t-1}.

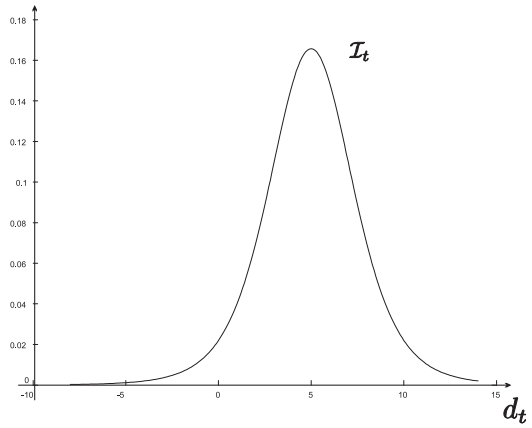


Figure 6.3: Factor prediction as a function of d_t
The figure shows $\mathcal{I}_t$ taken as a function od d_t for $\nu_{t-1} = 6$ and $V_{t-1} = \begin{pmatrix} 16.3333 & 1.6667 \\ 1.6667 & 0.3333 \end{pmatrix}$.

6.1.3 Factor Update

According to Proposition 35, $\mathcal{S}_t^U \equiv [V_t^U, \nu_t^U]$ can be evaluated in the following way:

$$\begin{aligned} V_t^U &= V_{t-1} + \Psi_t \Psi_t' \\ \nu_t^U &= \nu_{t-1} + 1. \end{aligned} \tag{6.3}$$

Using Proposition 33, the relation (6.3) can be rewritten into the $C, \hat{\theta}, {}^{\lfloor d}D$ representation in the following way :

$$\begin{aligned} C_t^U &= C_{t-1} + h_c z_t z_t', \qquad \hat{\theta}_t^U = \hat{\theta}_{t-1} + h_\theta z_t, \qquad {}^{\lfloor d}D_t^U = {}^{\lfloor d}D_{t-1} + \frac{\hat{e}_t^2}{1+\zeta_t} \\ z_t &\equiv C_{t-1}\psi_t, \qquad h_c \equiv -\frac{1}{1+\zeta_t}, \qquad h_\theta \equiv \frac{\hat{e}_t}{1+\zeta_t} \end{aligned}$$

Example 12 (Factor update)

$$\begin{aligned} \psi_t &\equiv (1) && \textit{(regression vector)} \\ \Theta &\equiv (\theta, r) && \textit{(unknown parameter)} \\ f(d_t|\psi_t, \Theta) &\equiv \mathcal{N}_{d_t}(\theta, r) && \textit{(normal static factor)} \\ \pi(\Theta|\mathcal{S}_{t-1}) &\equiv GiW_{\theta,r}(V_{t-1}, \nu_{t-1}) && \textit{(GiW posterior pdf)} \\ \pi^U(\Theta|\mathcal{S}_t^U) &= GiW_{\theta,r}(V_t^U, \nu_t^U) && \textit{(updated GiW posterior pdf)} \end{aligned}$$

The table 6.1 shows statistics of the involved pdfs and some other mentioned auxiliary values for two cases.

6.1.4 Optimization of Statistics

We will use Proposition 1. First we have to check if our case fulfills its assumptions. The pdf f from Proposition 1 has the form

$$f(\theta, r) = (1-w)GiW_{\theta,r}(C_{t-1}, \hat{\theta}_{t-1}, {}^{\lfloor d}D_{t-1}, \nu_{t-1}) + wGiW_{\theta,r}(C_t^U, \hat{\theta}_t^U, {}^{\lfloor d}D_t^U, \nu_t^U).$$

Using basic properties of GiW pdf (Proposition 31) we get:

$$p \equiv \int \frac{1}{r} f(\theta, r) d\theta dr = \underbrace{(1-w)\frac{\nu_{t-1}}{{}^{\lfloor d}D_{t-1}}}_{\equiv p_0} + \underbrace{w\frac{\nu_t^U}{{}^{\lfloor d}D_t^U}}_{\equiv p_u} \tag{6.4}$$

a)			*b)*		
V_{t-1}	$\equiv$	$\begin{pmatrix} 1.16 & 0.12 \\ 0.12 & 0.83 \end{pmatrix}$	V_{t-1}	$\equiv$	$\begin{pmatrix} 1.96 & -1.47 \\ -1.47 & 6.07 \end{pmatrix}$
ν_{t-1}	$\equiv$	102.82	ν_{t-1}	$\equiv$	108.06
d_t	$\equiv$	-0.59	d_t	$\equiv$	-0.79
$\hat{\theta}_{t-1}$	$=$	0.14	$\hat{\theta}_{t-1}$	$=$	-0.24
C_{t-1}	$=$	1.20	C_{t-1}	$=$	0.16
$\lfloor^d D_{t-1}$	$=$	1.14	$\lfloor^d D_{t-1}$	$=$	1.60
V_t^U	$=$	$\begin{pmatrix} 1.50 & 0.12 \\ 0.12 & 1.83 \end{pmatrix}$	V_t^U	$=$	$\begin{pmatrix} 2.58 & -2.26 \\ -0.47 & 7.07 \end{pmatrix}$
ν_t^U	$=$	103.82	ν_t^U	$=$	109.06
ζ_t	$=$	1.20	ζ_t	$=$	0.16
$\hat{e}_t$	$=$	-0.730	$\hat{e}_t$	$=$	-0.54
z_t	$=$	-0.73	z_t	$=$	0.16
h_C	$=$	-0.45	h_C	$=$	-0.86
h_θ	$=$	-0.33	h_θ	$=$	-0.47
$\hat{\theta}_t^U$	$=$	-0.25	$\hat{\theta}_t^U$	$=$	-0.32
C_t^U	$=$	0.54	C_t^U	$=$	0.14
$\lfloor^d D_t^U$	$=$	1.38	$\lfloor^d D_t^U$	$=$	1.86

Table 6.1: Statistics of updated posterior densities

The table shows statistics of posterior pdf and updated posterior pdf for two cases. It also shows some auxiliary values needed for evaluating the statistics of updated pdf. In subsequent examples, another computations with the statistics and auxiliary variables will be performed.

$$s \equiv \int \ln(r)\, f(\theta, r) d\theta dr = (1-w) \ln\left(0.5 \lfloor^d D_{t-1}\right) + w \ln\left(0.5 \lfloor^d D_t^U\right) - \\ - (1-w)\psi_0\left(0.5\nu_{t-1}\right) - w\psi_0\left(0.5\nu_t^U\right).$$

It is clear that both scalars p, s are finite for $\lfloor^d D_t > 0,\ \nu_t > 0$. Now let us evaluate the form of pdf $h(\theta, r)$ from Proposition 1. Again, using Proposition 31, we simply get:

$$h(\theta, r) = \frac{p_0}{p} GiW_{\theta,r}(C_{t-1}, \hat{\theta}_{t-1},\ \lfloor^d D_{t-1}, \nu_{t-1} + 2) + \frac{p_u}{p} GiW_{\theta,r}(C_t^U, \hat{\theta}_t^U,\ \lfloor^d D_t^U, \nu_t^U + 2).$$

The use of Proposition 22 gives:

$$\mathbf{cov}\left[\theta\right]_h = \frac{p_0}{p} \frac{\lfloor^d D_{t-1}}{\nu_{t-1}} C_{t-1} + \frac{p_u}{p} \frac{\lfloor^d D_t^U}{\nu_t^U} C_t^U + \frac{p_0 p_u}{p^2} (\hat{\theta}_{t-1} - \hat{\theta}_t^U)(\hat{\theta}_{t-1} - \hat{\theta}_t^U)'.$$

Matrices C_{t-1} and C_t^U were positive definite, hence $\mathbf{cov}\left[\theta\right]_h$ is also positive definite.

The assumptions of Proposition 1 are hence fulfilled, and we can obtain the optimization result using the definition of p (6.4).

$$\begin{aligned} C_t &= p\ \mathbf{cov}\left[\theta\right]_h = (1-w) C_{t-1} + w(C_{t-1} + h_C z_t z_t') + \\ &+ \frac{p_0 p_u}{p} (\hat{\theta}_{t-1} - \hat{\theta}_{t-1} - h_\theta z_t)(\hat{\theta}_{t-1} - \hat{\theta}_{t-1} - h_\theta z_t)' = \\ &= C_{t-1} + \left[w h_c + \frac{p_0 p_u}{p} h_\theta^2 \right] z_t z_t' \\ \hat{\theta}_t &= \mathcal{E}\left[\theta\right]_h = \frac{p_0}{p} \hat{\theta}_{t-1} + \frac{p_u}{p} (\hat{\theta}_{t-1} + h_\theta z_t) = \hat{\theta}_{t-1} + \left[\frac{p_u}{p} h_\theta \right] z_t \\ \nu_t &= \text{solution of } \ln(0.5\nu_t) - \psi_0(0.5\nu_t) = \ln(p) + s \\ \lfloor^d D_t &= \frac{\nu_t}{p} \end{aligned}$$

Straightforward application of previous considerations yields the following algorithm.

Algorithm 5 (Optimization of statistics) $(C_t, \hat{\theta}_t, \lfloor^d D_t, \nu_t) = \text{FACUPDT}\big(w, C_{t-1}, \nu_{t-1}, \hat{\theta}_{t-1}, \lfloor^d D_{t-1}\big)$

1. $\hat{e}_t = d_t - \hat{\theta}'_{t-1}\psi_t, \quad \zeta_t = \psi'_t C_{t-1}\psi_t$
2. $\nu_t^U = \nu_{t-1} + 1, \quad \lfloor^d D_t^U = \lfloor^d D_{t-1} + \frac{\hat{e}_t^2}{1+\zeta_t}$
3. $p_0 = (1-w)\frac{\nu_{t-1}}{\lfloor^d D_{t-1}}, \quad p_u = w\frac{\nu_t^U}{\lfloor^d D_t^U}, \quad p = p_0 + p_u$
4. $h_\theta = \frac{\hat{e}_t}{1+\zeta_t}, \quad h_C = -\frac{1}{1+\zeta_t}$
5. $\Upsilon = (1-w)\left[\psi_0\left(0.5\nu_{t-1}\right) - \ln\left(\lfloor^d D_{t-1}\right)\right] + w\left[\psi_0\left(0.5\nu_t^U\right) - \ln\left(\lfloor^d D_t^U\right)\right] - \ln\left(0.5p\right)$
6. $z_t = C_{t-1}\psi_t$
7. $\nu_t = GETNU(\Upsilon)$ *(Algorithm 19, page 96)*
8. $\lfloor^d D_t = \frac{\nu_t}{p}$
9. $\hat{\theta}_t = \hat{\theta}_{t-1} + \left[\frac{p_u}{p}h_\theta\right] z_t$
10. $C_t = C_{t-1} + \left[wh_c + \frac{p_0 p_u}{p}h_\theta^2\right] z_t z_t'$

Example 13 (Optimization of statistics)

ψ_t	$\equiv$	(1)	*(regression vector)*
Θ	$\equiv$	(θ, r)	*(unknown parameter)*
$f(d_t\|\psi_t, \Theta)$	$\equiv$	$\mathcal{N}_{d_t}(\theta, r)$	*(normal static factor)*
$\pi(\Theta\|\mathcal{S}_{t-1})$	$\equiv$	$GiW_{\theta,r}(V_{t-1}, \nu_{t-1})$	*(GiW posterior)*
$\pi^U(\Theta\|\mathcal{S}_t^U)$	$\equiv$	$GiW_{\theta,r}(V_t^U, \nu_t^U)$	*(factor update)*
$f(\theta, r)$	$\equiv$	$(1-w)GiW_{\theta,r}(V_{t-1}, \nu_{t-1}) + wGiW_{\theta,r}(V_t^U, \nu_t^U)$	*(Bayesian update)*

Statistic $\mathcal{S}_t \equiv (V_t, \nu_t) \equiv (C_t, \hat{\theta}_t, \lfloor^d D_t, \nu_t)$ *was evaluated using projection based algorithm for the same two cases as Example 12. Table 6.2 shows statistics of the involved pdfs and some other mentioned auxiliary values for both cases. Figure 6.4 plots marginal pdfs of some involved pdfs.*

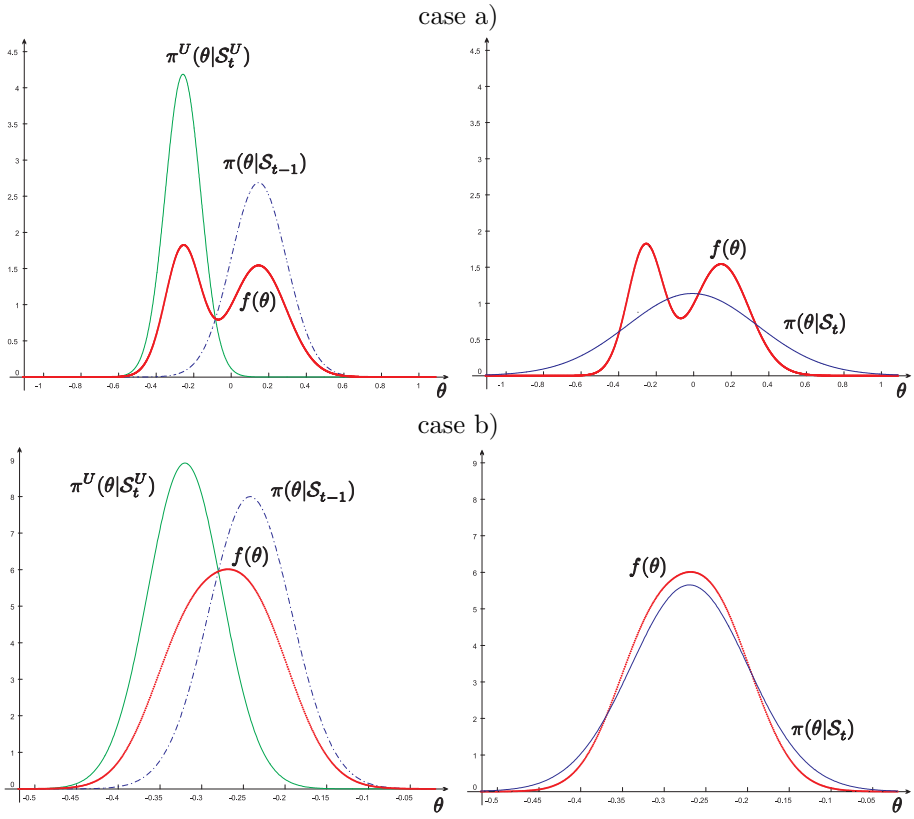


Figure 6.4: Marginal pdfs resulting from PB algorithm

The left part shows marginal pdfs of original factor posterior $\pi(\theta|\mathcal{S}_{t-1})$ (dashdot), its update $\pi^U(\theta|\mathcal{S}_t^U)$ (dotted) and the correct Bayesian update $f(\theta)$ (thick), i.e. the mixture of the two mentioned factors. The right part shows how the result of PB algorithm (solid) approximates the correct Bayesian update $f(\theta)$ (thick). In the case a), the approximation doesn't look very nice, but it at least covers the correct range. In the second case, the approximation looks nice enough.

a)			*b)*		
V_{t-1}	$\equiv$	$\begin{pmatrix} 1.16 & 0.12 \\ 0.12 & 0.83 \end{pmatrix}$	V_{t-1}	$\equiv$	$\begin{pmatrix} 1.96 & -1.47 \\ -1.47 & 6.07 \end{pmatrix}$
ν_{t-1}	$\equiv$	102.82	ν_{t-1}	$\equiv$	108.06
d_t	$\equiv$	-0.59	d_t	$\equiv$	-0.79
w	$\equiv$	0.43	w	$\equiv$	0.39
p_0	$=$	51.29	p_0	$=$	41.09
p_u	$=$	32.18	p_u	$=$	22.85
p	$=$	83.47	p	$=$	63.94
Υ	$=$	-0.0138	Υ	$=$	-0.0115
ν_t	$=$	72.57	ν_t	$=$	86.93
$\hat{\theta}_t$	$=$	-0.01	$\hat{\theta}$	$=$	-0.27
C_t	$=$	4.10	C_t	$=$	0.24
$\lfloor^d D_t$	$=$	0.87	$\lfloor^d D_t$	$=$	1.36

Table 6.2: Statistics optimized using PB algorithm

6.1.5 Quasi-Bayes as Approximation

According to Propositions 23 and 24, the minimization

$$(V_t, \nu_t) \in \mathrm{Arg}\min_{(V_t,\nu_t)} \mathcal{D}\left((1-w)GiW_{\theta,r}(V_{t-1},\nu_{t-1}) + wGiW_{\theta,r}(V_t^U,\nu_t^U) \;\middle\|\; GiW_{\theta,r}(V_t,\nu_t)\right)$$

is equivalent to minimization

$$\begin{aligned}(V_t, \nu_t) \in \mathrm{Arg}\min_{(V_t,\nu_t)} \quad & (1-w)\mathcal{D}\left(GiW_{\theta,r}(V_{t-1},\nu_{t-1}) \;\middle\|\; GiW_{\theta,r}(V_t,\nu_t)\right) + \\ & + w\mathcal{D}\left(GiW_{\theta,r}(V_t^U,\nu_t^U) \;\middle\|\; GiW_{\theta,r}(V_t,\nu_t)\right).\end{aligned}$$

If we approximate

$$\mathcal{D}\left(GiW_{\theta,r}(V_{t-1},\nu_{t-1}) \;\middle\|\; GiW_{\theta,r}(V_t,\nu_t)\right) \to ||V_{t-1}-V_t||^2 + ||\nu_{t-1}-\nu_t||^2$$

and

$$\mathcal{D}\left(GiW_{\theta,r}(V_t^U,\nu_t^U) \;\middle\|\; GiW_{\theta,r}(V_t,\nu_t)\right) \to ||V_t^U-V_t||^2 + ||\nu_t^U-\nu_t||^2,$$

we can quickly achieve the result

$$V_t = V_{t-1} + w\Psi_t\Psi_t',\ \nu_t = \nu_{t-1} + w,$$

which is exactly the same as the quasi-Bayes update [21].

Example 14 (QB update) *Table 6.3 shows numerical results of the QB algorithm on the same cases as Example 13. Figure 6.5 shows how the result of QB estimation differs from the correct Bayesian update.*

a)			*b)*		
ν_t	=	103.25	ν_t	=	108.45
$\hat{\theta}_t$	=	0.14	$\hat{\theta}$	=	−0.24
C_t	=	1.20	C_t	=	0.16
$\lfloor^d D_t$	=	1.14	$\lfloor^d D_t$	=	1.60

Table 6.3: Statistics of pdfs updated with QB algorithm

The presented approximation is not important in the sense of a speed increase. It has almost the same computational complexity as PB algorithm. Just Step 7 of PB algorithm (Algorithm 5) is replaced with simple assignment $\nu_t = \nu_{t-1} + w_t$. This doesn't bring substantial speed increase, because the one-dimensional nonlinear equation in Step 7 is solved very fast using Newton method and a good starting point.

It is important, because it explains the well -known heuristic quasi-Bayes algorithm as an approximation of the general PB approach.

6.2 Normal Factors with Known Variance

In this section, we assume the parameterized factor to be dynamic Gaussian pdf with a known variance. This factor variant is important, because in some applications of mixture estimation, we have to assume the knowledge of factor variance [10]. Because the evaluation of all problems related to normal factors with known variance are simplified cases of those with unknown variance, we will concentrate on showing the results not on their derivation.

In this case, $\Theta \equiv (\theta)$, the form of the factor is the same as in previous case. The variance r is expected to be known and it is not specified as input to algorithms.

$$f(d|\psi,\Theta) = \mathcal{N}_d(\theta'\psi, r) = \frac{1}{\sqrt{2\pi r}}\exp\left(-\frac{(d_t-\theta'\psi_t)^2}{2r}\right) \tag{6.5}$$

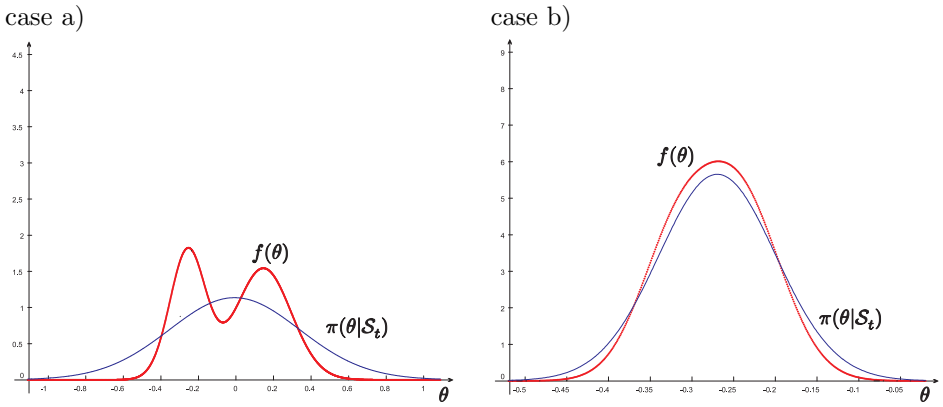


Figure shows how the QB update (solid) approximates the correct Bayesian update $f(\theta)$ (thick).

Figure 6.5: Marginal pdf of QB update

We can compare result of QB algorithm, with result of PB algorithm displayed on Figure 6.4. In case a, both approximation are inaccurate, but the PB algorithm at least covers the correct range. In case b, both approximation gives acceptable results, but it can be seen that the result of PB algorithms looks better.

6.2.1 Form of the Posterior Pdf

The conjugated pdf to factor (6.5) is the Gaussian pdf with statistics $\mathcal{S}_t \equiv (M_t, R_t)$, where M_t is mean and R_t is covariance matrix. Hence we will use it in the class of considered mixture posterior pdfs.

$$\pi(\Theta|\mathcal{S}_t) \equiv \mathcal{N}_\theta\left(M_t, R_t\right) \propto \exp\left\{-\frac{1}{2}(\theta - M_t)' R_t^{-1}(\theta - M_t)\right\}$$

6.2.2 Factor Prediction

The factor prediction $\mathcal{I}_t$ for normal factor with known variance and conjugate prior is evaluated as follows:

$$\mathcal{I}_t = \frac{\exp\left(-\frac{\hat{e}_t^2}{2r(1+\zeta_t)}\right)}{\sqrt{2\pi r(1+\zeta_t)}}, \text{ where} \tag{6.6}$$

$$\begin{aligned} \zeta_t &\equiv \psi_t' R_{t-1} \psi_t \\ \hat{e}_t &\equiv d_t - M_{t-1}' \psi_t \end{aligned}$$

Algorithm 6 (Factor prediction) $(\mathcal{L}_t) = \text{FACNORM}\left(M_{t-1}, R_{t-1}\right)$

1. *Evaluate* $\zeta_t = \psi_t' R_{t-1} \psi_t$

2. *Evaluate* $\hat{e}_t = d_t - M_{t-1}' \psi_t$

3. *Evaluate*

$$\mathcal{L}_t \equiv \ln\left(\mathcal{I}_t\right) = -\frac{\hat{e}_t^2}{2r(1+\zeta_t)} - 0.5\ln\left(2\pi r\right) - 0.5\ln\left(1+\zeta_t\right)$$

6.2.3 Factor Update

$\mathcal{S}_t^U \equiv [R_t^U, M_t^U]$ can be evaluated in the following way:

$$\begin{aligned} R_t^U &= R_{t-1} + h_R z_t z_t', \qquad M_t^U = M_{t-1} + h_M z_t \\ h_R &\equiv -\frac{1}{r+\zeta}, \quad h_M \equiv \frac{\hat{e}_t}{r+\zeta_t} \end{aligned}$$

6.2.4 Optimization of Statistics

We will use Proposition 2. First we have to check if our case fulfills its assumptions. The pdf f from the proposition has the form

$$f(\theta, r) \equiv (1-w)\mathcal{N}_\theta\left(M_{t-1}, R_{t-1}\right) + w\mathcal{N}_\theta\left(M_t^U, R_t^U\right).$$

Now we can use Proposition 22, which yields:

$$\mathbf{cov}\left[\theta\right]_f = (1-w)R_{t-1} + wR_t^U + w(1-w)(M_{t-1} - M_{t-1}^U)(M_{t-1} - M_{t-1}^U)'.$$

Matrices R_{t-1} and R_t^U were positive definite, hence $\mathbf{cov}\left[\theta\right]_f$ is also positive definite. The assumptions of Proposition 2 are hence fulfilled, and we can obtain the result.

$$\begin{aligned} R_t &= \mathbf{cov}\left[\theta\right]_h = (1-w)R_{t-1} + w(R_{t-1} + h_R z_t z_t') + \\ &\quad + w(1-w)(M_{t-1} - M_{t-1} - h_M z_t)(M_{t-1} - M_{t-1} - h_\theta z_t)' = \\ &= R_{t-1} + \left[wh_R + w(1-w)h_M^2\right] z_t z_t' \\ M_t &= \mathcal{E}\left[\theta\right]_h = (1-w)M_{t-1} + w(M_{t-1} + h_M z_t) = M_{t-1} + \left[wh_M\right] z_t \end{aligned}$$

Straightforward application of previous results gives the following algorithm.

Algorithm 7 (Optimization of statistics) $(R_t, M_t) = \mathrm{FACUPDT}(w, R_{t-1}, M_{t-1})$

1. $\hat{e}_t = d_t - M_{t-1}'\psi_t, \quad \zeta_t = \psi_t' R_{t-1}\psi_t$
2. $h_R = -\frac{1}{r+\zeta}, \quad h_M = \frac{\hat{e}_t}{r+\zeta_t}$
3. $z_t = R_{t-1}\psi_t$
4. $M_t = M_{t-1} + \left[wh_M\right] z_t$
5. $R_t = R_{t-1} + \left[wh_R + w(1-w)h_M^2\right] z_t z_t'$

Chapter 7

Optimization of Statistics of Component Weighting Functions

This chapter deals with general steps from Chapter 5 for some specific types of cwfs. Specially, we need to perform following tasks:

to evaluate weight estimates $\hat{\alpha}_{c;t-1}$ (5.3) (page 35),

to evaluate cwf updates $\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U)$ (5.6),

to perform minimization
$\mathcal{H}_t \in \operatorname{Arg\,min}_{\mathcal{H}_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \;\middle|\middle|\; \rho(\Omega|\mathcal{H}_t)\right)$ (5.8).

In other words, we need to design the algorithms proposed in Section 5.3:

$(\mathcal{Z}_{c;t-1})$= WEIGHTNORM$(\mathcal{H}_{t-1})$,

$(\mathcal{H}_t)$= WEIGHTUPDT$(\mathcal{H}_{t-1}, w_\bullet)$.

The algorithm second algorithm solves the second and third tasks listed above.

7.1 Constant Component Weights

In this section, we examine the simplest case of cwf:

$$\alpha_c(\phi_{t-1}|\Omega) \equiv \alpha_c(\phi_{t-1}|\alpha) \equiv \alpha_c, \ \forall c, \tag{7.1}$$

where $\Omega \equiv \alpha$ is a vector of $\mathring{c}$ nonnegative entries fulfilling the condition $\sum_{c=1}^{\mathring{c}} \alpha_c = 1$.

7.1.1 Form of Posterior Pdf

It is reasonable to choose the posterior pdf of α as Dirichlet distribution (Section C.2).

$$\rho(\Omega|\mathcal{H}_{t-1}) \equiv \rho(\alpha|\kappa_{t-1}) \equiv Di_\alpha(\kappa_{t-1}), \tag{7.2}$$

where $\mathcal{H}_{t-1} \equiv \kappa_{t-1}$ is a vector with $\mathring{c}$ positive entries.

7.1.2 Weight Estimate

It holds that

$$\hat{\alpha}_{c;t-1} = \int \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega \overbrace{\equiv}^{(7.1),(7.2)} \int \alpha_c Di_\alpha(\kappa_{t-1})d\alpha \overbrace{=}^{(C.2.3)} \frac{\kappa_{c;t-1}}{\sum_{\check{c}=1}^{\mathring{c}} \kappa_{\check{c};t-1}}.$$

Thus, we can simply formulate the algorithm WEIGHTNORM evaluating logarithm of $\hat{\alpha}_{\bullet;t-1}$.

Algorithm 8 (Weight estimate) $(\mathcal{Z}_{\bullet;t-1})=$ WEIGHTNORM$\big(\kappa_{t-1}\big)$

1. *Evaluate temporary variable* $Q = \ln\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;t-1}\right)$
2. *For each component* c *evaluate* $\mathcal{Z}_{c;t-1} = \ln\left(\kappa_{c;t-1}\right) - Q$.

7.1.3 Cwf Update

$$\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) = \frac{\alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})}{\hat{\alpha}_{c;t-1}} \overbrace{\equiv}^{(7.1),(7.2)} \frac{\alpha_c Di_\alpha(\kappa_{t-1})}{\hat{\alpha}_{c;t-1}} \overbrace{=}^{(C.10)} Di_\alpha(\kappa_{t-1}+\delta_{\bullet c}) \tag{7.3}$$

7.1.4 Optimization of Cwf Statistics

We have to perform the following optimization task (5.8):

$$\mathcal{H}_t \in \operatorname{Arg\,min}_{\mathcal{H}_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) \;\middle\|\; \rho(\Omega|\mathcal{H}_t)\right)$$

Applied to our task, it reads (using (7.3)):

$$\kappa_t \in \operatorname{Arg\,min}_{\kappa_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t} Di_\alpha(\kappa_{t-1}+\delta_{\bullet c}) \;\middle\|\; Di_\alpha(\kappa_t)\right). \tag{7.4}$$

The following proposition converts this task to minimization of an algebraic expression.

Proposition 6 (Minimization with respect to κ_t)
For κ_t *minimizing*

$$\mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t} Di_\alpha(\kappa_{t-1}+\delta_{\bullet c}) \;\middle\|\; Di_\alpha(\kappa_t)\right)$$

it holds that

$$\kappa_{\bullet;t} \in \operatorname{Arg\,min}\left\{\sum_{c=1}^{\mathring{c}}\Big[\ln\left(\Gamma\left(\kappa_{c;t}\right)\right) - \kappa_{c;t}\xi_{c;t}\Big] - \ln\left(\Gamma\left(\sum_{c=1}^{\mathring{c}}\kappa_{c;t}\right)\right)\right\}$$

where

$$\xi_{c;t} = \left(\psi_0\left(\kappa_{c;t-1}\right) + \frac{w_{c,t}}{\kappa_{c;t-1}} - \psi_0\left(\sum_{c=1}^{\mathring{c}}\kappa_{c;t-1}+1\right)\right),$$

$\Gamma(x)$ *is gamma function and* $\psi_0\left(x\right)$ *is digamma function (see Appendix B.3).*

Proof: According to Propositions 23 and 24, we can minimize

$$\sum_{c=1}^{\mathring{c}} w_{c;t} \mathcal{D}\left(Di_{\alpha}(\kappa_{t-1}+\delta_{\bullet c}) \,\middle\|\, Di_{\alpha}(\kappa_t)\right).$$

Proposition 27, which evaluates KL divergence of two Dirichlet pdfs, yields the following expression to be minimized:

$$\begin{aligned}
&\sum_{c=1}^{\mathring{c}} w_{c;t} \mathcal{Z}(\kappa_{t-1}, \kappa_t, c), \text{ where} \\
&\mathcal{Z}(\kappa_{t-1}, \kappa_t, c) = \sum_{j=1}^{\mathring{c}} \left[\ln\left(\Gamma\left(\kappa_{j;t}\right)\right) - \kappa_{j;t}\psi_0\left(\kappa_{j;t-1}+\delta_{cj}\right)\right] - \\
&- \left[\ln(\Gamma(\sum_{k=1}^{\mathring{c}} \kappa_{k;t}) - \sum_{j=1}^{\mathring{c}} \kappa_{j;t}\psi_0\left(\sum_{k=1}^{\mathring{c}} \kappa_{k;t-1}+1\right)\right].
\end{aligned}$$

Because $\psi_0\left(\kappa_{j;t-1}+\delta_{cj}\right) = \psi_0\left(\kappa_{j;t-1}\right) + \frac{\delta_{cj}}{\kappa_{j;t-1}}$(Proposition 16) and $\sum_{j=1}^{\mathring{c}} \frac{\delta_{cj}}{\kappa_{j;t-1}} = \frac{1}{\kappa_{c;t-1}}$:

$$\begin{aligned}
\mathcal{Z}(\kappa_{t-1}, \kappa_t, c) \;=\; & \sum_{j=1}^{\mathring{c}} \left[\ln\left(\Gamma\left(\kappa_{j;t}\right)\right) - \kappa_{j;t}\psi_0\left(\kappa_{j;t-1}\right) - \kappa_{j;t}\psi_0\left(\sum_{k=1}^{\mathring{c}} \kappa_{k;t-1}+1\right)\right] - \\
& - \ln\left(\Gamma\left(\sum_{k=1}^{\mathring{c}} \kappa_{k;t}\right)\right) - \frac{1}{\kappa_{c;t-1}}.
\end{aligned}$$

Because the only term depending on c is $\frac{1}{\kappa_{c;t-1}}$ and $\sum_{c=1}^{\mathring{c}} w_{c;t} = 1$:

$$\begin{aligned}
\sum_{c=1}^{\mathring{c}} w_{c;t} \mathcal{Z}_{c;t} \;=\; & \sum_{j=1}^{\mathring{c}} \left[\ln\left(\Gamma\left(\kappa_{j;t}\right)\right) - \kappa_{j;t}\psi_0\left(\kappa_{j;t-1}\right) - \kappa_{j;t}\psi_0\left(\sum_{k=1}^{\mathring{c}} \kappa_{k;t-1}+1\right)\right] - \\
& - \ln\left(\Gamma\left(\sum_{k=1}^{\mathring{c}} \kappa_{k;t}\right)\right) - \sum_{c=1}^{\mathring{c}} \frac{w_{c;t}}{\kappa_{c;t-1}} = \\
=\; & \sum_{j=1}^{\mathring{c}} \left[\ln\left(\Gamma\left(\kappa_{j;t}\right)\right) - \kappa_{j;t} \underbrace{\left(\psi_0\left(\kappa_{j;t-1}\right) + \frac{w_{j,t}}{\kappa_{j;t-1}} - \psi_0\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;t-1}+1\right)\right)}_{\xi_{j;t}}\right] - \\
& - \ln\left(\Gamma\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;t}\right)\right)
\end{aligned}$$

□

Proposition 6 yields the following algorithm.

Algorithm 9 (Optimization of cwf statistics) $(\kappa_{\bullet;t}) = \text{WEIGHTUPDT}\left(w_{\bullet;t}, \kappa_{\bullet;t-1}\right)$

1. *For each component c evaluate* $\xi_{c;t} = \psi_0\left(\kappa_{c;t-1}\right) + \frac{w_{c,t}}{\kappa_{c;t-1}} - \psi_0\left(\sum_{c=1}^{\mathring{c}} \kappa_{c;t-1}+1\right)$
2. $\kappa_{\bullet;t} \in \operatorname{Arg\,min}\left\{\sum_{j=1}^{\mathring{c}} \left[\ln\left(\Gamma\left(\kappa_{j;t}\right)\right) - \kappa_{j;t}\xi_{j;t}\right] - \ln\left(\Gamma\left(\sum_{c}^{\mathring{c}} \kappa_{c;t}\right)\right)\right\}$

Remarks 6
The minimization problem in step 2 can be solved numerically or by suitable approximation.(See the next paragraph) For a detailed solution of this problem, see [42].

7.1.5 Quasi-Bayes as Approximation

Minimization (7.4) can be simply approximated. According to Propositions 23 and 24, the minimization

$$\kappa_t \in \mathrm{Arg}\min_{\kappa_t} \mathcal{D}\left(\sum_{c=1}^{\mathring{c}} w_{c;t} Di_\alpha(\kappa_{t-1}+\delta_{\bullet c}) \;\middle|\middle|\; Di_\alpha(\kappa_t)\right)$$

reduces to the minimization

$$\kappa_t \in \mathrm{Arg}\min_{\kappa_t} \sum_{c=1}^{\mathring{c}} w_{c;t} \mathcal{D}\left(Di_\alpha(\kappa_{t-1}+\delta_{\bullet c}) \;\middle|\middle|\; Di_\alpha(\kappa_t)\right).$$

By approximating $\mathcal{D}\left(Di_\alpha(\kappa_{t-1}+\delta_{\bullet c}) \;\middle|\middle|\; Di_\alpha(\kappa_t)\right)$ with square of the Euclidean norm $||\kappa_{t-1}+\delta_{\bullet c}-\kappa_t||^2$, the problem is transformed into minimization of

$$\sum_{c=1}^{\mathring{c}} w_{c;t}||\kappa_{t-1}+\delta_{\bullet c}-\kappa_t||^2.$$

It can be simply shown that the previous expression is minimized by $\kappa_t = \kappa_{t-1} + w_t$, which is identical to the solution obtained using the quasi-Bayes algorithm (Appendix A).

The approximation replaced the problem of finding minimizer of a convex function with $\mathring{c}$ variables with a simple assignment. It was shown [42] that results obtained using the approximation are in fact almost the same as results using numerical solution. Hence, in the resulting PB algorithm, this approximation is used. Although there exist a good approximation of the starting point for iterative numerical algorithm, which guarantees relatively quick solution of this task [42], it pays back to use the mentioned approximation.

7.2 Dynamic Weights

We will try to derive algorithms for updating the statistics $\mathcal{H}_t$ as general as possible. Hence we will not specify the precise form of component weighting functions in following evaluations, but we will make some assumption about the parameter Ω.

Because some variables and statistics introduced in the following text have the same names as the variables and statistics related to factors and their posteriors, the variables and statistics related to cwfs are prefixed by the sign $^{\lfloor\alpha}$, e.g. $^{\lfloor\alpha}\hat{\theta}$.

7.2.1 Form of Posterior Pdf

Let us assume that Ω consists of n conditionally independent parts $\Omega \equiv (\Omega_1, \cdots, \Omega_n)$. The posterior pdf on Ω is then equal to

$$\rho(\Omega|\mathcal{H}_t) = \prod_{k=1}^{n} \rho_k(\Omega_k|\mathcal{H}_{k;t}),\ \mathcal{H}_t \equiv (\mathcal{H}_{1;t}, \cdots, \mathcal{H}_{n;t}). \tag{7.5}$$

The particular pdfs $\rho_k(\Omega_k|\mathcal{H}_{k;t})$ will be assumed to be either GiW or Gaussian pdfs.

GiW

In the case when the parameter Ω_k consists of a pair (vector,scalar), $\Omega_k \equiv ({}^{\lfloor\alpha}\theta_k,\ {}^{\lfloor\alpha}r_k)$, we can consider the posterior pdf on $({}^{\lfloor\alpha}\theta_k,\ {}^{\lfloor\alpha}r_k)$ to be GiW pdf given by the statistics $({}^{\lfloor\alpha}V_{k;t}\ {}^{\lfloor\alpha}\nu_{k;t})$.

$$\rho_k(\Omega_k|\mathcal{H}_{k;t}) \equiv GiW_{{}^{\lfloor\alpha}\theta_k,\ {}^{\lfloor\alpha}r_k}({}^{\lfloor\alpha}V_{k;t},\ {}^{\lfloor\alpha}\nu_{k;t}),\ \mathcal{H}_{k;t} \equiv ({}^{\lfloor\alpha}V_{k;t}\ {}^{\lfloor\alpha}\nu_{k;t})$$

Gaussian

In the case when the parameter Ω_k is a vector ($\Omega_k \equiv {}^{\lfloor\alpha}\theta_k$), we can consider the posterior pdf on ${}^{\lfloor\alpha}\theta_k$ to be Gaussian pdf given by the statistics $(M_{k;t}, R_{k;t})$.

$$\rho_k(\Omega_k|\mathcal{H}_{k;t}) \equiv \mathcal{N}_{{}^{\lfloor\alpha}\theta_k}(M_{k;t}, R_{k;t}),\ \mathcal{H}_{k;t} \equiv (M_{k;t}, R_{k;t})$$

For formal purposes, let us define two sets: $GA \subset \{1,\cdots,n\},\ GI \subset \{1,\cdots,n\}$, GA contains all indexes for which $\rho_k(\Omega_k|\mathcal{H}_{k;t})$ is Gaussian pdf, GI is complement of GA, $GI = \{1,\cdots,n\} - GA$, i.e. indexes in GI point to GiW pdfs.

7.2.2 Weight Estimate

The weight estimate $\hat{\alpha}_{c;t-1}$ is defined as follows:

$$\hat{\alpha}_{c;t-1} = \int \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega.$$

It can not be simplified at this general level, we can just recall that we are looking for an algorithm:

$$\ln(\hat{\alpha}_{\bullet;t-1}) \equiv \mathcal{Z}_{\bullet,t-1} = \text{WEIGHTNORM}(\mathcal{H}_{t-1}).$$

7.2.3 Cwf Update

We have to evaluate the expression

$$\rho_c^U(\Omega|\mathcal{H}^U_{c;t-1}) \propto \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1}),$$

but at this general level, we cannot proceed similarly as in previous section. This form will be used in the next computations.

7.2.4 Optimization of Cwf Statistics

We have to minimize:

$$\mathcal{H}_t \in \operatorname{Arg\,min}_{\mathcal{H}_t} \mathcal{D}\left(\underbrace{\sum_{c=1}^{\hat{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}^U_{c;t-1})}_{\equiv h(\Omega)} \,\Big|\Big|\, \rho(\Omega|\mathcal{H}_t)\right). \tag{7.6}$$

According to Proposition 25, for the selected form of posterior pdf (7.5), the previous problem reduces to subproblems:

$$\mathcal{H}_{k;t} \in \operatorname{Arg\,min} \mathcal{D}\left(h(\Omega_k) \,\Big|\Big|\, \rho_k(\Omega_k|\mathcal{H}_{k;t})\right),\quad \forall k \in \hat{n},$$

where $h(\Omega_k)$ are corresponding marginal pdfs of $h(\Omega)$ in (7.6).

For our case, it means that we need to solve subproblems of type

$$\text{Arg}\min_{M_{k;t},R_{k;t}} \mathcal{D}\left(h(\,{}^{\lfloor\alpha}\theta_k)\;\middle\|\;\mathcal{N}_{\lfloor\alpha\theta_k}(M_{k;t},R_{k;t})\right)\;\forall k\in GA \text{ and} \tag{7.7}$$

$$\text{Arg}\min_{\lfloor\alpha V_{k;t},\,\lfloor\alpha\nu_{k;t}} \mathcal{D}\left(h(\,{}^{\lfloor\alpha}\theta_k,\,{}^{\lfloor\alpha}r_k)\;\middle\|\;GiW_{\lfloor\alpha\theta_k,\,\lfloor\alpha r_k}(\,{}^{\lfloor\alpha}V_{k;t},\,{}^{\lfloor\alpha}\nu_{k;t})\right)\;\forall k\in GI. \tag{7.8}$$

According to Proposition 2 (assuming its assumptions hold), the subproblems (7.7) have solution

$$\begin{aligned} M_{k;t} &\equiv \mathcal{E}\left[\,{}^{\lfloor\alpha}\theta_k\right]_h = \int {}^{\lfloor\alpha}\theta_k h(\Omega_k)d\,{}^{\lfloor\alpha}\theta_k = \int {}^{\lfloor\alpha}\theta_k h(\Omega)d\Omega \\ R_{k;t} &\equiv \mathbf{cov}\left[\,{}^{\lfloor\alpha}\theta_k\right]_h = \int {}^{\lfloor\alpha}\theta_k\,{}^{\lfloor\alpha}\theta_k' h(\Omega_k)d\,{}^{\lfloor\alpha}\theta_k - M_{k;t}M_{k;t}', \end{aligned}$$

Solution of the subproblems (7.8) is little bit more complicated. The resulting expression are in $C,\hat{\theta},\,{}^{\lfloor d}D$ representation again. According to Proposition 1 (assuming its assumptions hold):

$$\begin{aligned} {}^{\lfloor\alpha}\hat{\theta}_{k;t} &\equiv \mathcal{E}\left[\,{}^{\lfloor\alpha}\theta_k\right]_{\frac{h}{p_k\,\lfloor\alpha r_k}} = \frac{1}{p_k}\int \frac{{}^{\lfloor\alpha}\theta_k}{{}^{\lfloor\alpha}r_k}h(\Omega_k)d\Omega_k \\ {}^{\lfloor\alpha}C_{k;t} &\equiv \mathbf{cov}\left[\,{}^{\lfloor\alpha}\theta_k\right]_{\frac{h}{p_k\,\lfloor\alpha r_k}} = p_k\left(\int \frac{{}^{\lfloor\alpha}\theta_k\,{}^{\lfloor\alpha}\theta_k'}{p_k\,{}^{\lfloor\alpha}r_k}h(\Omega_k)d\Omega_k - \hat{\theta}_{k;t}\hat{\theta}_{k;t}'\right) \\ {}^{\lfloor\alpha}\nu_{k;t} &\quad \text{solves } \ln\left(0.5\,{}^{\lfloor\alpha}\nu_{k;t}\right) - \psi_0\left(0.5\,{}^{\lfloor\alpha}\nu_{k;t}\right) = \ln(p_k) + s_k \\ {}^{\lfloor\alpha d}D_{k;t} &\equiv \frac{{}^{\lfloor\alpha}\nu_{k;t}}{p_k}, \text{ where} \\ p_k &= \int \frac{1}{{}^{\lfloor\alpha}r_k}h(\Omega_k)d\Omega_k \\ s_k &= \int \ln\left({}^{\lfloor\alpha}r_k\right)h(\Omega_k)d\Omega_k \end{aligned}$$

Remarks 7

- *The assumption of Proposition 1 must be checked during the use of this algorithm. Nevertheless, they will almost sure never be violated.*
- *These results are very important, because they converted the problem of minimization and divergence evaluation into the evaluation of moments "only". Unfortunately, these moments can be rarely evaluated analytically.*

7.2.5 Approximation

Our ability to obtain feasible algorithms depends on the ability to evaluate the integral (5.3)

$$\hat{\alpha}_{c;t-1} = \int \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega$$

and integrals of type

$$\int K(\Omega_k)h(\Omega)d\Omega, \text{ where}$$

$$h(\Omega) = \sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t}^U) = \rho(\Omega|\mathcal{H}_{t-1})\sum_{c=1}^{\mathring{c}}\frac{w_{c;t}}{\hat{\alpha}_{c;t}}\alpha_t(\phi_{t-1}|\Omega).$$

We need to evaluate the mentioned integrals for the following forms of function K

$$\begin{array}{llllll} K(\Omega_k) & \equiv & \frac{\lfloor^{\alpha}\theta_k}{\lfloor^{\alpha}r_k}, & K(\Omega_k) & \equiv & \frac{\lfloor^{\alpha}\theta_k \lfloor^{\alpha}\theta'_k}{\lfloor^{\alpha}r_k}, \quad K(\Omega_k) \equiv \lfloor^{\alpha}\theta_k, \\ K(\Omega_k) & \equiv & \lfloor^{\alpha}\theta_k \lfloor^{\alpha}\theta'_k, & K(\Omega_k) & \equiv & \frac{1}{\lfloor^{\alpha}r_k}, \quad K(\Omega_k) \equiv \ln\left(\lfloor^{\alpha}r_k\right). \end{array}$$

The simplest and universal approximation of all the mentioned integrals is Monte Carlo integration. Hence it was used on the examined cases. In future research, others approximation of the integral have to be used.

Let us generate N samples from $\rho(\Omega|\mathcal{H}_{t-1})$ and denote them $(\Omega^1, \cdots \Omega^N)$. Then, the mentioned integrals can be approximated as follows:

$$\begin{aligned} \hat{\alpha}_{c;t-1} &= \int \alpha_c(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega \approx \frac{1}{N}\sum_{l=1}^{N}\alpha_c(\phi_{c;t-1}|\Omega^l) \\ \int K(\Omega)h(\Omega)d\Omega &\approx \frac{1}{N}\sum_{l=1}^{N}K(\Omega^l)\underbrace{\sum_{c=1}^{\mathring{c}}\frac{w_{c;t}}{\hat{\alpha}_{c;t}}\alpha_c(\phi_{c;t-1}|\Omega^l)}_{\equiv N\times \upsilon_l} \equiv \sum_{l=1}^{N}\upsilon_l K(\Omega^l). \end{aligned} \tag{7.9}$$

The vector υ of length N defined above will be called MC weights.

To apply this approximation, we need to be able to take efficiently samples from $\rho(\Omega|\mathcal{H}_{t-1})$ and to evaluate $\alpha_c(\phi_{t-1}|\Omega)$. For detailed description of Monte-Carlo methods see e.g [43].

Sample Generation

Thanks to the selected form of pdf

$$\rho(\Omega|\mathcal{H}_{t-1}) = \prod_{k=1}^{n}\rho_k(\Omega_k|\mathcal{H}_{k;t-1}),\ \mathcal{H}_{t-1} \equiv (\mathcal{H}_{1;t-1}, \cdots, \mathcal{H}_{n;t-1}),$$

the sample $\Omega^l \equiv (\Omega_1^l, \cdots, \Omega_n^l)$ consists of samples Ω_k^l from $\rho_k(\Omega_k|\mathcal{H}_{k;t-1})$, $\forall k \in \{1, \cdots, n\}$. Because we consider two possible types of densities $\rho_k(\Omega_k|\mathcal{H}_{k;t-1})$, the generation of samples Ω_k^l is performing either for Gaussian pdf ($\Omega_k^l \equiv \lfloor^{\alpha}\theta_k^l$) or for GiW pdf $\Omega_k^l \equiv (\lfloor^{\alpha}\theta_k^l, \lfloor^{\alpha}r_k^l)$. The following algorithm summarizes the sample generation.

Algorithm 10 (Sampling from posterior pdf) $(\Omega^1, \cdots \Omega^N) = \text{SAMPLE}(\mathcal{H}_{t-1}, N)$

FOR $l = 1 : N$

FOR $k = 1 : n$

if $k \in GA$
$(\Omega_k^l \equiv (\lfloor^{\alpha}\theta_k^l)) = \text{GAUSSGEN}(M_{k;t-1}, R_{k;t-1})$ *(Algorithm 23, page 103)*

if $k \in GI$
$(\Omega_k^l \equiv (\lfloor^{\alpha}\theta_k^l, \lfloor^{\alpha}r_k^l)) = \text{GIWGEN}(\lfloor^{\alpha}C_{k;t-1}, \lfloor^{\alpha}\hat{\theta}_{k;t-1}, \lfloor^{\alpha}\lfloor^{d}D_{k;t-1}, \lfloor^{\alpha}\nu_{k;t-1})$ *(Algorithm 22, page 103)*

END FOR

END FOR

Weight-evaluating

We expect, that for each type of cwf there exist an algorithm $(Q_\bullet) = \text{EVAL\_WEIGHT}(\Omega^l)$, evaluating

$$Q_c = \alpha_c(\phi_{c;t-1}|\Omega^l).$$

This algorithm is in-fact the only connection to the form of cwfs. This means, that we can simply use the presented approach with various types of cwfs specifying only algorithm EVAL_WEIGHT for each cwf type.

Weight Estimate

Using the algorithms defined above, it is easy to create an algorithm for approximate evaluation of $\hat{\alpha}_{\bullet;t-1}$

Algorithm 11 (Weight estimate) $(\mathcal{Z}_{\bullet;t-1})$= WEIGHT_NORM$\big(\mathcal{H}_{t-1}\big)$

1. *Choose N*
2. $\hat{\alpha}_{\bullet;t-1} = 0$
3. $(\Omega^1, \cdots \Omega^N)$= SAMPLE$\big(\mathcal{H}_{t-1}, N\big)$ *(Algorithm 10)*
4. *FOR l=1:N*
5. $(Q_\bullet)$= EVAL_WEIGHT$\big(\Omega^l\big)$ *(Algorithm 3)*
6. $\hat{\alpha}_{\bullet;t-1} = \hat{\alpha}_{\bullet;t-1} + \frac{1}{N} Q_\bullet$
7. *END FOR*
8. $\mathcal{Z}_{\bullet;t-1} = \ln\left(\hat{\alpha}_{\bullet;t-1}\right)$

Pre-computation

For simplifying the algorithms, let us design a special algorithm for computing the MC weights υ_l (defined in (7.9)).

Algorithm 12 (MC weights) (υ)= MC_WEIGHTS$\big(\Omega^1, \cdots, \Omega^N, w_{\bullet;t}, \hat{\alpha}_{\bullet;t-1}\big)$

1. *FOR* $l = 1 : N$
2. $(Q_\bullet)$= EVAL_WEIGHT$\big(\Omega^l\big)$ *(Algorithm 3)*
3. $\upsilon_l = \sum_{c=1}^{\hat{c}} \frac{w_c Q_c}{\hat{\alpha}_{c;t-1}}$
4. *END FOR*

Optimization of Cwf Statistics

We are now also able to design an algorithm WEIGHTUPDT for approximate update of cwf statistics. Before specifying the algorithm, let us recall the structure of Ω and $\mathcal{H}_t$.

$$
\begin{array}{lll}
\forall l \in \{1, \cdots, N\} & \Omega^l \equiv (\Omega_1^l, \cdots \Omega_n^l) & \mathcal{H}_t \equiv (\mathcal{H}_{1;t}, \cdots, \mathcal{H}_{n;t}) \\
\forall l \in \{1, \cdots, N\}, \quad \forall k \in GA: & \Omega_k^l \equiv {}^{\lfloor\alpha}\theta_k^l, & \mathcal{H}_{k;t-1} \equiv (M_{k;t-1}, R_{k;t-1}) \\
\forall l \in \{1, \cdots, N\}, \quad \forall k \in GI: & \Omega_k^l \equiv ({}^{\lfloor\alpha}\theta_k^l, {}^{\lfloor\alpha}r_k^l), & \mathcal{H}_{k;t-1} \equiv ({}^{\lfloor\alpha}C_{k;t-1}, {}^{\lfloor\alpha}\hat{\theta}_{k;t-1}, \\
 & & \quad {}^{\lfloor\alpha\lfloor d}D_{k;t-1}, {}^{\lfloor\alpha}\nu_{k;t-1})
\end{array}
$$

The main algorithm for optimization of cwf statistics reads:

Algorithm 13 (Optimization of cwf statistics) $\mathcal{H}_t$=*WEIGHTUPD*$(\mathcal{H}_{t-1}, w_t)$

1. $(\Omega^1, \cdots \Omega^N)$= SAMPLE$\big(\mathcal{H}_{t-1}, N\big)$ *(Algorithm 10, page 59)*
2. $(\hat{\alpha}_{\bullet;t-1})$= WEIGHT_NORM$\big(\mathcal{H}_{t-1}\big)$ *(Algorithm 8, page 54)*
3. $(\upsilon_\bullet)$= MC_WEIGHTS$\big(\Omega^1, \cdots \Omega^N, w_{\bullet;t}, \hat{\alpha}_{\bullet;t-1)}\big)$ *(Algorithm 12, page 60)*
4. *FOR* $k \in GA$*:*
   $(M_{k;t}, R_{k;t})$= GAUSSUPD$\big({}^{\lfloor\alpha}\theta_k^1, \cdots {}^{\lfloor\alpha}\theta_k^N, \upsilon_\bullet\big)$ *(Algorithm 14, page 61)*

5. *FOR $k \in GI$:*
   $(\ {}^{\lfloor\alpha}C_{k;t},\ {}^{\lfloor\alpha}\hat{\theta}_{k;t},\ {}^{\lfloor\alpha\lfloor d}D_{k;t},\ {}^{\lfloor\alpha}\nu_{k;t}) = \mathrm{GIWUPD}\big((\ {}^{\lfloor\alpha}\theta_k^1,\ {}^{\lfloor\alpha}r_k^1), \cdots (\ {}^{\lfloor\alpha}\theta_k^N,\ {}^{\lfloor\alpha}r_k^N), \upsilon_\bullet\big)$
   (Algorithm 15, page 61)

Algorithm 14 (Gaussian updating) $(M_{k;t}, R_{k;t}) = \mathrm{GAUSSUPD}\big(\ {}^{\lfloor\alpha}\theta_k^1, \cdots\ {}^{\lfloor\alpha}\theta_k^N, \upsilon\big)$

1. $M_{k;t} = \sum_{l=1}^{N} \upsilon_l\ {}^{\lfloor\alpha}\theta_k^l$
2. $R_{k;t} = \sum_{l=1}^{N} \upsilon_l\ {}^{\lfloor\alpha}\theta_k^l\ {}^{\lfloor\alpha}\theta_k^{l'} - M_k M_k'$

Algorithm 15 (GiW updating)
$(\ {}^{\lfloor\alpha}C_{k;t},\ {}^{\lfloor\alpha}\hat{\theta}_{k;t},\ {}^{\lfloor\alpha\lfloor d}D_{k;t},\ {}^{\lfloor\alpha}\nu_{k;t}) = \mathrm{GIWUPD}\big((\ {}^{\lfloor\alpha}\theta_k^1,\ {}^{\lfloor\alpha}r_k^1), \cdots (\ {}^{\lfloor\alpha}\theta_k^N,\ {}^{\lfloor\alpha}r_k^N), \upsilon\big)$

1. $p_k = \sum_{l=1}^{N} \frac{1}{{}^{\lfloor\alpha}r_k^l} \upsilon_l$
2. $s_k = \sum_{l=1}^{N} \ln\big({}^{\lfloor\alpha}r_k^l\big)\, \upsilon_l$
3. ${}^{\lfloor\alpha}\hat{\theta}_{k;t} = \frac{1}{p_k} \sum_{l=1}^{N} \frac{{}^{\lfloor\alpha}\theta_k^l}{{}^{\lfloor\alpha}r_k^l} \upsilon_l$
4. ${}^{\lfloor\alpha}C_{k;t} = \sum_{l=1}^{N} \frac{{}^{\lfloor\alpha}\theta_k^l\ {}^{\lfloor\alpha}\theta_k^{l'}}{{}^{\lfloor\alpha}r_k^l} \upsilon_l - p_k\ {}^{\lfloor\alpha}\hat{\theta}_{k;t}\ {}^{\lfloor\alpha}\hat{\theta}_{k;t}'$
5. $(\ {}^{\lfloor\alpha}\nu_{k;t}) = \mathrm{GETNU}\big(\ln(p_k) + s_k\big)$ *(Algorithm 19, page 96)*
6. ${}^{\lfloor d}D_{k;t} = \frac{{}^{\lfloor\alpha}\nu_{k;t}}{p_k}$

Remarks 8

- *The software realization of the algorithms can be done in a bit more clever way. For example, the weight estimate $\hat{\alpha}_{t-1}$ need not to be computed twice.*
- *Evaluation of matrices ${}^{\lfloor\alpha}C_k$ should be of course realized in L'DL decomposition (Section C.5.2).*
- *For achieving of feasibility, effective stopping rules [44] should be designed so that the number of simulated samples needed for each step is minimized.*

7.2.6 Specific Forms of Component Weighting Functions

Switching Weight

In the case when ϕ_{t-1} is scalar and $\mathring{c} = 2$, we can use this type of cwf. It has low practical applicability. It illustrates the derived relations and serves for checking the Monte-Carlo evaluation, because the result can be found analytically here.

$$\alpha_1(\phi_{t-1}|\Omega) = \left\{ \begin{array}{ll} 1 & \phi_{t-1} > \Omega \\ 0 & \phi_{t-1} \leq \Omega \end{array} \right. ,\ \alpha_2(\phi_{t-1}|\Omega) = \left\{ \begin{array}{ll} 0 & \phi_{t-1} > \Omega \\ 1 & \phi_{t-1} \leq \Omega \end{array} \right.$$

The cwf parameter Ω is scalar in this case. The posterior pdf on this parameter can be chosen as Gaussian pdf with mean M_t and variance R_t.

$$\rho(\Omega|\mathcal{H}_t) \equiv \mathcal{N}_\Omega\left(M_t, R_t\right),\ \mathcal{H}_t \equiv (M_t, R_t)$$

For this case, we are able to evaluate the weight estimate more or less analytically.

$$\begin{aligned} \hat{\alpha}_{1;t-1} &= \int \alpha_1(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})d\Omega = \int_{-\infty}^{\phi_{t-1}} \mathcal{N}_\Omega\left(M_{t-1}, R_{t-1}\right) d\Omega = \mathcal{J}(M_{t-1}, R_{t-1}, -\infty, \phi_{t-1}) \\ \hat{\alpha}_{2;t-1} &= 1 - \hat{\alpha}_{1;t-1}, \end{aligned}$$

where $\mathcal{J}(\mu, R, a, b)$ is normalization integral of so called Truncated Gaussian Distribution (C.3). The updated pdfs $\rho_1^U(\Omega|\mathcal{H}_{1;t-1}^U)$ and $\rho_2^U(\Omega|\mathcal{H}_{2;t-1}^U)$ are itself Truncated Gaussian distributions:

$$\begin{aligned}\rho_1^U(\Omega|\mathcal{H}_{1;t-1}^U) &= \frac{\alpha_1(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})}{\hat{\alpha}_{1;t}} = \mathcal{TN}_\Omega\left(M_{t-1}, R_{t-1}, -\infty, \phi_{t-1}\right)\\ \rho_2^U(\Omega|\mathcal{H}_{2;t-1}^U) &= \frac{\alpha_2(\phi_{t-1}|\Omega)\rho(\Omega|\mathcal{H}_{t-1})}{\hat{\alpha}_{2;t}} = \mathcal{TN}_\Omega\left(M_{t-1}, R_{t-1}, \phi_{t-1}, \infty\right)\end{aligned}$$

The function $h(\Omega)$ defined in (7.6) (page 57) reads:

$$\begin{aligned}h(\Omega) &\equiv \sum_{c=1}^{\mathring{c}} w_{c;t}\rho_c^U(\Omega|\mathcal{H}_{c;t-1}^U) =\\ &= w_{1;t}\mathcal{TN}_\Omega\left(M_{t-1}, R_{t-1}, -\infty, \phi_{t-1}\right) + w_{2;t}\mathcal{TN}_\Omega\left(M_{t-1}, R_{t-1}, \phi_{t-1}, \infty\right).\end{aligned}$$

According to the results presented in Section 7.2.4, the new values of statistics M_t and R_t can be evaluated as follows:

$$\begin{aligned}M_t &= \mathcal{E}\left[\Omega\right]_h = w_{1;t}\mathcal{E}\left[\Omega\right]_{\rho_1^U} + w_{2;t}\mathcal{E}\left[\Omega\right]_{\rho_2^U}\\ R_t &= \mathbf{cov}\left[\Omega\right]_h = w_{1;t}\mathbf{cov}\left[\Omega\right]_{\rho_1^U} + w_{2;t}\mathbf{cov}\left[\Omega\right]_{\rho_2^U} + w_{1;t}w_{2;t}\left(\mathcal{E}\left[\Omega\right]_{\rho_1^U} - \mathcal{E}\left[\Omega\right]_{\rho_2^U}\right)^2.\end{aligned}$$

We need to compute mean values, variances and normalizing integral of Truncated Gaussian Distribution. This will be done with algorithms *TRUNCSTAT* (Algorithm 21) and *TRUNCNORM* (Algorithm 20). Using these algorithms, we can formulate the algorithms for weight update *WEIGHTNORM* and *WEIGHTUPDT*.

Algorithm 16 (Switching-weight normalizing) $(\mathcal{Z}_{\bullet,t-1})=$ WEIGHTNORM$\left(M_{t-1}, R_{t-1}\right)$

1. $(\hat{\alpha}_{1;t-1})=$ TRUNCNORM$\left(M_{t-1}, R_{t-1}, -\infty, \phi_{t-1}\right)$ *(Algorithm 20, page 100)*
2. $\hat{\alpha}_{2;t-1} = 1 - \hat{\alpha}_{1;t-1}$
3. $\mathcal{Z}_{\bullet,t-1} = \ln\left(\hat{\alpha}_{\bullet,t-1}\right)$

Algorithm 17 (Switching-weight Updating) $(M_t, R_t)=$ WEIGHTUPDT$\left(M_{t-1}, R_{t-1}, w_t\right)$

1. $(E_1, C_1)=$ TRUNCSTAT$\left(M_{t-1}, R_{t-1}, -\infty, \phi_{t-1}\right)$ *(Algorithm 21, page 100)*
2. $(E_2, C_2)=$ TRUNCSTAT$\left(M_{t-1}, R_{t-1}, \phi_{t-1}, +\infty\right)$
3. $M_t = w_{1;t}E_1 + w_{2;t}E_2$
4. $R_t = w_{1;t}C_1 + w_{2;t}C_2 + w_{1;t}w_{2;t}(E_1 - E_2)^2$

Example 15 (Updating of truncated Gaussian distribution) *Let us suppose the following case:*

$$\phi_{t-1} = 3, M_{t-1} = 2, R_{t-1} = 1, w = [0.75, 0.25].$$

Old posterior pdf on cwf parameter Ω is Gaussian pdf. Its updates $\rho_1^U(\Omega|\mathcal{H}_{1;t-1}^U)$ and $\rho_2^U(\Omega|\mathcal{H}_{2;t-1}^U)$ are truncated normal distributions. Function $h(\Omega)$ is mixture of the updates. Figure 7.1 shows all involved pdfs in details.

Remarks 9 *It is not possible to generalize this type of cwf to multiple component case, because it doesn't allow permutation of components during estimation and hence it is very sensitive on initial conditions.*

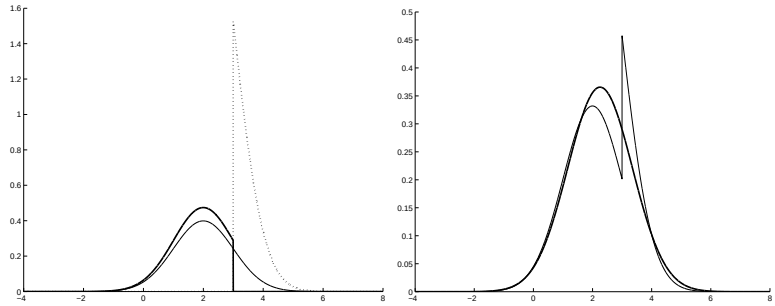


Figure 7.1: Updating of truncated gaussian distribution
The left part shows old posterior pdf $\rho(\Omega|\mathcal{H}_{t-1})$(thin) and updates $\rho_1^U(\Omega|\mathcal{H}_{1;t}^U)$(dotted) and $\rho_2^U(\Omega|\mathcal{H}_{2;t}^U)$(thick). The right part shows how the result of optimization $\rho(\Omega|\mathcal{H}_t)$ (thick) approximates the pdf $h(\Omega)$ (thin).

Gaussian Ratio

We have to define more general cwfs than the specified ones. In general, it suffice to select $\mathring{c}$ nonnegative functions $g_c(\phi_{t-1}|\Omega_c)$, each parameterized by own parameter Ω_c. Then the cwfs can be defined as

$$\alpha_c(\phi_{t-1}|\Omega) \equiv \frac{g_c(\phi_{t-1}|\Omega_c)}{\sum_{c=1}^{\mathring{c}} g_c(\phi_{t-1}|\Omega_c)},\ \Omega \equiv (\Omega_1, \cdots, \Omega_c),$$

which guarantees that $\sum_{c=1}^{\mathring{c}} \alpha_c(\phi_{t-1}|\Omega) = 1,\ \forall\phi_{t-1},\ \forall\Omega$.

We will deal with $g_c(\phi_{t-1}|\Omega_c)$ defined as a value of factorized multivariate Gaussian distribution. (This approach has a good justification. See [10].) The factorization is performed in the same way as in Section 4.1. It is usual to denote the factorized elements with two indices, but the theory presented in this chapter indexes the parts of Ω and related pdfs with only one index. We will face this problem by defining operator $<>$, which uniquely converts two indexes into one:

$$< oc >= (o-1) \times \mathring{c} + c.$$

Using the mentioned notation, we can define function $g_c(\phi_{t-1}|\Omega_c)$ Analogical to factors defined in Section 4.1:

$$g_c(\phi_{t-1}|\Omega_c) = \prod_{i=1}^{\mathring{\phi}} \mathcal{N}_{\phi_{i;t-1}}\left({}^{\lfloor\alpha}\theta'_{<ic>}\ {}^{\lfloor\alpha}\psi_{<ic>;t-1},\ {}^{\lfloor\alpha}r_{<ic>}\right),\ \text{where}\ {}^{\lfloor\alpha}\psi_{<ic>;t-1}$$

is a subvector of vector $[\phi_{i+1,\cdots,\mathring{\phi};t-1}, 1]$ and $\Omega_c \equiv \{({}^{\lfloor\alpha}\theta_{<ic>},\ {}^{\lfloor\alpha}r_{<ic>})|i \in \{1, \cdots, \mathring{\phi}\}\}$. Hence, the cwfs are defined as follows::

$$\begin{aligned}\alpha_c(\phi_{t-1}|\Omega) &= \frac{\prod_{i=1}^{\mathring{\phi}} \mathcal{N}_{\phi_{i;t-1}}\left({}^{\lfloor\alpha}\theta'_{<ic>}\ {}^{\lfloor\alpha}\psi_{<ic>;t},\ {}^{\lfloor\alpha}r_{<ic>}\right)}{\sum_{\tilde{c}=1}^{\mathring{c}} \prod_{i=1}^{\mathring{\phi}} \mathcal{N}_{\phi_{i;t-1}}\left({}^{\lfloor\alpha}\theta'_{<i\tilde{c}>}\ {}^{\lfloor\alpha}\psi_{<ic>;t},\ {}^{\lfloor\alpha}r_{<ic>}\right)},\\ \Omega &\equiv \{{}^{\lfloor\alpha}\theta_k,\ {}^{\lfloor\alpha}r_k|k \in \{1, \cdots, \mathring{c} \times \mathring{\phi}\}\}.\end{aligned}$$

If we want to use the numeric approximations derived in Section 7.2, we have only to design specific version of algorithm *EVAL_WEIGHT*.

Algorithm 18 (Cwf evaluation) $(Q_\bullet)$= EVAL_WEIGHT$\left(\Omega^l\right)$

1. *For each component c, evaluate* $l_c = \sum_{i=1}^{\mathring{\phi}} \left(-\frac{\ln\left({}^{\lfloor\alpha}r_{<ic>}\right)}{2} - \frac{({}^{\lfloor\alpha}\theta'_{<ic>}\psi_{<ic>;t} - \phi_{i;t-1})^2}{2\,{}^{\lfloor\alpha}r_{<ic>}} \right)$

2. $l_\bullet = \exp\left(l_\bullet - max(l_\bullet)\right)$
3. $Q_\bullet = \frac{l_\bullet}{sum(l_\bullet)}$

Remarks 10 *Examples of this cwf type are plotted in Section 8.2.2 and 8.2.3.*

Chapter 8

Experiments

This chapter illustrates the developed theory on several examples. Mostly, it shows evolution of the estimates over time to demonstrate the algorithms behavior. In section dealing with constant-weights mixtures, the PB algorithm is compared with classical QB algorithm (Appendix A).

8.1 Gaussian Mixtures with Constant Weights

This section deals with normal factors (Section 6.1) and constant component weighting functions (Section 7.1). First, the behavior of the algorithm is demonstrated on simple examples. Then, the comparison of the PB and QB algorithms is performed.

8.1.1 The Simplest Case

Model

Let us have a 2-component static mixture defined on scalar data. For a better readability, the index denoting the data channel is omitted here.(It is 1 in all cases.)

$$\begin{array}{rcll} \mathring{d} & = & 1 & \text{(data are scalar)} \\ \mathring{c} & = & 2 & \text{(2 components)} \\ \phi_{t-1} & \equiv & (1) & \text{(system is static)} \\ \Omega & \equiv & (\alpha_1,\alpha_2), \alpha_i > 0, \sum_{i=1}^{2} \alpha_i = 1 & \text{(parameter of cwfs)} \\ \Theta & \equiv & (\theta_1,\theta_2,r_1,r_2,\alpha_1,\alpha_2) & \text{(mixture parameter)} \end{array}$$

$$\begin{array}{rcll} \alpha_1(\phi_{t-1}|\Omega) & \equiv & \alpha_1(1|\alpha_1,\alpha_2) = \alpha_1 & \text{(1st cwf)} \\ \alpha_2(\phi_{t-1}|\Omega) & \equiv & \alpha_2(1|\alpha_1,\alpha_2) = \alpha_2 & \text{(2nd cwf)} \\ f_1(d_t|\phi_{t-1},\Theta_1) & \equiv & f_1(d_t|\Theta_1) = \mathcal{N}_{d_t}(\theta_1,r_1) & \text{(1st component)} \\ f_2(d_t|\phi_{t-1},\Theta_2) & \equiv & f_2(d_t|\Theta_2) = \mathcal{N}_{d_t}(\theta_2,r_2) & \text{(2nd component)} \\ f(d_t|\phi_{t-1},\Theta) & \equiv & \alpha_1\mathcal{N}_{d_t}(\theta_1,r_1) + \alpha_2\mathcal{N}_{d_t}(\theta_2,r_2) & \text{(mixture)} \end{array}$$

Form of Prior and Posterior pdf

$$\begin{array}{rcl} \rho(\Omega|\mathcal{H}_t) & \equiv & \rho(\alpha_1,\alpha_2|\kappa_{1;t},\kappa_{2;t}) = Di_{\alpha_1,\alpha_2}(\kappa_{1;t},\kappa_{2;t}) \\ \pi_1(\Theta_1|\mathcal{S}_{1;t}) & \equiv & \pi_1(\theta_1,r_1|V_{1;t},\nu_{1;t}) = GiW_{\theta_1,r_1}(V_{1;t},\nu_{1;t}) \\ \pi_2(\Theta_2|\mathcal{S}_{2;t}) & \equiv & \pi_2(\theta_2,r_2|V_{2;t},\nu_{2;t}) = GiW_{\theta_2,r_2}(V_{2;t},\nu_{2;t}) \\ \mathcal{H}_t & \equiv & (\kappa_{1;t},\kappa_{2;t}),\ \mathcal{S}_{1;t} \equiv (V_{1;t},\nu_{1;t}),\ \mathcal{S}_{2;t} \equiv (V_{2;t},\nu_{2;t}) \\ \mathcal{G}_t & \equiv & (\kappa_{1;t},\kappa_{2;t},V_{1;t},\nu_{1;t},V_{2;t},\nu_{2;t}) \\ \pi(\Theta|\mathcal{G}_t) & \equiv & \pi(\theta_1,\theta_2,r_1,r_2,\alpha_1,\alpha_2|\kappa_{1;t},\kappa_{2;t},V_{1;t},\nu_{1;t},V_{2;t},\nu_{2;t}) \equiv \\ & \equiv & GiW_{\theta_1,r_1}(V_{1;t},\nu_{1;t})GiW_{\theta_2,r_2}(V_{2;t},\nu_{2;t})Di_{\alpha_1,\alpha_2}(\kappa_{1;t},\kappa_{2;t}) \end{array}$$

Posterior pdf on $\Omega \equiv (\alpha_1, \alpha_2)$ was chosen as Dirichlet pdf. Posterior pdf on factor parameters was selected as GiW pdfs. The overall posterior pdf is thus product of two GiW pdfs and one Dirichlet pdf. The posterior statistic $\mathcal{G}_t$ is formed with statistics of Dirichlet pdf and GiW pdfs. Of course, equivalent representations of GiW statistics V are considered. (See Agreement 7)

The True Value of Parameter and the Initial Statistics

$$\begin{aligned}
\Theta_{true} &\equiv (\theta_{1true} \equiv 2.5,\ \theta_{2true} \equiv 1,\ r_{1true} \equiv 0.005,\ r_{2true} \equiv 0.001,\\
&\quad \alpha_{1true} \equiv 0.3333,\ \alpha_{2true} \equiv 0.6666)\\
\mathcal{G}_0 &\equiv (\kappa_{1;0} \equiv 6,\ \kappa_{2;0} \equiv 6,\\
&\quad C_{1;0} \equiv 1000,\ \hat{\theta}_{1;0} \equiv 0.0401,\quad {}^{\lfloor d}D_{1;0} \equiv 0.022,\ \nu_{1;0} \equiv 4.20,\\
&\quad C_{2;0} \equiv 1000,\ \hat{\theta}_{2;0} \equiv -0.6209,\ {}^{\lfloor d}D_{2;0} \equiv 0.022,\ \nu_{2;0} \equiv 4.20)
\end{aligned}$$

The true system model $f(d_t|\Theta = \Theta_{true})$, and initial point estimate $f(d_t|\Theta = \hat{\Theta}_0), \hat{\Theta}_0 = \mathcal{E}\left[\Theta\right]_{\pi(\Theta|\mathcal{G}_0)}$ are depicted on Figure 8.1.

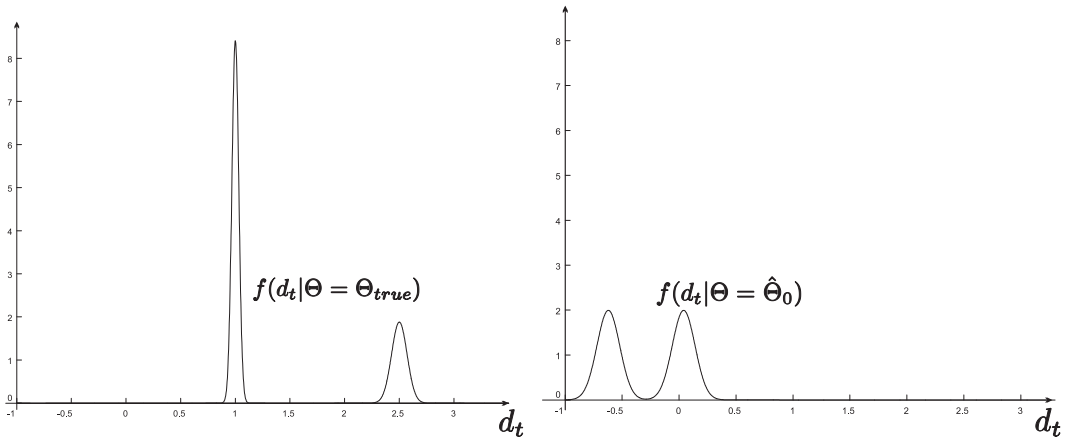


Figure 8.1: The true system model and initial mixture

The left hand part of this figure shows the true system model $f(d_t|\Theta = \Theta_{true})$. It is a scalar 2-component static Gaussian mixture. The right part shows point estimate of the system $f(d_t|\Theta = \hat{\Theta}_0)$ based on the prior pdf given by the statistic $\mathcal{G}_0$. It can be seen that this initial point estimate is completely different then the true system model.

Processing

We simulated 60 data records generated by the true system and estimated their model using PB algorithm. The simulated data are depicted on Figure 8.2.

We want to show behavior of PB algorithm in details, hence evolutions of important statistic during estimation are displayed. The most important statistics $\hat{\theta}_{1;t},\ \hat{\theta}_{2;t},\ C_{1;t},\ C_{2;t}$ are depicted on Figure 8.3. (They are scalars in this case.) Because the statistic $\hat{\theta}_{c;t}$ represents a point estimate of θ_c ($\hat{\theta}_{c;t} = \mathcal{E}\left[\theta_c|\,\hat{\theta}_{c;t}\right]$), we can simply observe the quality of the estimation. According to Proposition 31, the covariance $\mathbf{cov}\left[\theta_c|\,\nu_{c;t},\ {}^{\lfloor d}D_{c;t}, C_{c;t}\right] = \hat{r}_{c;t}C_{c;t}$. It means that covariance of the point estimate is direct proportional to the value of statistic $C_{c;t}$.

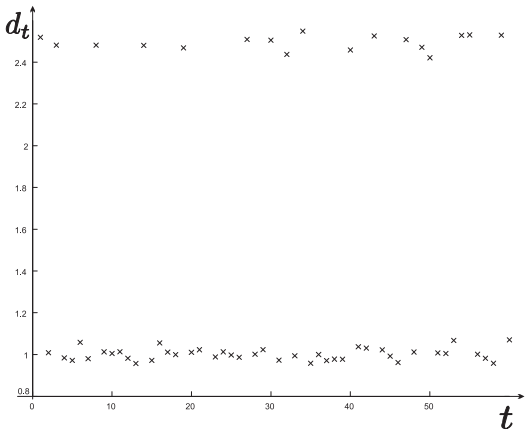


Figure 8.2: Simulated data

The figure shows 60 data records generated by the true system model $f(d_t|\Theta = \Theta_{true})$. According to the form of the system model (see Figure 8.1), it is clear that data must be concentrated in regions near by $\theta_{1true} \equiv 2.5$ and $\theta_{2true} \equiv 1$.

Also the evolution of statistics $\nu_{c;t}$ and ${}^{\lfloor d}D_{c;t}$ should be displayed. Instead, we display point estimates $\hat{r}_{c;t}$ of r_c and variance $s_{c;t}$ of this estimate. According to Proposition 31,

$$\hat{r}_{c;t} \equiv \mathcal{E}\left[r_c|\,\nu_{c;t},\,{}^{\lfloor d}D_{c;t}\right] = \frac{{}^{\lfloor d}D_{c;t}}{\nu_{c;t}-2},\; s_{c;t} = \mathbf{cov}\left[r_c|\,\nu_{c;t},\,{}^{\lfloor d}D_{c;t}\right] = \frac{\hat{r}^2_{c;t}}{\nu_{c;t}-4}.$$

Evolution of statistics $\hat{r}_{c;t}$ and $s_{c;t}$ can be seen on Figure 8.4.

Figure 8.5 shows, how the point estimates of the component weights $\hat{\alpha}_{c;t}$ evolve during estimation.

Another significant indicator of the estimation quality is the difference from the correct Bayesian estimation. Of course, we are not able to perform correct Bayesian estimation of a mixture model, unless we know the relation of each data record to the component it was generated from. This is possible for simulated systems. We can simply remember active components during the simulation and then confront this information with the weights $w_{c;t}$ from PB algorithm. It is obvious (See Remarks 2) that Bayesian estimation can be formulated as PB estimation with $w_{\bullet;t}$ having the only one nonzero element on the position which corresponds to the component being active in time t. We call such weight as Bayesian weight. Of course, the numbering of components in estimated mixture need not be the same as the numbering in simulated mixture, hence we may need to permute the Bayesian weights to be comparable with the PB weights. Let us denote the permuted Bayesian weights as $w_{B\bullet;t}$. Then the quality of estimating each particular component during the time can be measured as $Q_{c;t} = \mathrm{abs}(w_{c;t} - w_{Bc;t})$. It is clear that in ideal case $Q_{c;t}$ is zero for all c, t. It is also clear that in our case of two component mixture, $Q_{1;t} = Q_{2;t}\ \forall t$. Hence it suffice to display $Q_{1;t}$ only.

The QB algorithm (Appendix A) uses the weights $w_{c;t}$ analogically to the PB algorithm. Hence we can define QB quality indicator $\mathcal{Q}_{c;t}$ as analogy to $Q_{c;t}$. Evolution of $Q_{1;t}$ and $\mathcal{Q}_{1;t}$ during the estimation is depicted on Figure 8.5.

Resulting point estimate of the mixture parameters obtained using PB estimation and resulting point estimate obtained using QB estimation are depicted on Figure 8.6.

Conclusions

The presented example shows that PB algorithm behaves reasonably. On this simple example it gives a very good result, better than the result of QB algorithm.

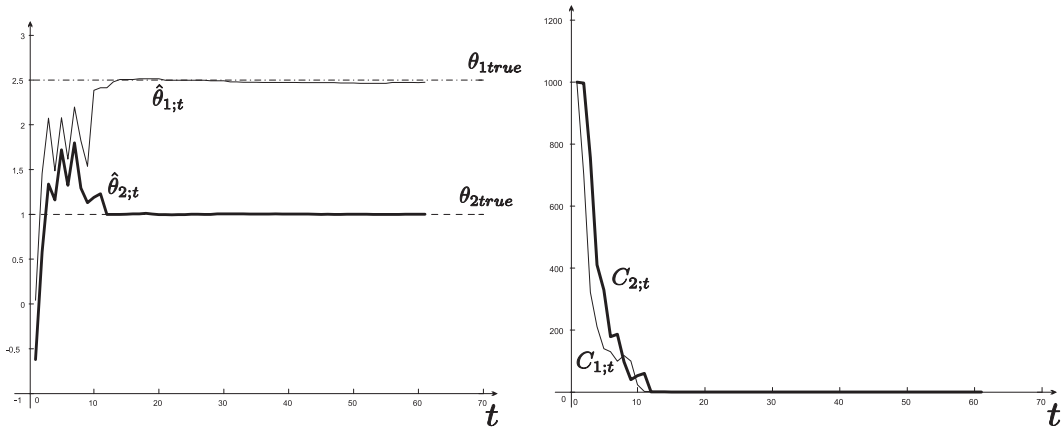


Figure 8.3: Evolution of statistics $\hat{\theta}_{c;t}$ and $C_{c;t}$

The left hand part of this figure shows how the point estimates of factor means $\hat{\theta}_{1;t}$, $\hat{\theta}_{2;t}$ approach the true values $\theta_{1true}, \theta_{2true}$. It can be seen that after processing approximately 16 data records, the point estimates started to be almost perfect. The right part of this figure shows evolution of statistics $C_{1;t}$, $C_{2;t}$. Because covariance of point estimates $\hat{\theta}_{c;t}$ depends proportionally on $C_{c;t}$, the decreasing trends of $C_{1;t}, C_{2;t}$ indicates increasing quality of the point estimate.

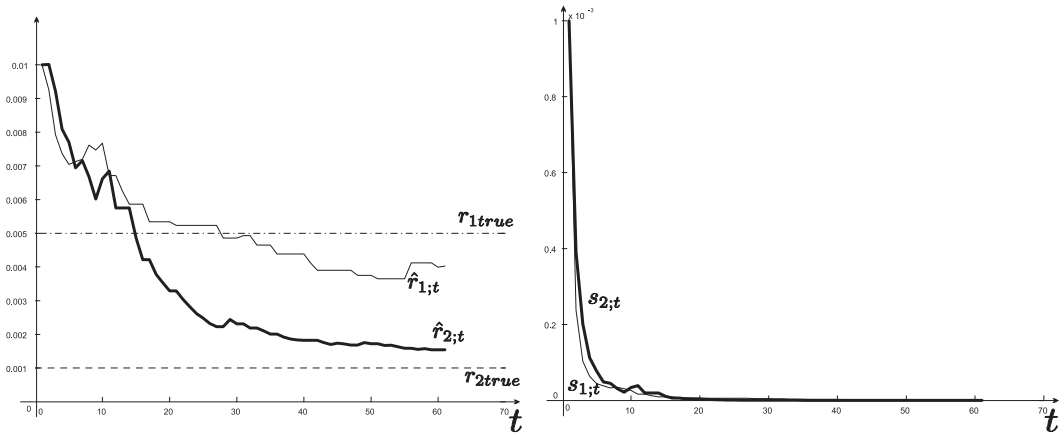


Figure 8.4: Evolution of point estimates of factor variances $r_{c;t}$

The left hand part of this figure shows how the point estimates of factor variances $\hat{r}_{1;t}$, $\hat{r}_{2;t}$ approach the true values r_{1true}, r_{2true}. It can be seen that estimating the factor variance is more complex problem than estimating the means, but it can be seen that the estimates are slowly approaching the true values. The right hand part of this figure shows evolution of variances of point estimates $\hat{r}_{c;t}$, which are quickly decreasing.

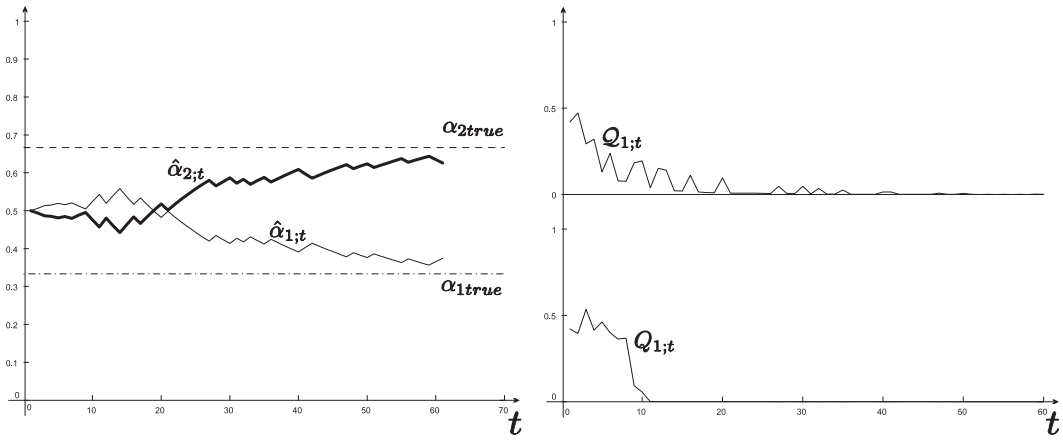


Figure 8.5: Evolution of point estimate of component weights κ

The left hand part of this figure shows how the point estimates of component weights $\hat{\alpha}_{1;t}$, $\hat{\alpha}_{2;t}$ approach the true values $\alpha_{1true}, \alpha_{2true}$. The right hand part of this figure shows evolution of quality indicators $Q_{1;t}$ determining the quality of PB estimation and $\mathcal{Q}_{1;t}$ determining the quality of QB estimation. It can be seen that after some time both indicators $Q_{1;t}$ and $\mathcal{Q}_{1;t}$ approach zero. This means that after some time, both algorithms perform almost exactly as the Bayesian estimation in this case.

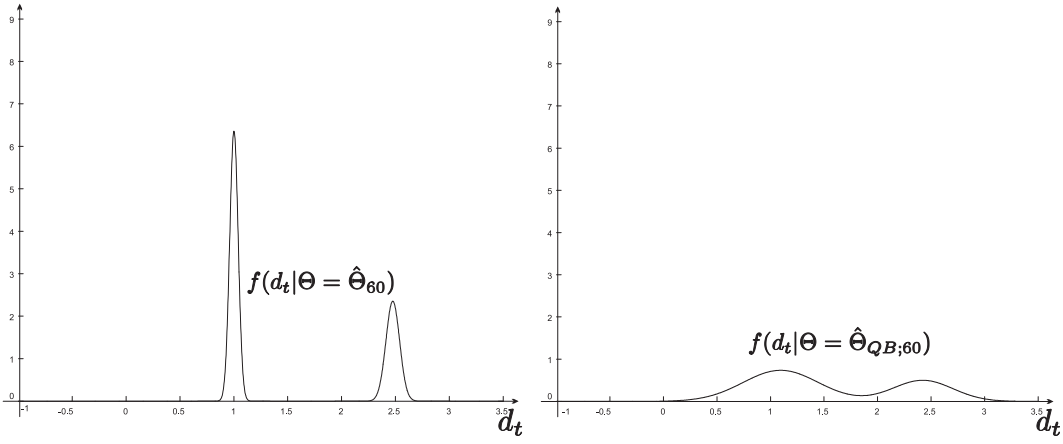


Figure 8.6: Resulting point estimates

Resulting point estimate $f(d_t|\Theta = \hat{\Theta}_{60})$ of the mixture is depicted on left hand part of this figure. Point estimate of the same system obtained through QB algorithm $f(d_t|\Theta = \hat{\Theta}_{QB;60})$ is depicted on right hand part of this figure. If we compare these results with the true system model from Figure 8.1, we can see that both algorithms estimated the parameters θ_1, θ_2 well. But the estimates of parameters r_1, r_2 determining the factors variances are much better in PB estimation.

8.1.2 Banana Shape

This example belongs to the set of classical examples for testing of mixture estimation. System is a two-dimensional static mixture with 32 components. Figure 8.7 shows the true system $f(d_t|\Theta = \Theta_{true})$ and 1500 data records generated.

We modelled this system with 20-component mixture. Initial statistics of the PB estimation was selected randomly. Figure 8.8 shows the mixture $f(d_t|\Theta = \hat{\Theta}_0)$ with the point estimate of Θ based on initial statistics. Second part of this figure shows the mixture $f(d_t|\Theta = \hat{\Theta}_{1500})$ with point estimate based on statistics obtained with PB algorithm. For comparison, Figure 8.9 shows point estimate based on QB algorithm $f(d_t|\Theta = \hat{\Theta}_{QB;1500})$.

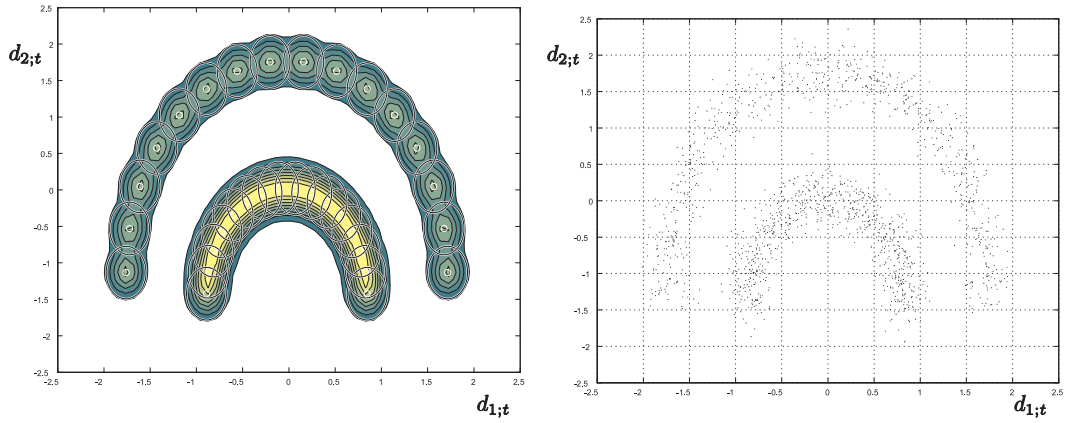


Figure 8.7: Banana shape: System and simulated data

Left hand part of this figure shows the true system $f(d_t|\Theta = \Theta_{true})$. It is a two-dimensional function and it is displayed as so called contour plot, i.e. the value in a point on the grid is given by the color of this point. The right hand part of this figure shows the data generated by the system. These data are then used for estimating the model.

Conclusions

The presented example shows estimation results with PB algorithm on a more complex example. It can be seen that again very good result was obtained. The result of QB algorithm is worse. If we would use initial statistic obtained using algorithm `mixinit` (See Appendix A) instead of random ones, even the QB algorithm would get very good result.

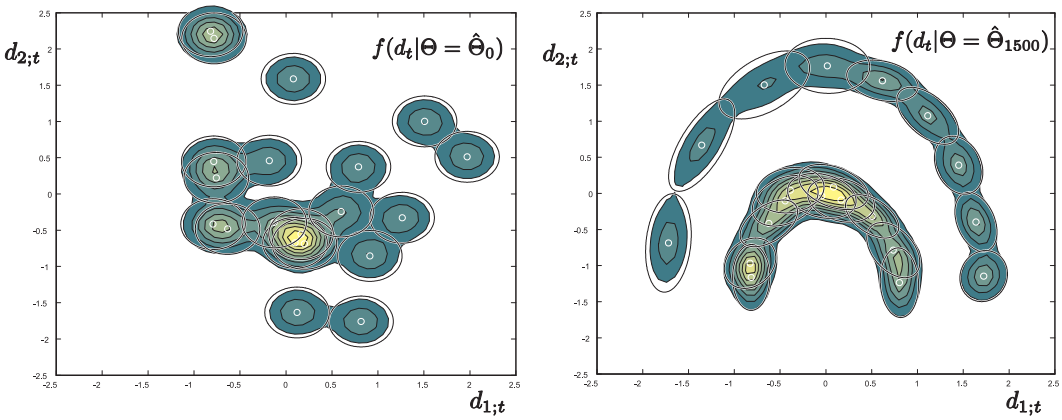


Figure 8.8: Banana shape: initial mixture and result of estimation

Left hand part of this figure shows the point estimate of the system $f(d_t|\Theta = \hat{\Theta}_0)$ based on the initial statistics. It can be seen that this initial estimate is completely different from the true system. The right hand part of this figure shows the point estimate based on statistics obtained from the PB algorithm. It can be seen that the result is similar to the true system.

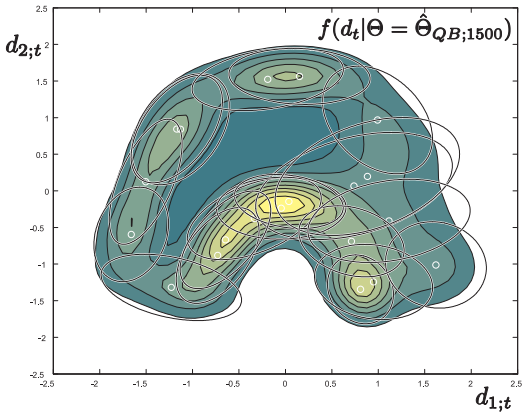


Figure 8.9: Result of QB estimation

The figure shows the point estimate of the system $f(d_t|\Theta = \hat{\Theta}_{QB;1500})$ based on the statistics obtained from the QB algorithm. It can be seen that the result is worse than the result of the PB algorithm.

8.1.3 Comparison on "Classical" Examples

Intensive tests consisting of 1396 data sets were performed. Data used for this test represent various types of systems (static, dynamic, multidimensional) and are a part of standard testing procedure of new algorithms within Mixtools system [45]. As a quality measure, we used the v-likelihood [11] of the estimated model. For each set, we evaluated a criterion h, which is the difference between the loglikelihood obtained by the PB algorithm and the QB algorithm. Thus, $h > 0$ if the PB algorithm was better. Table 8.1 shows the results. Mean value of h over all sets is 6.18. The cases where likelihood of one result is not greater than $\exp(2) \times$ likelihood of the second are taken as a draw. This leads to condition $abs(h) < 2$ on the draw cases. The overall computing time spent by this testing was approximately 20 hours.

condition	number of cases	percentage
$h > 0$	1125	80.6%
$h < 0$	271	19.4%
$\text{abs}(h) < 2$	1126	80.6%
$h > 2$	251	18.0%
$h < -2$	19	1.4%

Table 8.1: Results of experimental comparison
The table shows the number of cases fulfilling several conditions for h. Since the values with $abs(h) < 2$ are taken as a draw, we can conclude that the PB algorithm was worse than the QB algorithms in only 1.4% of cases. Without this condition, the PB algorithm improves (slightly) the QB result in 80% of cases.

8.1.4 Comparison on Randomly Generated Examples

In order to compare the PB algorithm on other than the classic examples, random generator was used to generate stable systems. We generated 198 mixtures with dimension from 1 to 20, with 2 to 10 components and with order 0 to 5. Number of data generated from each of these systems was selected randomly between 1000 and 3000 and increased by 400-multiple of the system dimension. Histograms showing the frequencies of used dimensions, orders etc. are displayed on Figures 8.10 and 8.11.

Initial estimate and model structure was obtained using the algorithm `mixinit`. (See Appendix A.) Since the `mixinit` algorithm is based on repetitive using of mixture estimation, we can speak about QB and PB variant of `mixinit`. Hence we tested two versions:

- QB variant of `mixinit`, QB variant of mixture estimation.
- PB variant of `mixinit`, PB variant of mixture estimation.

Results of estimation were processed in the same way as in previous section. The table 8.2 shows them. Mean value of h over all sets is 36716.5454. The overall computing time spent by this testing was approximately 12 days.

condition	number of sets	percentage
$h > 0$	328	98.8%
$h < 0$	4	1.2%
$\text{abs}(h) < 2$	2	0.6%
$h > 2$	327	98.5%
$h < -2$	3	0.9%

Table 8.2: Results of experimental comparison with random systems
The table shows the number of cases fulfilling several conditions for h. Since values with $abs(h) < 2$ are taken as a draw, we can conclude that PB algorithm was worse than QB algorithm in approximately 1% of cases and was better in 98.5% cases.

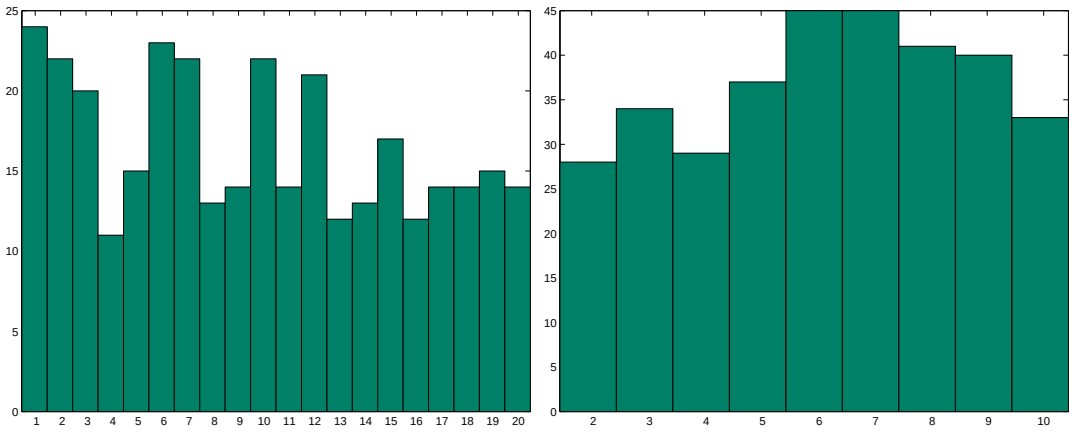


Figure 8.10: Histograms of systems characteristics
The left hand part of this figure shows histogram of dimensions of generated systems. The right hand part shows histogram of components numbers.

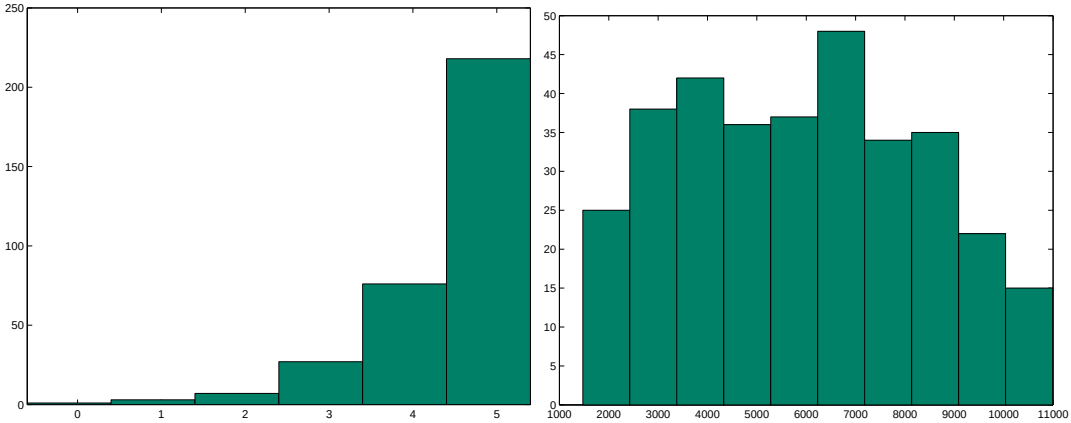


Figure 8.11: Result of QB estimation
The left hand part of this figure shows histogram of orders of generated systems. The right hand part shows histogram of the numbers of data. Note that system order is defined as the maximal order of all its parts. Orders of particular components were selected uniformly from {0,1,2,3,4,5}.

8.1.5 Comparison on Cluster-Analysis Examples

In order to be able to compare our algorithm with a plethora of others, we apply it in the field of cluster analysis. Cluster analysis can be viewed as estimation of static mixture on features-space and then predicting the value of the cluster label. The following text describes the mixture-based clustering in detail.

The mixtures can be used for clustering tasks in the following way:

1. Include the class label into the data records as its last item $d_{\mathring{d};t}$.
2. Choose structure of static mixture $f(d_t|\Theta)$ and construct initial estimate $\pi(\Theta|\mathcal{G}_0)$.
3. Estimate static mixture $f(d_t|\Theta)$, i.e obtain $\pi(\Theta|\mathcal{G}_{\mathring{t}})$.
4. Construct the predictive pdf $f(d_t) = \int f(d_t|\Theta)\pi(\Theta|\mathcal{G}_t)d\Theta$.
5. Construct the conditional pdf $f(d_{\mathring{d};t}|d_{1;t}\cdots d_{\mathring{d}-1;t})$.

The resulting pdf is our classifier. Knowing the values of features $d_{1;t}\cdots d_{\mathring{d}-1;t}$ it gives distribution on the class labels $f(d_{\mathring{d};t})$. As a class label we can take a label with the highest probability.

Remarks 11

- *In fact, the class label need not be on the last position of d_t. It can be placed on arbitrary position. Naturally, the resulting classifier must be pdf on the class label determined by the other channels.*
- *Because we are not able to model efficiently dependency of discrete data on continuous data, the discrete data are modelled as continuous ones. The resulting class label is then selected as the mean value of the pdf $f(d_{\mathring{d};t})$ rounded to the nearest discrete value of class label.*
- *The step 2 can of course significantly influence the clustering quality. The structure must be rich enough, but it must not be richer than the number of training samples allows to estimate. The algorithm* `mixinit` *(See Appendix A) solves this problem. Its result is both, the mixture structure and the initial estimate. The algorithm* `mixinit` *performs mixture estimation as its subtask. Hence we have two variants of* `mixinit`*:* `mixinit` *with PB and* `mixinit` *with QB.*
- *In the tests performed, we distinguished two variants of classifiers:*

  Mix PB *Step 2 performed using PB variant of* `mixinit`*, step 3 performed using PB estimation.*

  Mix QB *Step 2 performed using QB variant of* `mixinit`*, step 3 performed using QB estimation.*

The data and results of other algorithms come from [46]. Authors of the referred paper adopted majority of the data sets from repository of University of California (http://www.ics.uci.edu/ mlearn/MLSummary.html). The paper provides clustering results for following methods:

RBF Radial Based Functions Network, classical neural networks method. [18]

AdaBoost Adaptive boosting. [46]. The mentioned paper describes several variants of AdaBoost. We are comparing only the best one.

SVM Support Vector Machine [47]

KFD Kernel Fisher Discriminant [48]

The tested data consist of several datasets. Each dataset has 100 realizations and each realization consist of training data, training labels, test data and test labels. Detailed information about each dataset is in Table 8.3. For each realization, the classifier is built using the training data and training labels. Then, the classifier assigns a label to each data record in test data. Percentual number of misclassified data records is then evaluated. Its mean value and standard deviation over the 100 realizations is taken as the result for each classification method.

Table 8.4 shows the results for all investigated data-sets. We can se, that mixture-based classifier is comparable with other methods. It confirms that PB estimation gives reasonable results.

dataset name	twonorm	flare-solar	heart	german	ringnorm
training data records	400	666	170	700	400
test date records	7000	400	100	300	7000
data dimension	20	9	13	20	20
dataset name	titanic	thyroid	diabetis	breast-cancer	
training data records	150	140	468	200	
test date records	2051	75	300	77	
data dimension	3	5	8	9	

Table 8.3: Characteristics of data sets

Conclusions

The presented results shows that the mixture-based clustering gives results comparable with other methods. It shows that the estimation algorithms works well.

twonorm		
method	mean	std
Mix QB	2.58	0.20
Mix PB	2.60	0.22
KFD	2.61	0.15
AdaBoost	2.70	0.24
RBF	2.85	0.28
SVM	2.96	0.23

flare-solar		
method	mean	std
SVM	32.43	1.82
KFD	33.16	1.72
AdaBoost	34.20	2.18
RBF	34.37	1.95
Mix PB	35.49	1.38
Mix QB	36.66	1.98

heart		
method	mean	std
SVM	15.95	3.26
KFD	16.14	3.39
AdaBoost	16.47	3.51
RBF	17.55	3.25
Mix PB	21.51	3.94
Mix QB	21.69	3.77

german		
method	mean	std
SVM	23.61	2.07
KFD	23.71	2.20
AdaBoost	24.34	2.08
RBF	24.71	2.38
Mix PB	25.95	2.86
Mix QB	26.49	3.27

ringnorm		
method	mean	std
KFD	1.49	0.12
AdaBoost	1.58	0.12
SVM	1.66	0.12
Mix QB	1.69	0.23
Mix PB	1.69	0.27
RBF	1.70	0.21

titanic		
method	mean	std
SVM	22.42	1.02
Mix PB	22.43	1.31
Mix QB	22.45	1.44
AdaBoost	22.64	1.20
KFD	23.25	2.05
RBF	23.26	1.34

thyroid		
method	mean	std
Mix PB	3.39	1.78
Mix QB	3.51	1.92
KFD	4.20	2.07
RBF	4.52	2.12
AdaBoost	4.55	2.19
SVM	4.80	2.19

diabetis		
method	mean	std
KFD	23.21	1.63
SVM	23.53	1.73
AdaBoost	23.79	1.80
RBF	24.29	1.88
Mix QB	26.58	2.17
Mix PB	26.66	2.72

breast-cancer		
method	mean	std
KFD	24.77	4.63
Mix PB	25.66	4.74
SVM	26.04	4.74
AdaBoost	26.51	4.47
Mix QB	27.17	4.86
RBF	27.64	4.71

Table 8.4: Results of cluster analysis examples
Although the mixture based clustering is not the best one in all cases, it can be seen that it gives reasonable results. The PB variant seems to behave a little better than the QB variant.

8.1.6 Conclusions

Behavior of the PB estimation was illustrated on simple examples. On more complex examples, the PB algorithm was compared with the current QB algorithm. It was shown that using the PB estimation instead of the QB estimation brings significant quality increase. Moreover, it was shown that probabilistic mixtures can be successfully used in cluster analysis. Consequently, the PB estimation was selected as the default estimation method in MATLAB toolbox Mixtools.

8.2 Gaussian Mixtures with Dynamic Weights

This section deals with normal factors (Section 6.1) and various types of component weighting functions. The aim of this section is to present behavior of the PB estimation of mixture with dynamic weights.

8.2.1 Switching Weights

Here, the cwfs of type "hard bounded" (Section 7.2.6) are considered. Because the data are scalar, we can omit the channel index 1 again.

Model

$$\begin{aligned}
\mathring{d} &= 1 && \text{(data are scalar valued)}\\
\mathring{c} &= 2 && \text{(2 components)}\\
\phi_{t-1} &\equiv (d_{t-1}, 1) && \text{(state of the model)}\\
\Omega &\equiv (scalar) && \text{(parameter of cwfs)}\\
\Theta &\equiv (\theta_1, \theta_2, r_1, r_2, \Omega) && \text{(mixture parameter)}
\end{aligned}$$

$$\begin{aligned}
\alpha_1(\phi_{t-1}|\Omega) &\equiv \alpha_1(d_{t-1}, 1|\Omega) = \begin{cases} 0 & \text{if } d_{t-1} > \Omega \\ 1 & \text{if } d_{t-1} \le \Omega \end{cases} && \text{(1st cwf)}\\
\alpha_2(\phi_{t-1}|\Omega) &\equiv \alpha_2(d_{t-1}, 1|\Omega) = \begin{cases} 1 & \phi_{t-1} > \Omega \\ 0 & \phi_{t-1} \le \Omega \end{cases} && \text{(2nd cwf)}\\
f_1(d_t|\phi_{t-1}, \Theta_1) &\equiv f_1(d_t|\phi_{t-1}, \Theta_1) = \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_1, r_1\right) && \text{(1st component)}\\
f_2(d_t|\phi_{t-1}, \Theta_2) &\equiv f_2(d_t|\phi_{t-1}, \Theta_2) = \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_2, r_2\right) && \text{(2nd component)}\\
f(d_t|\phi_{t-1}, \Theta) &\equiv \begin{cases} \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_2, r_2\right) & \text{if } d_{t-1} > \Omega \\ \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_1, r_1\right) & \text{if } d_{t-1} \le \Omega \end{cases} && \text{(Mixture)}
\end{aligned}$$

Form of Prior and Posterior Pdfs

$$\begin{aligned}
\rho(\Omega|\mathcal{H}_t) &\equiv \rho(\Omega|M_t, R_t) = \mathcal{N}_\Omega\left(M_t, R_t\right)\\
\pi_1(\Theta_1|\mathcal{S}_{1;t}) &\equiv \pi_1(\theta_1, r_1|V_{1;t}, \nu_{1;t}) = GiW_{\theta_1, r_1}(V_{1;t}, \nu_{1;t})\\
\pi_2(\Theta_2|\mathcal{S}_{2;t}) &\equiv \pi_2(\theta_2, r_2|V_{2;t}, \nu_{2;t}) = GiW_{\theta_2, r_2}(V_{2;t}, \nu_{2;t})\\
\mathcal{H}_t &\equiv (M_t, R_t),\ \mathcal{S}_{1;t} \equiv (V_{1;t}, \nu_{1;t}),\ \mathcal{S}_{2;t} \equiv (V_{2;t}, \nu_{2;t})\\
\mathcal{G}_t &\equiv (M_t, R_t, V_{1;t}, \nu_{1;t}, V_{2;t}, \nu_{2;t})\\
\pi(\Theta|\mathcal{G}_t) &\equiv \pi(\theta_1, \theta_2, r_1, r_2, \Omega|M_t, R_t, V_{1;t}, \nu_{1;t}, V_{2;t}, \nu_{2;t}) \equiv\\
&\equiv \mathcal{N}_\Omega\left(M_t, R_t\right) GiW_{\theta_1, r_1}(V_{1;t}, \nu_{1;t}) GiW_{\theta_2, r_2}(V_{2;t}, \nu_{2;t})
\end{aligned}$$

True Value of Parameter and the Initial Statistics

$$\begin{aligned}
\Theta_{true} &\equiv (\theta_1 \equiv [0.200, 0.300],\ \theta_2 \equiv [0.200, -0.300],\\
&\quad r_1 \equiv 0.200,\ r_2 \equiv 0.100,\ \Omega \equiv -0.108)\\
\mathcal{G}_0 &\equiv \big(M_0 \equiv -2.000,\ R_0 \equiv 40.000,\\
&\quad C_{1;0} \equiv diag([2.000, 2.000]),\ \hat{\theta}_{1;0} \equiv [1.000, 1.000],\\
&\quad \lfloor^d D_{1;0} \equiv 0.315,\ \nu_{1;0} = 4.100,\\
&\quad C_{2;0} \equiv diag([2.000, 2.000]),\ \hat{\theta}_{2;0} \equiv [1.000, -1.000],\\
&\quad \lfloor^d D_{2;0} \equiv 0.315,\ \nu_{2;0} = 4.100\big)
\end{aligned}$$

We simulated 500 data records. The simulated data and diagram of the correspondence of each data record to the component it was generated from are depicted on Figure 8.12. Figure 8.13 shows evolution

of statistics M_t and R_t during the estimation. Because M_t is in fact a point estimate of the unknown cwf parameter Ω, we can simply see that the point estimate approaches the true value.

Figure 8.14 shows the quality measure Q, discussed in Section 8.1.1. For comparison, Q is displayed even for the case of treating this model as a mixture with constant weights. It just illustrates the obvious fact, that mixtures with dynamic weights can not be simply approximated by static-weights mixtures of the same complexity.

For estimation of this model, analytical expressions derived in Section 8.1.1 were used. For debugging purposes, we also tried to estimate the same model using the general Monte-Carlo approximation from Section 7.2.5. For $N \equiv 10000$ MC samples per approximation, we obtained exactly the same result as the presented one.

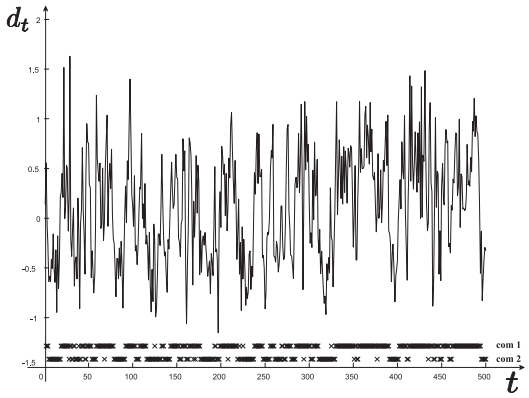


Figure 8.12: Data generated and active component
The figure shows the data generated from the mixture with true parameters. The small crosses underneath the figure denotes which component was active in each particular time.

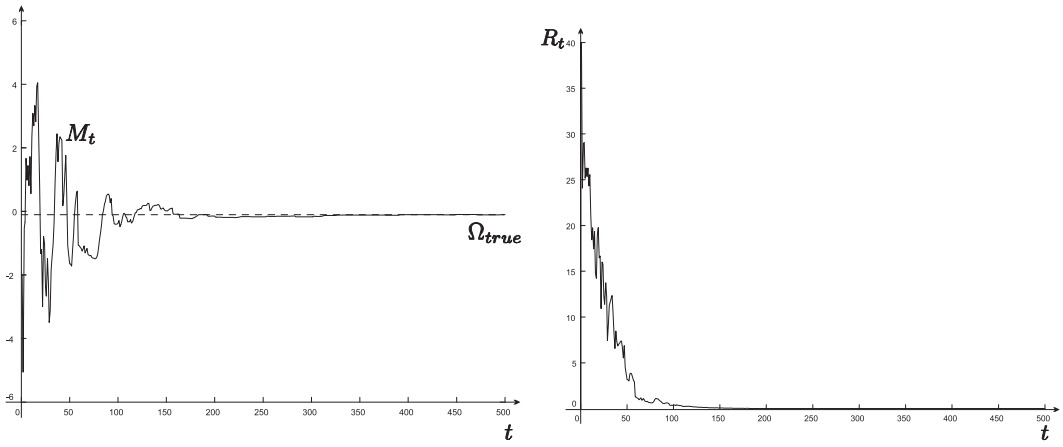


Figure 8.13: Evolution of statistics M_t and R_t
The left hand part of this figure shows how the point estimate of cwf parameter M_t approaches the true value Ω_{true}. The right hand part of this figure shows evolution of statistics R_t. Because the statistic R_t is in fact variance of point estimate M_t, the decreasing trend of R_t indicates increasing quality of the point estimate.

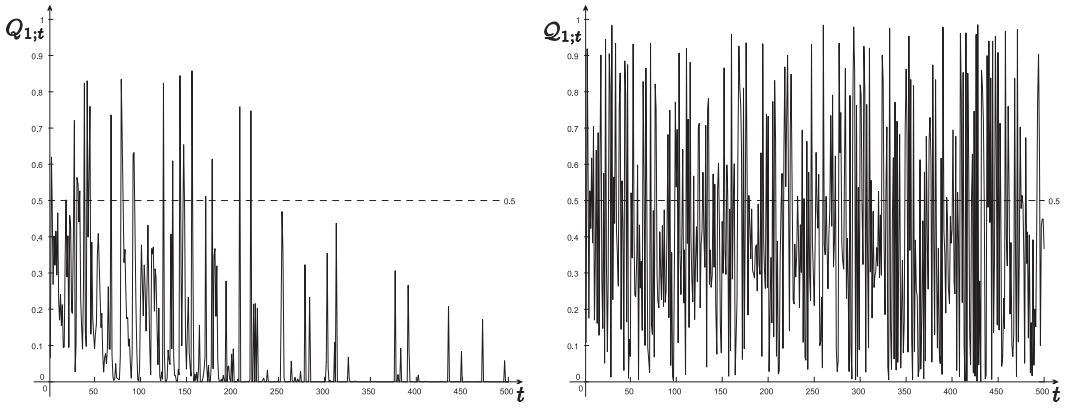


Figure 8.14: Quality of estimation

The left hand part of this figure shows evolution of the quality indicator $Q_{1;t}$ determining the quality of the PB estimation. It can be seen that the quality is increasing during time. After some time the algorithm performs almost exactly as the Bayesian estimation. The right hand part of this figure shows evolution of quality indicator $\mathcal{Q}_{1;t}$ determining the quality of the PB estimation with the static-weights model. It just illustrates the obvious fact, that mixtures with dynamic weights can not be simply approximated by static-weights mixtures of the same complexity.

8.2.2 Gaussian Ratio Dynamic Weights

Here, the cwfs of type Gaussian ratio (Section 7.2.6) are considered. Because the data are scalars, we can omit the channel index 1 again.

Model

$$
\begin{aligned}
\mathring{d} &= 1 && \text{(data are scalars)}\\
\mathring{c} &= 2 && \text{(2 components)}\\
\phi_{t-1} &\equiv (d_{t-1}, 1) && \text{(state of the model)}\\
\Omega &\equiv ({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_1,\ {}^{\lfloor\alpha}r_2) && \text{(parameter of cwfs)}\\
\Theta &\equiv (\theta_1, \theta_2, r_1, r_2, \Omega) && \text{(mixture parameter)}
\end{aligned}
$$

$$
\begin{aligned}
\alpha_1(\phi_{t-1}|\Omega) &\equiv \alpha_1(d_{t-1}|\,{}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_1,\ {}^{\lfloor\alpha}r_2) =\\
&= \frac{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)}{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)+\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)} && \text{(1st cwf)}
\end{aligned}
$$

$$
\begin{aligned}
\alpha_2(\phi_{t-1}|\Omega) &\equiv \alpha_2(d_{t-1}|\,{}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_1,\ {}^{\lfloor\alpha}r_2) =\\
&= \frac{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)}{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)+\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)} && \text{(2nd cwf)}\\
f_1(d_t|\phi_{t-1},\Theta_1) &\equiv f_1(d_t|\phi_{t-1},\Theta_1) = \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_1, r_1\right) && \text{(1st component)}\\
f_2(d_t|\phi_{t-1},\Theta_2) &\equiv f_2(d_t|\phi_{t-1},\Theta_2) = \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_2, r_2\right) && \text{(2nd component)}\\
f(d_t|\phi_{t-1},\Theta) &\equiv \frac{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)}{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)+\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)}\mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_1, r_1\right) +\\
&\quad + \frac{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)}{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)+\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)}\mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_2, r_2\right) && \text{(Mixture)}
\end{aligned}
$$

Form of Prior and Posterior Pdfs

$$
\begin{aligned}
\rho(\Omega|\mathcal{H}_t) &\equiv \rho({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_1,\ {}^{\lfloor\alpha}r_2|\,{}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}\nu_{1;t},\ {}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{2;t}) \equiv\\
&\equiv GiW_{{}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1}({}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}\nu_{1;t}) GiW_{{}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_2}({}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{2;t})\\
\pi_1(\Theta_1|\mathcal{S}_{1;t}) &\equiv \pi_1(\theta_1, r_1|V_{1;t}, \nu_{1;t}) = GiW_{\theta_1,r_1}(V_{1;t}, \nu_{1;t})\\
\pi_2(\Theta_2|\mathcal{S}_{2;t}) &\equiv \pi_2(\theta_2, r_2|V_{2;t}, \nu_{2;t}) = GiW_{\theta_2,r_2}(V_{2;t}, \nu_{2;t})\\
\mathcal{H}_t &\equiv ({}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{1;t},\ {}^{\lfloor\alpha}\nu_{2;t}),\ \mathcal{S}_{1;t} \equiv (V_{1;t}, \nu_{1;t}),\ \mathcal{S}_{2;t} \equiv (V_{2;t}, \nu_{2;t})\\
\mathcal{G}_t &\equiv ({}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{1;t},\ {}^{\lfloor\alpha}\nu_{2;t}, V_{1;t}, \nu_{1;t}, V_{2;t}, \nu_{2;t})\\
\pi(\Theta|\mathcal{G}_t) &\equiv \pi(\theta_1, \theta_2, r_1, r_2,\ {}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_1,\ {}^{\lfloor\alpha}r_2|\,{}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{1;t},\ {}^{\lfloor\alpha}\nu_{2;t}, V_{1;t}, \nu_{1;t}, V_{2;t}, \nu_{2;t}) \equiv\\
&\equiv GiW_{\theta_1,r_1}(V_{1;t}, \nu_{1;t}) GiW_{\theta_2,r_2}(V_{2;t}, \nu_{2;t}) \times\\
&\quad \times GiW_{{}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1}({}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}\nu_{1;t}) GiW_{{}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_2}({}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{2;t})
\end{aligned}
$$

True Value and Initial Statistics

$$
\begin{aligned}
\Theta_{true} &\equiv (\theta_1 \equiv [0.200, 0.300],\ \theta_2 \equiv [0.200, -0.300],\ r_1 = 0.200,\ r_2 = 0.100,\\
&\quad {}^{\lfloor\alpha}\theta_1 \equiv 1.000,\ {}^{\lfloor\alpha}\theta_2 \equiv -1.000,\ {}^{\lfloor\alpha}r_1 = 1.500,\ {}^{\lfloor\alpha}r_2 = 2.000)\\
\mathcal{G}_0 &\equiv (\\
&\quad C_{1;0} = diag([2.000, 2.000]),\ \hat{\theta}_{1;0} \equiv [1.000, 1.000],\\
&\quad {}^{\lfloor d}D_{1;0} = 0.315,\ \nu_{1;0} = 4.100,\\
&\quad C_{2;0} = diag([2.000, 2.000]),\ \hat{\theta}_{2;0} \equiv [1.000, -1.000],
\end{aligned}
$$

$$\begin{aligned}
&{}^{\lfloor d}D_{2;0} = 0.315,\ \nu_{2;0} = 4.100\\
&{}^{\lfloor\alpha}C_{1;0} \equiv 40.000,\ {}^{\lfloor\alpha}\hat{\theta}_{1;0} \equiv 0.000,\ {}^{\lfloor\alpha\,\lfloor d}D_{1;0} \equiv 6.600,\ {}^{\lfloor\alpha}\nu_{1;0} \equiv 4.200\\
&{}^{\lfloor\alpha}C_{2;0} \equiv 40.000,\ {}^{\lfloor\alpha}\hat{\theta}_{2;0} \equiv 0.000,\ {}^{\lfloor\alpha\,\lfloor d}D_{2;0} \equiv 6.600,\ {}^{\lfloor\alpha}\nu_{2;0} \equiv 4.200\big)
\end{aligned}$$

Figure 8.15 shows the data generated and active components in each time. Right hand part of this figure shows the true cwfs $\alpha(d_{t-1}|\Omega = \Omega_{true})$. Evolution of statistics ${}^{\lfloor\alpha}\hat{\theta}_{1;t}$, ${}^{\lfloor\alpha}\hat{\theta}_{2;t}$ and ${}^{\lfloor\alpha}C_{1;t}$, ${}^{\lfloor\alpha}C_{2;t}$ is depicted on Figure 8.16. Because these statistics ${}^{\lfloor\alpha}\hat{\theta}_{1;t}$, ${}^{\lfloor\alpha}\hat{\theta}_{2;t}$ are also point estimates of cwf parameters ${}^{\lfloor\alpha}\theta_1$, ${}^{\lfloor\alpha}\theta_2$, we can confront them with the true values ${}^{\lfloor\alpha}\theta_{1true}$, ${}^{\lfloor\alpha}\theta_{2true}$. We can see that the estimates are close to the true value, but they are not approaching it. This is still reasonable behavior, because for this type of cwfs, different values of parameters can give very similar forms of cwfs. Hence we should look on another quality indicators.

Figure 8.17 displays the indicator of estimation quality $\mathcal{Q}$ (see Section 8.1.1) and point estimate of cwfs $\alpha(d_{t-1}|\Omega = \hat{\Omega}_{500})$. Also the difference from the correct cwfs $E(d_{t-1}) = \text{abs}\left(\alpha_1(d_{t-1}|\Omega = \hat{\Omega}_{500}) - \alpha_1(d_{t-1}|\Omega = \Omega_{true})\right)$ is displayed there.

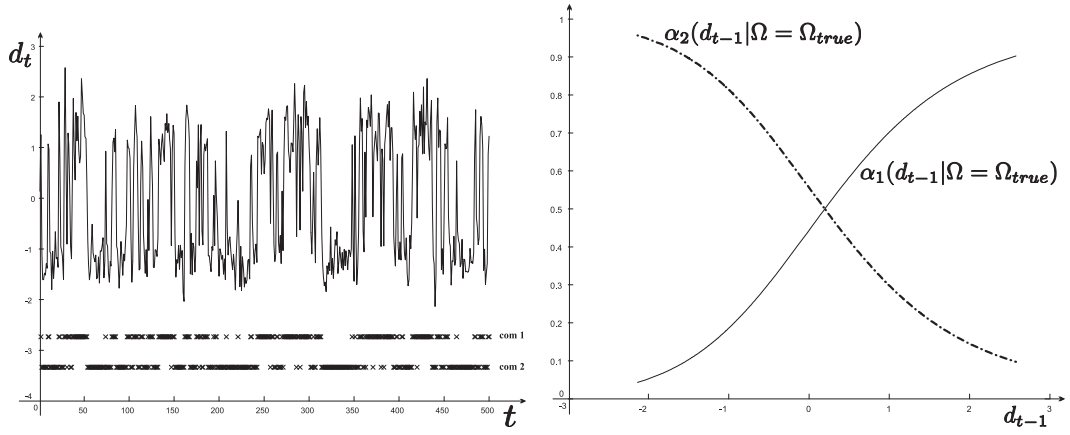


Figure 8.15: Data generated and true cwf

Left hand part of this figure shows the data generated and active components in each time. Right hand part of this figure shows the true cwfs. It can be seen how the last data record d_{t-1} influences the active component in the next step. If d_{t-1} is near to zero, both components have approximately the same chance to become active. With d_{t-1} receding from zero, chances of one of the components to be active are increasing.

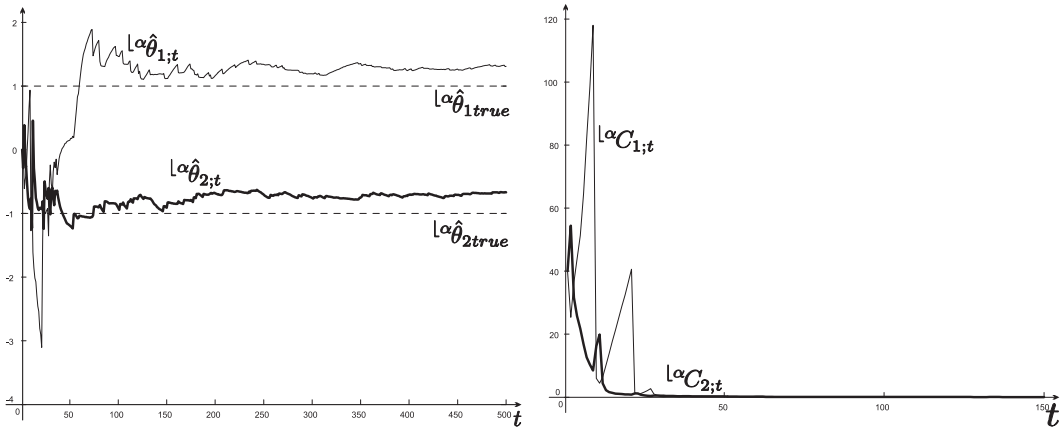


Figure 8.16: Evolution of statistics during estimation

Left hand part of this figure shows evolution of the statistics $\lfloor^{\alpha}\hat{\theta}_{1;t}$, $\lfloor^{\alpha}\hat{\theta}_{2;t}$. *Because these statistics are also point estimates of cwf parameters* $\lfloor^{\alpha}\theta_1$, $\lfloor^{\alpha}\theta_2$, *we can confront them with the true values* $\lfloor^{\alpha}\theta_{1true}$, $\lfloor^{\alpha}\theta_{2true}$. *Right hand part of this figure shows evolution of statistics* $\lfloor^{\alpha}C_{1;t}$, $\lfloor^{\alpha}C_{2;t}$. *Because the covariance of point estimates* $\lfloor^{\alpha}\hat{\theta}_{c;t}$ *is proportional to* $\lfloor^{\alpha}C_{c;t}$, *the decreasing trends of* $\lfloor^{\alpha}C_{1;t}$, $\lfloor^{\alpha}C_{2;t}$ *indicates increasing quality of the point estimates.*

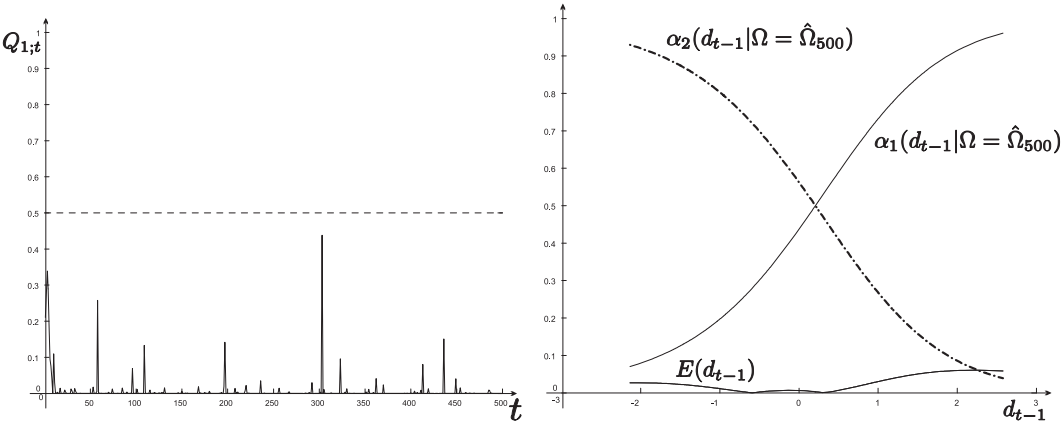


Figure 8.17: Estimation quality and point estimate of cwf

Left hand part of this figure displays the estimation quality Q_t *(see section 8.1.1). It can be seen, that the value of* Q_t *is very low, which indicates that almost correct Bayesian estimation was performed. Right hand part of this figure shows the point estimate of cwfs* $\alpha(d_{t-1}|\Omega = \hat{\Omega}_{500})$. *Also the difference from the correct cwfs* $E(d_{t-1}) = abs\Big(\alpha_1(d_{t-1}|\Omega = \hat{\Omega}_{500}) - \alpha_1(d_{t-1}|\Omega = \Omega_{true})\Big)$ *is displayed here. It can be seen, that the estimated cwf is very close to the true one.*

8.2.3 Gaussian Ratio Weights II

Because the data are scalars, we can omit the channel index 1 again.

Model

$$
\begin{array}{rcll}
\mathring{d} & = & 1 & \text{(data are scalar)} \\
\mathring{c} & = & 3 & \text{(3 components)} \\
\phi_{t-1} & \equiv & (d_{t-1}, 1) & \text{(state of the model)} \\
\Omega & \equiv & ({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}\theta_3,\ {}^{\lfloor\alpha}r_1,\ {}^{\lfloor\alpha}r_2,\ {}^{\lfloor\alpha}r_3) & \text{(parametr of cwfs)} \\
\Theta & \equiv & (\theta_1, \theta_2, \theta_3, r_1, r_2, r_3, \Omega) & \text{(mixture parameter)}
\end{array}
$$

$$
\begin{array}{rcll}
\alpha_c(\phi_{t-1}|\Omega) & \equiv & \frac{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_c,\ {}^{\lfloor\alpha}r_c)}{\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_1,\ {}^{\lfloor\alpha}r_1)+\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_2,\ {}^{\lfloor\alpha}r_2)+\mathcal{N}_{d_{t-1}}({}^{\lfloor\alpha}\theta_3,\ {}^{\lfloor\alpha}r_3)} & \text{(c-th cwf)} \\
f_c(d_t|\phi_{t-1}, \Theta_i) & \equiv & \mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_c, r_c\right) & \text{(c-th component)} \\
f(d_t|\phi_{t-1}, \Theta) & \equiv & \sum_{c=1}^{3} \alpha_c(\phi_{t-1}|\Omega)\mathcal{N}_{d_t}\left(\phi'_{t-1}\theta_c, r_c\right) & \text{(Mixture)}
\end{array}
$$

Form of Prior and Posterior Pdfs

$$
\begin{array}{rcl}
\rho(\Omega|\mathcal{H}_t) & \equiv & \rho(\Omega|\,{}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}\nu_{1;t},\ {}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}\nu_{2;t},\ \ {}^{\lfloor\alpha}V_{3;t},\ {}^{\lfloor\alpha}\nu_{3;t}) \equiv \\
& \equiv & \prod_{c=1}^{3} GiW_{{}^{\lfloor\alpha}\theta_c,\ {}^{\lfloor\alpha}r_c}({}^{\lfloor\alpha}V_{c;t},\ {}^{\lfloor\alpha}\nu_{c;t}) \\
\pi_c(\Theta_c|\mathcal{S}_{c;t}) & \equiv & \pi_c(\theta_c, r_c|V_c, \nu_c) = GiW_{\theta_c, r_c}(V_{c;t}, \nu_{c;t}) \\
\mathcal{H}_t & \equiv & ({}^{\lfloor\alpha}V_{1;t},\ {}^{\lfloor\alpha}V_{2;t},\ {}^{\lfloor\alpha}V_{3;t},\ {}^{\lfloor\alpha}\nu_{1;t},\ {}^{\lfloor\alpha}\nu_{2;t},\ {}^{\lfloor\alpha}\nu_{3;t}) \\
\mathcal{S}_{c;t} & \equiv & (V_{c;t}, \nu_{c;t}) \\
\mathcal{G}_t & \equiv & ({}^{\lfloor\alpha}V_{c;t},\ {}^{\lfloor\alpha}\nu_{c;t}, V_{c;t}, \nu_{c;t},\ c \in (1,2,3)) \\
\pi(\Theta|\mathcal{G}_t) & \equiv & \prod_{c=1}^{3} GiW_{{}^{\lfloor\alpha}\theta_c,\ {}^{\lfloor\alpha}r_c}({}^{\lfloor\alpha}V_{c;t},\ {}^{\lfloor\alpha}\nu_{c;t}) \prod_{c=1}^{3} GiW_{\theta_c, r_c}(V_{c;t}, \nu_{c;t})
\end{array}
$$

True Value and Initial Statistics

$$
\begin{array}{rcl}
\Theta_{true} & \equiv & (\theta_1 \equiv -0.300,\ \theta_2 \equiv -1.300,\ \theta_3 \equiv 1.000, \\
& & r_1 \equiv 0.100,\ r_2 \equiv 0.050,\ r_3 \equiv 0.040, \\
& & {}^{\lfloor\alpha}\theta_1 \equiv 0.000,\ {}^{\lfloor\alpha}\theta_2 \equiv -3.000,\ {}^{\lfloor\alpha}\theta_3 \equiv 2.000, \\
& & {}^{\lfloor\alpha}r_1 \equiv 0.100,\ {}^{\lfloor\alpha}r_2 \equiv 0.500,\ {}^{\lfloor\alpha}r_3 \equiv 0.500) \\
\mathcal{G}_0 & \equiv & \big(\\
& & C_{1;0} \equiv 20.000,\ \hat{\theta}_{1;0} \equiv 0.000,\ {}^{\lfloor d}D_{1;0} \equiv 0.1050,\ \nu_{1;0} \equiv\ 4.100, \\
& & C_{2;0} \equiv 20.000,\ \hat{\theta}_{2;0} \equiv -4.000,\ {}^{\lfloor d}D_{2;0} \equiv 2.1,\ \nu_{2;0} \equiv 4.100 \\
& & C_{3;0} \equiv 20.000,\ \hat{\theta}_{3;0} \equiv -2.000,\ {}^{\lfloor d}D_{3;0} \equiv 0.1050,\ \nu_{3;0} \equiv 4.100 \\
& & {}^{\lfloor\alpha}C_{1;0} \equiv 40.000,\ {}^{\lfloor\alpha}\hat{\theta}_{1;0} \equiv 0.000,\ {}^{\lfloor\alpha\,\lfloor d}D_{1;0} \equiv 0.220,\ {}^{\lfloor\alpha}\nu_{1;0} \equiv 4.200 \\
& & {}^{\lfloor\alpha}C_{2;0} \equiv 40.000,\ {}^{\lfloor\alpha}\hat{\theta}_{2;0} \equiv 0.000,\ {}^{\lfloor\alpha\,\lfloor d}D_{2;0} \equiv 0.220,\ {}^{\lfloor\alpha}\nu_{2;0} \equiv 4.200 \\
& & {}^{\lfloor\alpha}C_{3;0} \equiv 40.000,\ {}^{\lfloor\alpha}\hat{\theta}_{3;0} \equiv 0.000,\ {}^{\lfloor\alpha\,\lfloor d}D_{3;0} \equiv 0.220,\ {}^{\lfloor\alpha}\nu_{3;0} \equiv 4.200\big)
\end{array}
$$

Figure 8.18 shows the data generated and active components in each time. Right hand part of this figure shows the true cwfs.

Evolution of statistics ${}^{\lfloor\alpha}\hat{\theta}_{1;t},\ {}^{\lfloor\alpha}\hat{\theta}_{2;t},\ {}^{\lfloor\alpha}\hat{\theta}_{3;t}$ and ${}^{\lfloor\alpha}C_{1;t},\ {}^{\lfloor\alpha}C_{2;t},\ {}^{\lfloor\alpha}C_{3;t}$ is depicted on Figure 8.19. Figure 8.20 displays the estimation quality Q (see Section 8.1.1) and the point estimate of cwfs. Evolution of statistics $\hat{\theta}_{1;t},\ \hat{\theta}_{2;t},\ \hat{\theta}_{3;t}$ is displayed on Figure 8.21.

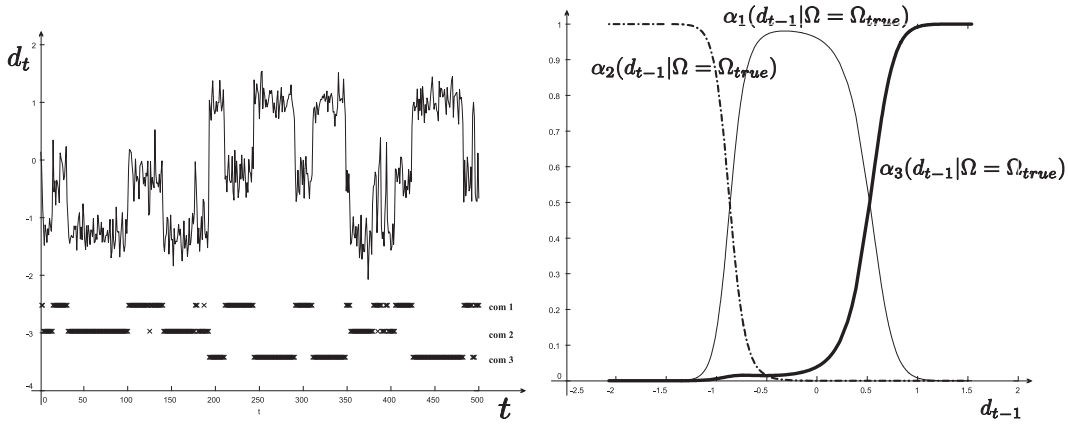


Figure 8.18: Data generated and original cwfs

Left hand part of this figure shows the data generated and active components in each time. Right hand part of this figure shows the true cwfs. Note that the third component was not active roughly in initial 200 time moments.

8.2.4 Conclusions

On three examples, we showed that the estimation of mixtures with dynamic weights using the presented algorithm gives reasonable results. Of course, use of Monte-Carlo integral approximation is limited to low-dimensional cases only. Alternative approximations are needed for high dimensional cases.

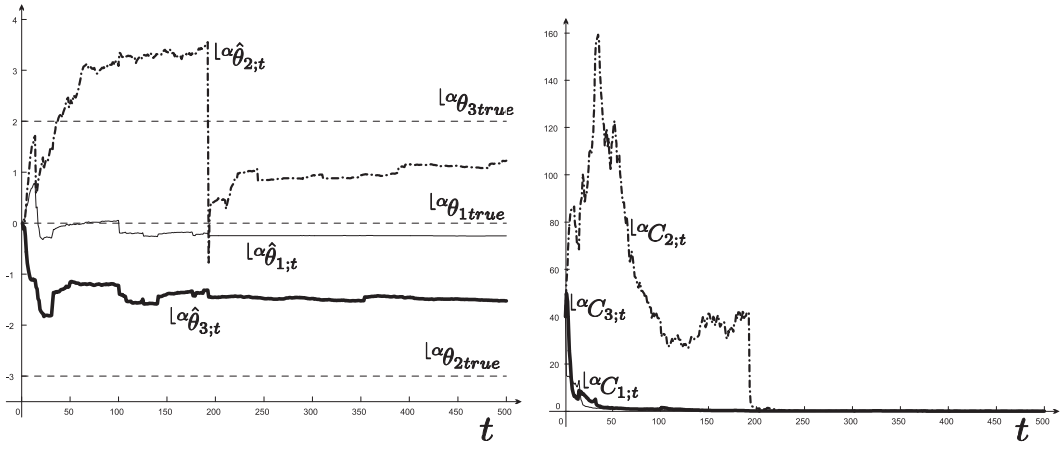


Figure 8.19: Evolution of statistics during estimation

Left hand part of this figure shows evolution of statistics $\lfloor^{\alpha}\hat{\theta}_{1;t}$, $\lfloor^{\alpha}\hat{\theta}_{2;t}$, $\lfloor^{\alpha}\hat{\theta}_{3;t}$. *Right hand part of this figure shows evolution of statistics* $\lfloor^{\alpha}C_{1;t}$, $\lfloor^{\alpha}C_{2;t}$, $\lfloor^{\alpha}C_{3;t}$. *It should be also mentioned that components 2 and 3 are permuted in the estimated mixture. It can be seen that* $\lfloor^{\alpha}C_{2;t}$ *is relatively high and* $\lfloor^{\alpha}\hat{\theta}_{2;t}$ *completely bad for time moments lower than 200. After that time moment, the third component started to be active for the first time and both* $\lfloor^{\alpha}\hat{\theta}_{2;t}$ *and* $\lfloor^{\alpha}C_{2;t}$ *have reasonable values almost immediately.*

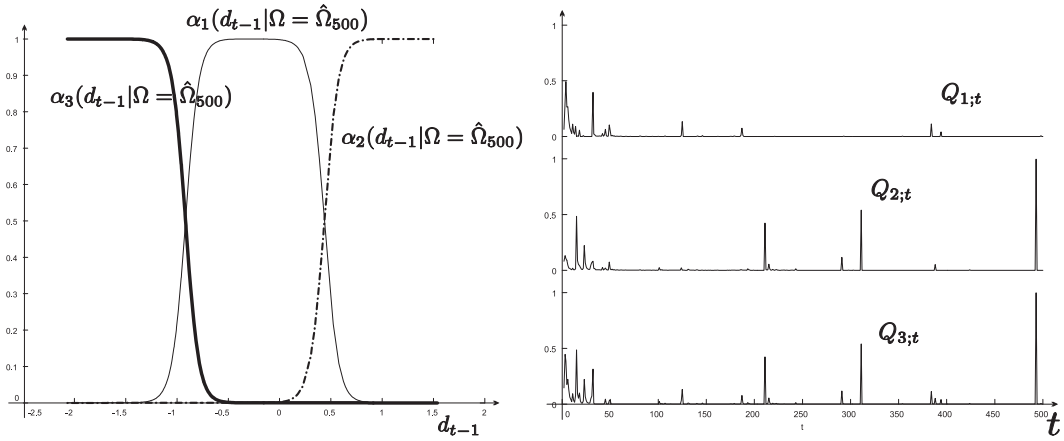


Figure 8.20: Estimation quality and point estimate of cwf

Left hand part of this figure shows point estimate of cwfs. It can be seen that this estimate is similar tu the true cwfs up to the fact that cwf 3 and 2 are permuted. Right hand part of this figure shows the quality indicators $Q_{1;t}, Q_{2;t}, Q_{3;t}$. *It can be seen that the estimation was very good during almost all the time. The several time moments with big values of* Q_t *can not influence the overall result.*

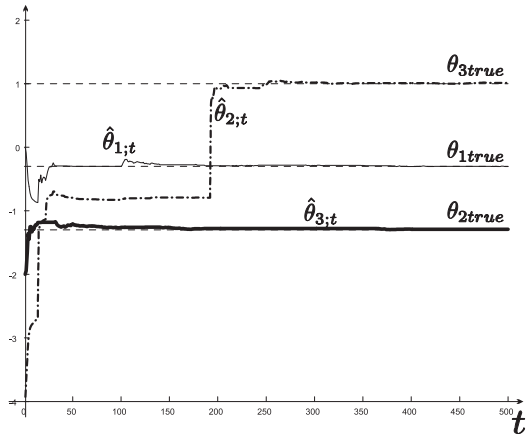


Figure 8.21: Evolution of statistics $\hat{\theta}_{1;t},\ \hat{\theta}_{2;t},\ \hat{\theta}_{3;t}$

This figure shows evolution of statistics $\hat{\theta}_{1;t}, \hat{\theta}_{2;t}, \hat{\theta}_{3;t}$, which represent point estimates of the component parameters. It can be seen how the estimates approach the true values. It can also be seen that components 3 and 2 are switched.

Chapter 9

Conclusions

Within this work, estimation of dynamic probabilistic mixtures was improved by designing new projection based (PB) algorithm. Moreover, the dynamic probabilistic mixtures were generalized to work with data-dependent component weights. Here, the main outcomes of the work are summarized:

- Dynamic probabilistic mixture model with dynamic weights was defined as a generalization of the current dynamic mixture with static weights. (Chapter 4)
- General algorithm for recursive estimation of the generalized model was elaborated. Problem of minimization of KL divergence was converted into a simpler task of evaluation of moments of involved pdfs. Monte-Carlo integration was successfully used for evaluating these moments in low-dimensional cases. (Chapters 5,3,7)
- The algorithm was applied to components composed of normal factors with known or unknown variance. Two types of component weighting functions were defined, one of them is very general. (Chapters 6,7)
- The algorithm was specialized for mixtures with static weights. (Chapter 7)
- All algorithms were implemented in MATLAB.
- Algorithms for static-weights mixtures were implemented in C and integrated into MATLAB toolbox Mixtools.
- Quality of the new algorithm was compared with the current quasi-Bayes algorithm on a large set of examples of estimation of a static-weights mixtures. Results of the comparison show that PB algorithm is better. Consecutively, PB estimation was selected as a default estimation method in the Mixtools toolbox. (Chapter 8)
- Static probabilistic mixtures was successfully used on the field of cluster analysis.(Chapter 8)
- Reliability of the estimation of mixture with dynamic weights was demonstrated on several simple examples.(Chapter 8)
- Results of the work were continuously published.([49, 50, 51, 52, 53, 54, 45, 55])

Significance for Science

Possibility of using approximations based on correct argument order of Kullback-Leibler divergence was shown on important class of models.

The work opened a new way of working with data dependent weights as it converted the problem of approximation of Bayesian estimation to approximation of moments of complex probability density functions.

The work contributed to improvement of Bayesian decision-making with probabilistic mixture models.

Significance for Applications

There exist many applications based on Bayesian decision making with probabilistic mixtures [56]. As the estimation forms one of the keystones of all such applications, its improvement has to have positive impact on them. Preliminary experiments confirms the overall improvement.

In the cases, where the mixtures with static weights was unsuccessfully applied, there is a chance that mixtures with dynamic weights can be successful.

Open Problems

The Monte-Carlo evaluation of pdf moments needed in the general version of PB estimation is applicable only to low-dimensional cases. The task of future research is to approximate the moments with another method, so that mixtures with dynamic weights can be estimated for high-dimensional component weighting functions.

The correct posterior pdf connected with the mixture model is a mixture with number of components growing up exponentially with number of data samples. Within this work, we approximate this mixture by one component only. In future, we should try to approximate this mixture by mixture with predefined fixed number of components. This will open new problems, because KL divergence of two mixtures cannot be simply evaluated.

Appendix A

The Quasi-Bayes Algorithm and Mixinit

Here, we will briefly describe the quasi-Bayes (QB) estimation algorithm and algorithm `mixinit` for initialization of mixture estimation. The QB algorithm has been used extensively in real-life applications [10], and it is proven to be reasonably reliable and computationally efficient. This text refers to it as a standard, which is to be improved. It was designed for mixtures with constant weights.

A.1 The Quasi-Bayes Algorithm

The general QB algorithm uses the following rule, see [21]:

$$\begin{aligned} \kappa_t &= \kappa_{t-1} + w_t \\ \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t}) &\propto [f_{ic}(d_{ic;t}|\psi_{ic;t}, \Theta_{ic})]^{w_{c;t}}\, \pi_{ic}(\Theta_{ic}|\mathcal{S}_{ic;t-1}), \end{aligned}$$

where w_t is defined in (5.2). Application of this general algorithm to normal factors yields:

$$V_{ic;t} = V_{ic;t-1} + w_c \Psi_{ic;t}\Psi'_{ic;t},\ \nu_{ic;t} = \nu_{ic;t-1} + w_c,\ \kappa_{\bullet;t} = \kappa_{\bullet;t-1} + w_{\bullet;t}, \tag{A.1}$$

where $V_{ic;t}, \nu_{ic;t}$ are defined in Section 6.1.1. We would receive exactly this result, if we used the PB algorithm with approximations from Sections 6.1.5 and 7.1.5.

A.2 Mixinit

In AS department ÚTIA, algorithm `mixinit` for initialization of mixture estimation was developed. As the input, it takes set of data records, maximum order of the system and prior information. Result of this algorithm is estimated structure of the system in form of dynamic probabilistic mixture with static weights and prior pdf, which can be used for consequent mixture estimation.

Mixinit consists of repetitional calls of mixture estimation algorithm. Roughly speaking, it selects system structure and prior pdf in a sophisticated way and then performs mixture estimation. This is repeated many times until the best v-likelihood [10] is achieved. As the `mixinit` algorithm consists of many mixture estimation steps, increase of quality of mixture estimation will also induce increase of quality of initialization.

Appendix B

Exploited Calculus and Linear Algebra

This chapter collects the most important used results from matrix calculus and algebra. The missing proofs can be found e.g. in [57, 41].

B.1 Matrix Calculus

Proposition 7 (Integral formulas with trace)

$$\begin{aligned} x'Ax &= \mathbf{tr}\left(x'Ax\right) = \mathbf{tr}\left(Axx'\right) \\ \int AXdX &= A\int XdX \\ \int \mathbf{tr}\left(AX\right)dX &= \mathbf{tr}\left(A\int XdX\right) \\ \int x'Axdx &= \mathbf{tr}\left(A\int xx'dx\right) \end{aligned}$$

Proposition 8 (Differential formulas for scalar functions of matrices) *Derivatives of scalar function of matrix arguments are defined element-wise, i.e.* $\left\{\frac{\partial f}{\partial x}\right\}_{ij} = \frac{\partial f}{\partial x_{ij}}$

$$\begin{aligned} \frac{\partial x'b}{\partial x} &= b \\ \frac{\partial x'Cx}{\partial x} &= 2Cx, \text{ for symmetric } C \\ \frac{\partial \mathbf{tr}\left(XA\right)}{\partial X} &= A' \\ \frac{\partial \ln\left(|X|\right)}{\partial X} &= X^{-1} \\ \frac{\partial a'Xb}{\partial X} &= ab' \end{aligned}$$

Proposition 9 (Differential formulas for matrix functions of matrices) *Derivatives of vector function of vector arguments is defined as follows.* $\left\{\frac{\partial f}{\partial x}\right\}_{ij} = \frac{\partial f_i}{\partial x_j}$. *Derivatives of matrix functions of matrix*

arguments are defined on vectors constructed from columns of the matrices.

$$\begin{aligned}\frac{\partial Cx}{\partial x} &= C\\ \frac{\partial C^{-1}}{\partial C} &= -C^{-1}\otimes C^{-1}\\ \frac{\partial Cx}{\partial C} &= I\otimes x,\ \textit{where } I \textit{ denotes identity matrix and } \otimes \textit{ denotes Kronecker product.}\end{aligned}$$

Proposition 10 (Minimization) *Let $f(x)$ be 2-times continuously differentiable multivariate function. Then f has local minimum (maximum) in point x_0 iff*

$$\begin{aligned}\frac{\partial f}{\partial x}(x_0) &= 0\\ \frac{\partial^2 f}{\partial x \partial x}(x_0) &\quad \textit{is positive (negative) definite}\end{aligned}$$

B.2 Matrix Algebra

Proposition 11 (Kronecker product) *Let C be positive definite matrix. Then, the Kronecker product $C\otimes C$ is positive definite.*

Proposition 12 (Silvester's criterion) *The matrix C is positive definite iff all main minors of its determinant are positive.*

Proposition 13 (Positive definiteness) *Let matrix C be regular and matrix A be symmetric and positive definite. Then the matrix $C'AC$ is symmetric positive definite.*

Proof:
The matrix A is positive definite, i.e for each $y \neq 0$ it holds: $y'Ay > 0$. We want to show that for each $x \neq 0,\ x'C'ACx > 0$.
C is regular, hence $Cx \neq 0$ for $x \neq 0$, hence $x'C'ACx = \underbrace{(Cx)'}_{z'} A \underbrace{(Cx)}_{z} = z'Az > 0$ □

Proposition 14 (Determinant of the matrix I+xx') *Let x be a column vector of the length n. Then*

$$|I + xx'| = 1 + x'x$$

Proof: First, we will prove that x is eigenvector of the matrix $(I + xx')$ with eigenvalue $1 + x'x$.

$$(I + xx')x = x + xx'x = x(1 + x'x) = (1 + x'x)x$$

Let's now take such linear independent vectors $y_1, \cdots, y_{\mathring{x}-1}$, so that $x'y_i = 0,\ \forall i$. We will prove, that such vectors are eigenvectors of the matrix $(I + xx')$ with eigenvalues 1.

$$(I + xx')y_i = y_i + xx'y_i = y_i + x(x'y_i) = y_i$$

□

B.3 Other Relations

Proposition 15 (Simple algebraic manipulation) *Let $\sum_{c=1}^{\mathring{c}} w_{c;t} = 1$. It holds:*

$$\sum_{j,c=1}^{\mathring{d},\mathring{c}} w_{c;t}\mathcal{K}^U_{jc} + \sum_{c=1}^{\mathring{c}} w_{c;t} \sum_{\substack{j,r=1\\ r\neq c}}^{\mathring{d},\mathring{c}} \mathcal{K}_{jr} = \sum_{j,c=1}^{\mathring{d},\mathring{c}} \left[w_{c;t}\mathcal{K}^U_{jc} + (1 - w_{c;t})\mathcal{K}_{jc}\right] \tag{B.1}$$

Proof:

$$\sum_{j,c=1}^{\mathring{d},\mathring{c}} w_{c;t}\mathcal{K}_{jc}^{U} + \sum_{c=1}^{\mathring{c}} w_{c;t} \sum_{\substack{j,r=1 \\ r\neq c}}^{\mathring{d},\mathring{c}} \mathcal{K}_{jr} = \sum_{j,c=1}^{\mathring{d},\mathring{c}} w_{c;t}\mathcal{K}_{jc}^{U} + \sum_{c=1}^{\mathring{c}} w_{c;t} \left[\sum_{j,r=1}^{\mathring{d},\mathring{c}} \mathcal{K}_{jr} - \sum_{j=1}^{\mathring{d}} \mathcal{K}_{jc} \right] =$$

$$= \sum_{j,c=1}^{\mathring{d},\mathring{c}} w_{c;t}\mathcal{K}_{jc}^{U} + \sum_{j,r=1}^{\mathring{d},\mathring{c}} \mathcal{K}_{jr} - \sum_{j,c=1}^{\mathring{d},\mathring{c}} w_{c;t}\mathcal{K}_{jc} = \sum_{j,c=1}^{\mathring{d},\mathring{c}} \left[w_{c;t}\mathcal{K}_{jc}^{U} + (1 - w_{c;t})\mathcal{K}_{jc} \right]$$

□

B.4 Properties of the Digamma and Trigamma Functions

This part summarizes some special properties of digamma and trigamma functions. Although these functions can be defined both for positive and negative values, we deal only with the part defined on $(0, +\infty)$. For a detailed description of these functions and for proofs see e.g.[42].

$$\begin{aligned} \text{digamma} \quad \psi_0(x) &= \frac{d\ln(\Gamma(x))}{dx} \\ \text{trigamma} \quad \psi_1(x) &= \frac{d\psi_0(x)}{dx} \end{aligned}$$

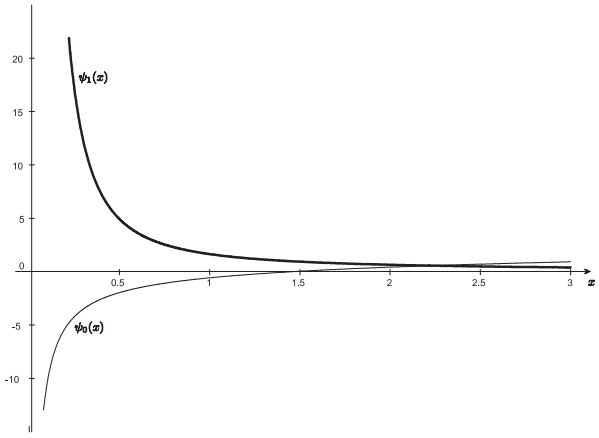


Figure B.1: Digamma and trigamma functions

Proposition 16 (Recursion for the function $\psi_0(x)$)

$$\psi_0(x+1) = \psi_0(x) + \frac{1}{x}, \ \forall x > 0$$

Proposition 17 (Properties of the function $\psi_0(x) - \ln(x)$)

Let the function $h(x) \equiv \psi_0(x) - \ln(x)$ be considered on $(0, +\infty)$. Then, it holds:

- *$h(x)$ is increasing and negative,*
- $\lim_{x\to+\infty} = 0$,
- $\lim_{x\to0^+} = -\infty$,

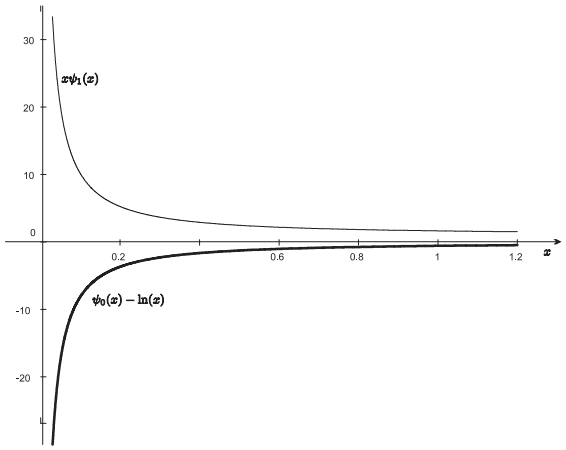


Figure B.2: Functions $h(x) \equiv \psi_0(x) - \ln(x)$ and $\psi_1(x)$

- *$h(x)$ is depicted in Figure B.2.*

Algorithm 19 (Solving equation $\psi_0(x) - \ln(x) = z$) $(x) = \mathrm{GETNU}(z)$
This algorithm numerically solves the equation $\psi_0(x) - \ln(x) = z$. The starting point of used Newton iterative method is selected using approximations of $\psi_0(x)$ so that the solution is very fast. For a detailed description of the numerical solution see [42].

Proposition 18 (Properties of the function $\psi_1(x)$)

- *$\psi_1(x)$ is decreasing and positive for positive arguments.*
- $x\psi_1(x) > 1, \forall x > 0$.
- *$\psi_1(x)$ is depicted in Figure B.2*

Appendix C

Calculus with Pdfs

C.1 General Propositions

Proposition 19 (Calculus with pdfs) *For any $(\alpha, \beta, \gamma) \in (\alpha, \beta, \gamma)^*$, the following relationships between pdfs hold.*

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	$f(\beta\|\alpha, \gamma) =$

$$= \frac{f(\alpha|\beta,\gamma)f(\beta|\gamma)}{f(\alpha|\gamma)} = \frac{f(\alpha|\beta,\gamma)f(\beta|\gamma)}{\int f(\alpha|\beta,\gamma)f(\beta|\gamma)\, d\beta} \propto f(\alpha|\beta,\gamma)f(\beta|\gamma). \quad \text{(C.1)}$$

Proposition 20 (Jensen inequality) *Let h be strictly concave function, let $f(x)$ be a pdf with a nonzero variance. Then $\mathcal{E}[h(x)]_f < h\left(\mathcal{E}[x]_f\right)$.*

Proposition 21 (Mean value transformation) *Let x be random quantity with a pdf f_x. Let y be random quantity obtained as a result of transformation $y = g(x)$, f_y is pdf of y. Then $\mathcal{E}[g(x)]_{f_x} = \mathcal{E}[y]_{f_y}$*

Proposition 22 (Covariance matrix of a mixture) *Let pdf f be a mixture of pdfs f_1 and f_2,*

$$f(x) \equiv \alpha f_1(x) + (1-\alpha) f_2(x),\ \alpha \in (0,1),\ \textit{then}$$

$$\mathbf{cov}[x]_f = \alpha \mathbf{cov}[x]_{f_1} + (1-\alpha)\mathbf{cov}[x]_{f_2} + \alpha(1-\alpha)(\mathcal{E}[x]_{f_1} - \mathcal{E}[x]_{f_2})(\mathcal{E}[x]_{f_1} - \mathcal{E}[x]_{f_2})'$$

C.1.1 Kullback-Leibler Divergence

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

$$\mathcal{D}(f||g) \equiv \int_{x^*} f(x) \ln\left(\frac{f(x)}{g(x)}\right) dx. \quad \text{(C.2)}$$

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

C.1.2 Kerridge Divergence

We can rearrange the expression of KL divergence:

$$\int_{x^*} f(x) \ln\left(\frac{f(x)}{g(x)}\right) dx = \int_{x^*} f(x) \ln\left(f(x)\right) dx - \int_{x^*} f(x) \ln\left(g(x)\right) dx \tag{C.3}$$

It is clear that the first element does not influence the result when minimizing the KL divergence with respect to the function $g(x)$. It leads to the notion. Kerridge divergence:

Let f, g be a pair of pdfs acting on a common set x^*. Then, the Kerridge divergence $\mathcal{K}(f||g)$ is defined by the formula

$$\mathcal{K}(f||g) \equiv -\int_{x^*} f(x) \ln\left(g(x)\right) dx. \tag{C.4}$$

Proposition 23 (Kerridge and Kullback-Leibler divergence) *Let $\int f ln f < +\infty$, than it holds:*

$$\operatorname{Arg\,min}_g \mathcal{D}(f||g) = \operatorname{Arg\,min}_g \mathcal{K}(f||g) \tag{C.5}$$

Proof:

$$\min_g \mathcal{D}(f||g) = \min_g \int f \ln \frac{f}{g} = \min_g \left\{ \int f \ln f - \int f \ln g \right\} = \int f \ln f + \min_g \left\{ - \int f \ln g \right\}$$

□

Proposition 24 (Kerridge divergence of a weighting sum of pdfs)

$$\mathcal{K}\left(\sum_{c=1}^{\mathring{c}} \alpha_c f_c(x) \;\middle|\middle|\; g(x) \right) = \sum_{c=1}^{\mathring{c}} \alpha_c \mathcal{K}\left(f_c(x) \;\middle|\middle|\; g(x) \right) \tag{C.6}$$

Proof:

$$\mathcal{K}\left(\sum_{c=1}^{\mathring{c}} \alpha_c f_c(x) \;\middle|\middle|\; g(x) \right) = -\int \sum_{c=1}^{\mathring{c}} \alpha_c f_c(x) \ln(g(x)) = \sum_{c=1}^{\mathring{c}} \alpha_c \left\{ -\int f_c(x) \ln(g(x)) \right\}$$

□

Proposition 25 (Kerridge divergence of a product of pdfs)

$$\mathcal{K}\left(f(x,y) \;\middle|\middle|\; g(x)v(y) \right) = \mathcal{K}\left(f(x) \;\middle|\middle|\; g(x) \right) + \mathcal{K}\left(f(y) \;\middle|\middle|\; v(y) \right), \tag{C.7}$$

where $f(x), f(y)$ are marginal probability densities of $f(x,y)$.

Proof:

$$\begin{aligned}
\mathcal{K}\left(f(x,y) \;\middle|\middle|\; g(x)v(y) \right) &= -\int f(x,y) \ln\Big(g(x)v(y) \Big) dxdy = \\
&= -\int f(x,y) \Big(\ln(g(x)) + \ln(v(y)) \Big) dxdy = \\
&= -\int f(x,y) \ln(g(x)) dxdy - \int f(x,y) \ln(v(y)) dxdy = \\
&= -\int f(x) \ln(g(x)) dx - \int f(y) \ln(v(y) dy
\end{aligned}$$

□

Proposition 26 (Kerridge divergence of product of independent pdfs)

$$\mathcal{K}\left(w(x)h(y)\ \middle|\middle|\ g(x)v(y)\right) = \mathcal{K}\left(w(x)\ \middle|\middle|\ g(x)\right) + \mathcal{K}\left(h(y)\ \middle|\middle|\ v(y)\right) \tag{C.8}$$

Proof: Simple consequence of the previous Proposition 25, with $f(x,y) \equiv w(x)h(y),\ f(x) \equiv w(x),\ f(y) \equiv h(y)$ □

C.2 Dirichlet Multivariate Pdf

C.2.1 Definition

$Di_\alpha(\kappa)$ denotes Dirichlet pdf of $\alpha \in \alpha^* \equiv \left\{\alpha_c \geq 0 : \sum_{c=1}^{\mathring{c}} \alpha_c = 1\right\}$ in the form :

$$Di_\alpha(\kappa) \equiv \frac{\prod_{c=1}^{\mathring{c}} \alpha_c^{\kappa_c - 1}}{\mathcal{B}(\kappa)},\ \mathcal{B}(\kappa) \equiv \frac{\prod_{c=1}^{\mathring{c}} \Gamma(\kappa_c)}{\Gamma(\sum_{c=1}^{\mathring{c}} \kappa_c)}.$$

Agreement 8 *We use notion "statistics" instead of "parameters" to avoid misunderstanding with unknown parameter Θ. Moreover, statistics are often used as parameters of pdfs within this text.*

C.2.2 Statistics

The statistic κ is a vector with $\mathring{c}$ positive entries.

C.2.3 Properties

$$\begin{aligned}
\mathcal{E}\left[\alpha_c | \kappa\right] &= \hat{\alpha}_c & \text{(C.9)}\\
\alpha_c Di_\alpha(\kappa) &= \hat{\alpha}_c Di_\alpha(\kappa + \delta_{\bullet,c}) & \text{(C.10)}\\
\hat{\alpha}_c &= \frac{\kappa_c}{\sum_{c=1}^{\mathring{c}} \kappa_c} & \text{(C.11)}
\end{aligned}$$

Proof:

$$\begin{aligned}
\mathcal{B}\left(\kappa + \delta_{\bullet,c}\right) &= \frac{\Gamma(\kappa_c + 1) \prod\limits_{k=1, k\neq c} \Gamma(\kappa_k)}{\Gamma\left(\sum \kappa_k + 1\right)} = \frac{\kappa_c \prod\limits_{k=1} \Gamma(\kappa_k)}{\Gamma\left(\sum \kappa_k\right) \sum \kappa_k} = \mathcal{B}\left(\kappa\right) \hat{\alpha}_c\\
\alpha_c Di_\alpha(\kappa) &= \alpha_c \frac{\prod\limits_{k=1}^{\mathring{c}} \alpha_k^{\kappa_k - 1}}{\mathcal{B}\left(\kappa\right)} = \hat{\alpha}_c \frac{\prod\limits_{k=1}^{\mathring{c}} \alpha_k^{\kappa_k - 1 + \delta_{k,c}}}{\mathcal{B}\left(\kappa + \delta_{\bullet,c}\right)} = \hat{\alpha}_c Di_\alpha(\kappa + \delta_{\bullet,c})\\
\mathcal{E}\left[\alpha_c | \kappa\right] &= \int \alpha_c Di_\alpha(\kappa) d\alpha = \hat{\alpha}_c \int Di_\alpha(\kappa + \delta_{\bullet c}) d\alpha = \hat{\alpha}_c
\end{aligned}$$

□

Proposition 27 (KL divergence of Di pdfs) *Let $f(\alpha) = Di_\alpha(\kappa)$, $\tilde{f}(\alpha) = Di_\alpha(\tilde{\kappa})$ be a pair of Dirichlet pdfs of parameters $\alpha \equiv (\alpha_1, \ldots, \alpha_{\mathring{c}}) \in \alpha^* = \left\{\alpha_c > 0,\ \sum_{c \in c^*} \alpha_c = 1\right\}$, $c^* \equiv \{1, \ldots, \mathring{c}\}$.*

Their KL divergence is given by the formula

$$\begin{aligned}
\mathcal{D}(f||\tilde{f}) &= \sum_{c=1}^{\mathring{c}} \left[\left(\kappa_c - \tilde{\kappa}_c\right) \psi_0\left(\kappa_c\right) + \ln\left(\frac{\Gamma\left(\tilde{\kappa}_c\right)}{\Gamma\left(\kappa_c\right)}\right)\right] - (\nu - \tilde{\nu})\psi_0\left(\nu\right) + \ln\left(\frac{\Gamma(\nu)}{\Gamma(\tilde{\nu})}\right)\\
\nu &\equiv \sum_{c=1}^{\mathring{c}} \kappa_c,\quad \tilde{\nu} \equiv \sum_{c=1}^{\mathring{c}} \tilde{\kappa}_c. & \text{(C.12)}
\end{aligned}$$

Moreover it holds:

$$\operatorname{Arg}\min_{\tilde{\kappa}} \mathcal{D}(f||\tilde{f}) = \operatorname{Arg}\min_{\tilde{\kappa}} \sum_{j=1}^{\mathring{c}} \left[\ln\left(\Gamma\left(\tilde{\kappa}_j\right)\right) - \tilde{\kappa}_j \psi_0\left(\kappa_j\right)\right] - \left[\ln(\Gamma(\tilde{\nu}) - \tilde{\nu}\psi_0\left(\nu\right)\right] \tag{C.13}$$

C.3 Truncated Gaussian Distribution

C.3.1 Definition

$\mathcal{TN}_x\left(M, R, a, b\right)$ denotes Truncated Gaussian pdf of scalar x of the form :

$$\mathcal{TN}_x\left(M, R, a, b\right) \equiv \begin{cases} \frac{\mathcal{N}_x(M,R)}{\mathcal{J}(M,R,a,b)} & \text{for } x \geq a \text{ and } x \leq b \\ 0 & \text{otherwise} \end{cases}$$

The normalizing integral $\mathcal{J}(M, R, a, b)$ will be discussed bellow. Truncated Gaussian distribution is obtained from Gaussian distribution by restricting its support to some interval (possibly infinite).

C.3.2 Statistics

Statistic M is scalar, Statistic R is positive scalar. Statistics a and b are (possibly infinite) scalars, fulfilling $a < b$.

C.3.3 Properties

We do not need to describe this pdf in details. There exists a simple algorithm for computing the normalizing integral $\mathcal{J}(M, R, a, b)$.

Algorithm 20 (Normalization integral of truncated Gaussian distribution)

$(\mathcal{J}) =$ TRUNCNORM$\big(M, R, a, b\big)$

There also exist a simple algorithm for evaluating mean value and variance of this distribution:

Algorithm 21 (Mean and variance of truncated Gaussian distribution)

$(E, C) =$ TRUNCSTAT$\big(M, R, a, b\big)$

For more detailed description of truncated Gaussian distribution and for the formulas for evaluating normalizing integral and the moments, see e.g. [39].

C.4 Inverse Gamma Distribution

C.4.1 Definition

$\mathcal{IG}_x\left(\alpha, \beta\right)$ denotes Inverse gamma pdf of positive scalar x of the form:

$$\mathcal{IG}_x\left(\alpha, \beta\right) \equiv \frac{x^{-(\alpha+1)} \exp\left(-\frac{\beta}{x}\right)}{\Gamma\left(\alpha\right) \beta^{-\alpha}}$$

C.4.2 Statistics

Statistics α and β are positive scalars.

C.4.3 Sampling

Sampling from inverse gamma distribution can be simply done using sampling from gamma distribution. For detailed expressions see e.g. [58].

C.5 Gauss-inverse-Wishart Pdf

C.5.1 Definition

$GiW_{\theta,r}(V,\nu)$ denotes Gauss-inverse-Wishart pdf of a vector θ and a positive scalar r of in form:

$$GiW_{\theta,r}(V,\nu) \equiv \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(V,\nu)} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(V[-1,\theta']'[-1,\theta']\right)\right\}. \tag{C.14}$$

The value of the normalization integral $\mathcal{I}(V,\nu)$ is described below, together with other properties of this important pdf.

C.5.2 Statistics

The statistic ν is positive scalar. The statistic V is square, symmetric, positive definite, extended information matrix with $\mathring{\Psi}$ rows. We often manipulate the matrix V through its $L'DL$ decomposition. (i.e. with lower triangular matrix L with unitary diagonal and diagonal matrix D, which fulfill the relation $V = L'DL$)

Let us split the information matrix V and its $L'DL$ decomposition as follows:

$$V = \begin{bmatrix} {}^{\lfloor d}V & {}^{\lfloor d\psi}V' \\ {}^{\lfloor d\psi}V & {}^{\lfloor \psi}V \end{bmatrix}, \ {}^{\lfloor d}V \text{ is scalar}, \tag{C.15}$$

$$L = \begin{bmatrix} 1 & 0 \\ {}^{\lfloor d\psi}L & {}^{\lfloor \psi}L \end{bmatrix}, \ D = \begin{bmatrix} {}^{\lfloor d}D & 0 \\ 0 & {}^{\lfloor \psi}D \end{bmatrix}, \ {}^{\lfloor d}D \text{ is scalar}. \tag{C.16}$$

Next, the matrices L and D can be equivalently expressed with help of the matrix C, vector $\hat{\theta}$ and scalar ${}^{\lfloor d}D$ as follows:

$$\hat{\theta} \equiv {}^{\lfloor \psi}L^{-1}\,{}^{\lfloor d\psi}L \equiv \text{ least-squares (LS) estimate of } \theta \tag{C.17}$$

$$C \equiv {}^{\lfloor \psi}L^{-1}\,{}^{\lfloor \psi}D^{-1}\left({}^{\lfloor \psi}L'\right)^{-1} \equiv \text{ covariance factor of LS estimate} \tag{C.18}$$

Proposition 28 (Relation between C and ${}^{\lfloor \psi}V$) *It holds:*

$$C = {}^{\lfloor \psi}V^{-1} \tag{C.19}$$

$$\hat{\theta} = {}^{\lfloor \psi}V^{-1}\,{}^{\lfloor d\psi}V \tag{C.20}$$

C.5.3 Properties

Proposition 29 (Alternative expressions of the GiW pdf) *$GiW_\Theta(V,\nu)$ has the following alternative expressions*

$$\begin{aligned} GiW_{\theta,r}(V,\nu) &\equiv \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(L,D,\nu)} \exp\left\{-\frac{1}{2r}\left[\left({}^{\lfloor \psi}L\theta - {}^{\lfloor d\psi}L\right)' {}^{\lfloor \psi}D\left({}^{\lfloor \psi}L\theta - {}^{\lfloor d\psi}L\right) + {}^{\lfloor d}D\right]\right\} \equiv \\ &\equiv \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(C,\,{}^{\lfloor d}D,\nu)} \exp\left\{-\frac{1}{2r}\left[(\theta-\hat{\theta})'C^{-1}(\theta-\hat{\theta}) + {}^{\lfloor d}D\right]\right\} \end{aligned}$$

Proposition 30 (Normalization integral)
The normalization integral can be evaluated as follows:

$$\mathcal{I}(L,D,\nu) = \Gamma(0.5\nu)\,{}^{\lfloor d}D^{-0.5\nu}\left|{}^{\lfloor \psi}D\right|^{-0.5}2^{0.5\nu}(2\pi)^{0.5\mathring{\psi}} \tag{C.21}$$

$$\mathcal{I}(C,{}^{\lfloor d}D,\nu) = \Gamma(0.5\nu)\,{}^{\lfloor d}D^{-0.5\nu}\left|C\right|^{0.5}2^{0.5\nu}(2\pi)^{0.5\mathring{\psi}}. \tag{C.22}$$

Proposition 31 (GiW moments)

$$\begin{aligned}
\mathbf{cov}\left[\theta|\,C,\hat{\theta},\nu,{}^{\lfloor d}D\right] &= \frac{{}^{\lfloor d}D}{\nu-2}C\\
\mathcal{E}\left[\frac{\theta}{r}\middle|\,C,\hat{\theta},\nu,{}^{\lfloor d}D\right] &= \frac{\nu}{{}^{\lfloor d}D}\hat{\theta}\\
\mathcal{E}\left[\frac{1}{r}\middle|\,C,\hat{\theta},\nu,{}^{\lfloor d}D\right] &= \frac{\nu}{{}^{\lfloor d}D}\\
\mathcal{E}\left[r|\,C,\hat{\theta},\nu,{}^{\lfloor d}D\right] &= \frac{{}^{\lfloor d}D}{\nu-2}\equiv\hat{r}\\
\mathbf{cov}\left[r|\,C,\hat{\theta},\nu,{}^{\lfloor d}D\right] &= \frac{2\hat{r}^2}{\nu-4}\\
\mathcal{E}\left[\ln(r)|\,C,\hat{\theta},\nu,{}^{\lfloor d}D\right] &= \ln\left(0.5\,{}^{\lfloor d}D\right)-\psi_0\left(0.5\nu\right)\\
\frac{GiW_{\theta,r}(C,\hat{\theta},{}^{\lfloor d}D,\nu)}{r} &= \frac{\nu}{{}^{\lfloor d}D}GiW_{\theta,r}(C,\hat{\theta},{}^{\lfloor d}D,\nu+2)
\end{aligned}$$

Proposition 32 (Update of matrix V) *Let the matrices $C,\hat{\theta},L,D,V$ be defined according to (C.16), (C.17), (C.18). Then, the operation*

$${}^{\lfloor \psi}\tilde{V} = {}^{\lfloor \psi}V + w_1\psi\psi',\ {}^{\lfloor d\psi}\tilde{V} = {}^{\lfloor d\psi}V + w_2 d\psi$$

can be rewritten to

$$\begin{aligned}
\tilde{C} &= C-\frac{w_1}{1+w_1\zeta}zz'\\
\tilde{\hat{\theta}} &= \hat{\theta}+\frac{w_2 d+w_1(\hat{e}-d)}{1+w_1\zeta}z,\ \textit{where}\\
z &= C\psi,\ \hat{e}=d-\psi'\hat{\theta},\ \zeta=\psi'C\psi.
\end{aligned}$$

Proposition 33 (Update of matrix V) *Let the matrices $C,\hat{\theta},L,D,V$ be defined according to (C.16),(C.17), (C.18). Then the operation*

$$\tilde{V} = V + w\Psi\Psi'$$

can be rewritten to

$$\begin{aligned}
\tilde{C} &= C-\frac{w}{1+w\zeta}zz'\\
\tilde{\hat{\theta}} &= \hat{\theta}+\frac{w\hat{e}}{1+w\zeta}z\\
{}^{\lfloor d}\tilde{D} &= {}^{\lfloor d}D+\frac{w\hat{e}^2}{1+w\zeta},\ \textit{where}\\
z &= C\psi,\ \hat{e}=d-\psi'\hat{\theta},\ \zeta=\psi'C\psi.
\end{aligned}$$

C.5.4 Sampling

Sampling from arbitrary pdf $f(\theta, r)$, can be split into two subproblems. According to the chain rule (Proposition 19):

$$f(\theta, r) = f(\theta|r)f(r).$$

First, we will generate samples from $f(r)$ and the generated samples are then used in the condition of $f(\theta|r)$.

In the case of GiW distribution, $f(r)$ is inverse gamma distribution (Section C.4), and $f(\theta|r)$ is multivariate Gaussian distribution (Section C.6).

$$f(r|C, \hat{\theta}, \lfloor^d D, \nu) = \mathcal{IG}_r\left(0.5 \lfloor^d D, 0.5\nu\right) \quad \text{(C.23)}$$

$$f(\theta|r, C, \hat{\theta}, \lfloor^d D, \nu) = \mathcal{N}_\theta\left(\hat{\theta}, rC\right) \quad \text{(C.24)}$$

Algorithm 22 (Sampling from GiW) $(r^s, \theta^s) = \text{GIWGEN}\left(C, \hat{\theta}, \lfloor^d D, \nu\right)$

1. *Take sample from inverse gamma distribution.* $r^s \sim IG(0.5 \lfloor^d D, 0.5\nu)$
2. *Take sample from multivariate Gaussian distribution.*
   $(\theta^s) = \text{GAUSSGEN}\left(r^s \times C, \hat{\theta}\right)$ *(Algorithm 23)*

Remarks 12 *We store the matrix C in $L'DL$ decomposition. The operation $r^s \times C$ then simply consist in multiplying diagonal matrix D with r^s.*

C.6 Gaussian Multivariate Pdf

C.6.1 Definition

$\mathcal{N}_\theta(M, R)$ denotes Gaussian pdf of vector θ of the form :

$$\mathcal{N}_\theta(M, R) \equiv (2\pi)^{-0.5\mathring{\theta}}|R|^{-0.5} \exp\{-0.5(\theta - M)'R(\theta - M)\}$$

C.6.2 Statistics

The statistic M is vector of length $\mathring{\theta}$. The statistic R is square, symmetric, positive definite matrix with $\mathring{\theta}$ rows.

C.6.3 Properties

$$\begin{aligned} \mathcal{E}[\theta|M, R] &= M \\ \mathbf{cov}[\theta|M, R] &= R \end{aligned}$$

Proposition 34 (Transformation of random variable) *Let θ be distributed with $\mathcal{N}(0, I)$, then random variable $A\theta + B$ is distributed with $\mathcal{N}(B, AA')$.*

C.6.4 Sampling

According to Proposition 34, we can generate sample from $\mathcal{N}(0, I)$ (let us denote it θ_0^s) and than transform it to be sample from $\mathcal{N}(M, R)$. The transformation is: $\theta^s = \sqrt{R}\theta_0^s + M$

Taking square roots of a matrix can be computationally intensive. We obviously store the matrix R in its $L'DL$ decomposition. Then $\sqrt{R} = L'\sqrt{D}$.

Algorithm 23 (sampling from Gaussian pdf) $(\theta^s) = \text{GAUSSGEN}\left(R \equiv L'DL, M\right)$

1. *Take sample from Gaussian distribution.* $\theta_0^s \sim \mathcal{N}(0, I)$
2. *evaluate* $\theta^s = L'\sqrt{D}\theta_0^s + M$

Appendix D

Estimation of Normal Factors

Because this chapter deals with only single factor at a specific time moment, we can omit the indexes $_{ic;t}$, i.e.

$$f_{ic}(d_{ic;t}|\psi_{ic;t},\Theta_{ic}) \equiv f(d|\psi,\Theta).$$

Here only the properties needed in Chapter 6 are mentioned. For a detailed description of this topic see e.g. [10].

D.1 Factor Definition

The normal parameterized factor predicts a real-valued variable d by the pdf

$$f(d|\psi,\Theta) = \mathcal{N}_d(\theta'\psi, r), \text{ where} \tag{D.1}$$

$\Theta \equiv [\theta, r] \equiv$ [regression coefficients, noise variance]

$$\mathcal{N}_d(\theta'\psi, r) \equiv (2\pi r)^{-0.5} \exp\left\{-\frac{(d-\theta'\psi)^2}{2r}\right\} \tag{D.2}$$

$$= (2\pi r)^{-0.5} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(\Psi\Psi'[-1,\theta']'[-1,\theta']\right)\right\}. \tag{D.3}$$

Normal factors belong to the exponential family, so that they possess conjugate (self-reproducing) prior. This pdf is known as Gauss-inverse-Wishart pdf (GiW). In the case of known noise variance r, the conjugate pdf is multivariate Gaussian pdf.

D.2 Form of Posterior Pdf

Gauss-inverse-Wishart pdf is conjugate pdf to normal factors.

$$\pi(\theta, r|\mathcal{S}_t) \equiv GiW_{\theta,r}(V_t, \nu_t)$$

D.3 Properties

Proposition 35 (Estimation of the normal factor) *Let the function*

$$GiW_{\theta,r}(V,\nu)\left[\mathcal{N}_d(\theta'\psi, r)\right]^w$$

have a finite integral, then it holds:

$$GiW_{\theta,r}(V,\nu)\left[\mathcal{N}_d(\theta'\psi,r)\right]^w = \frac{\mathcal{I}(V+w\Psi\Psi',\nu+w)}{(2\pi)^{0.5w}\mathcal{I}(V,\nu)} GiW_{\theta,r}(V+w\Psi\Psi',\nu+w) \tag{D.4}$$

Proof:

$$\begin{aligned} GiW_{\theta,r}(V,\nu)\left[\mathcal{N}_d(\theta'\psi,r)\right]^w &= \frac{r^{-0.5(\nu+\mathring{\psi}+2)}}{\mathcal{I}(V,\nu)} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(V[-1,\theta']'[-1,\theta']\right)\right\} \times \\ &\times (2\pi r)^{-0.5w} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left(w\Psi\Psi'[-1,\theta']'[-1,\theta']\right)\right\} = \\ &\frac{r^{-0.5(\nu+w+\mathring{\psi}+2)}}{(2\pi)^{0.5w}\mathcal{I}(V,\nu)} \exp\left\{-\frac{1}{2r}\mathrm{tr}\left([V+w\Psi\Psi'][-1,\theta']'[-1,\theta']\right)\right\} = \\ &= \frac{\mathcal{I}(V+w\Psi\Psi',\nu+w)}{(2\pi)^{0.5w}\mathcal{I}(V,\nu)} GiW_{\Theta}(V+w\Psi\Psi',\nu+w) \end{aligned}$$

□

Proposition 36 (Finiteness of integral) *The function $GiW_{\theta,r}(V,\nu)\left[\mathcal{N}_d(\theta'\psi,r)\right]^w$ has finite integral, iff*

$$w > -\nu, \quad w > -\frac{1}{\zeta}, w > -\frac{\lfloor^d D}{\hat{e}^2+\zeta\lfloor^d D}, \ where$$
$$\hat{e} = d-\psi'\theta, \ \zeta = \psi' C \psi.$$

Proof: It is simple observation that $GiW_{\theta,r}(V,\nu)\left[\mathcal{N}_d(\theta'\psi,r)\right]^w$ has finite integral, iff $V+w\Psi\Psi'$ is positive definite and $\nu+w>0$. According to Proposition 33, the operation $V+w\Psi\Psi'$ can be expressed as

$$\lfloor^d D + \frac{w\hat{e}^2}{1+w\zeta}, \ C + w_c z z', \ z = C\psi, \ w_C = -\frac{w}{1+w\zeta}$$

The first expression must be positive. $\lfloor^d D + \frac{w\hat{e}^2}{1+w\zeta} > 0$, which leads to $w > -\frac{\lfloor^d D}{\hat{e}^2+\zeta\lfloor^d D}$.

The second expression must be positive definite. C is symmetric and positive definite, hence there exists the square root $C^{\frac{1}{2}}$: $C = C^{\frac{1}{2}}C^{\frac{1}{2}}$, which is symmetric and regular. The second expression can be rewritten to

$$C^{\frac{1}{2}}\left(I + w_C C^{-\frac{1}{2}} z z' C^{-\frac{1}{2}}\right) C^{\frac{1}{2}}.$$

Thanks to Proposition 13, it suffices to prove only the positive definiteness of the matrix:

$$\left(I + w_C C^{-\frac{1}{2}} z z' C^{-\frac{1}{2}}\right).$$

According to proof of Proposition 14, the condition for the previous matrix to be positive definite is

$$0 < 1 + w_C z' C^{-\frac{1}{2}} C^{-\frac{1}{2}} z = 1 + w_C z' C^{-1} z = 1 + w_C \zeta.$$

Substituting $w_C = -\frac{w}{1+w\zeta}$ into it, we get expression $1+w\zeta > 0$. □

Proposition 37 (Factor prediction)

$$\begin{aligned} \frac{\mathcal{I}(V+w\Psi\Psi',\nu+w)}{(2\pi)^{0.5w}\mathcal{I}(V,\nu)} &= \frac{\Gamma(0.5(\nu+w))\lfloor^d D^{-0.5w}(1+w\zeta)^{-0.5}}{\pi^{0.5w}\Gamma(0.5\nu)\left(1+\frac{w\hat{e}^2}{\lfloor^d D(1+w\zeta)}\right)^{0.5(\nu+w)}}, \quad where \\ \hat{e} &\equiv d-\hat{\theta}'\psi \equiv prediction\ error \\ \zeta &\equiv \psi' C \psi, \end{aligned} \tag{D.5}$$

Proof:

According to Proposition 30, the normalizing integral can be evaluated as follows:

$$\mathcal{I}(L,D,\nu) = \Gamma(0.5\nu)\, {}^{\lfloor d}D^{-0.5\nu} \left| {}^{\lfloor \psi}D \right|^{-0.5} 2^{0.5\nu}(2\pi)^{0.5\mathring{\psi}}.$$

According to Proposition 33, the operation $\tilde{V} = V + w\Psi\Psi'$ can be rewritten to

$${}^{\lfloor d}\tilde{D} = {}^{\lfloor d}D + w\frac{\hat{e}^2}{1+w\zeta}.$$

We need to evaluate the determinant

$$\begin{aligned}
|\,{}^{\lfloor \psi}\tilde{D}| &= |\,{}^{\lfloor \psi}\tilde{V}| = |\,{}^{\lfloor \psi}V + w\psi\psi'| = \\
&= |\,{}^{\lfloor \psi}L'\sqrt{{}^{\lfloor \psi}D}||I + w\,{}^{\lfloor \psi}D^{-0.5}\,{}^{\lfloor \psi}L'^{-1}\psi\psi'\,{}^{\lfloor \psi}L^{-1}\,{}^{\lfloor \psi}D^{-0.5}||\sqrt{{}^{\lfloor \psi}D}\,{}^{\lfloor \psi}L| \overbrace{=}^{Prop.14} \\
&= (1 + w\psi'\,{}^{\lfloor \psi}L^{-1}\,{}^{\lfloor \psi}D\,{}^{\lfloor \psi}L'^{-1}\psi) = \left|{}^{\lfloor \psi}D\right|(1 + w\psi' C\psi) = \left|{}^{\lfloor \psi}D\right|(1+w\zeta)
\end{aligned}$$

Now we can use the obtained results in evaluation of the normalizing constant.

$$\begin{aligned}
J &= \frac{\mathcal{I}\left(V + w\Psi\Psi', \nu + w\right)}{(2\pi)^{0.5w}\mathcal{I}(V,\nu)} = \\
&= \frac{\Gamma(0.5\nu + 0.5w)({}^{\lfloor d}D + \frac{w\hat{e}^2}{1+w\zeta})^{-0.5\nu-0.5w} \left|{}^{\lfloor \psi}D\right|^{-0.5}(1+w\zeta)^{-0.5}2^{0.5\nu+0.5w}(2\pi)^{0.5\mathring{\psi}}}{(2\pi)^w\Gamma(0.5\nu)\,{}^{\lfloor d}D^{-0.5\nu}\left|{}^{\lfloor \psi}D\right|^{-0.5}2^{0.5\nu}(2\pi)^{0.5\mathring{\psi}}} = \\
&= \frac{\Gamma(0.5\nu + 0.5w)}{(2\pi)^{0.5w}\Gamma\left(0.5\nu\right)}\,{}^{\lfloor d}D^{-0.5w}\left(1 + \frac{w\hat{e}^2}{{}^{\lfloor d}D(1+w\zeta)}\right)^{-0.5\nu-0.5w}(1+w\zeta)^{-0.5}2^{0.5w} = \\
&= \frac{\Gamma(0.5(\nu+w))\,{}^{\lfloor d}D^{-0.5w}(1+w\zeta)^{-0.5}}{\pi^{0.5w}\Gamma(0.5\nu)\left(1 + \frac{w\hat{e}^2}{{}^{\lfloor d}D(1+w\zeta)}\right)^{0.5(\nu+w)}}
\end{aligned}$$

□

Proposition 38 (Factor prediction I)

$$\begin{aligned}
\mathcal{I}_{ic;t} &= \frac{\Gamma(0.5(\nu+1))\left[{}^{\lfloor d}D(1+\zeta)\right]^{-0.5}}{\sqrt{\pi}\Gamma(0.5\nu)\left(1 + \frac{\hat{e}^2}{{}^{\lfloor d}D(1+\zeta)}\right)^{0.5(\nu+1)}}, \quad \textit{where} \\
\hat{e} &\equiv d - \hat{\theta}'\psi \equiv \textit{prediction error} \\
\zeta &\equiv \psi' C \psi,
\end{aligned} \tag{D.6}$$

Proof: Simple use of the previous proposition with w=1. □

Index

Bibliography

[1] S. Kullback and R. Leibler, "On information and sufficiency", *Annals of Mathematical Statistics*, vol. 22, pp. 79–87, 1951.

[2] D.F. Kerridge, "Inaccuracy and inference", *Journal of Royal Statistical Society*, vol. B 23, pp. 284–294, 1961.

[3] P. Ettler, M. Kárný, and T. V. Guy, "Bayes for rolling mills: From parameter estimation to decision support", in *Preprints of the 16th World Congress of the International Federation of Automatic Control*, P. Horáček, M. Šimandl, and P. Zítek, Eds., Prague, July 2005, pp. 1–6, IFAC.

[4] J. Heřmanská and L. Jirsa, "Improved planning of radioiodine therapy for thyroid cancer — Reply", *Journal of Nuclear Medicine*, vol. 43, no. 5, pp. 714–714, May 2002, Reply to letters to editor.

[5] I. Nagy, M. Kárný, P. Nedoma, and Š. Voráčová, "Bayesian estimation of traffic lane state", *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 1, pp. 51–65, 2003.

[6] Kracík J. Kárný, M., "A normative probabilistic design of a fair governmental decision strategy - draft", Tech. Rep. 2087, ÚTIA AV ČR, Praha, 2003.

[7] V. Peterka, "Bayesian system identification", in *Trends and Progress in System Identification*, P. Eykhoff, Ed., pp. 239–304. Pergamon Press, Oxford, 1981.

[8] M. Kárný, "Algorithms for determining the model structure of a controlled system", *Kybernetika*, vol. 19, no. 2, pp. 164–178, 1983.

[9] D.M. Titterington, A.F.M. Smith, and U.E. Makov, *Statistical Analysis of Finite Mixtures*, John Wiley, New York, 1985.

[10] M. Kárný, J. Böhm, T.V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař, *Optimized Bayesian Dynamic Advising: Theory and Algorithms*, Springer, London, 2005.

[11] M. Kárný, J. Böhm, 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.

[12] N. Vlassis and A. Likas, "A kurtosis-based dynamic approach to gaussian mixture modeling", *IEEE Trans. Systems, Man, and Cybernetics, Part A*, vol. 29, pp. 393–399, 1999.

[13] M Aladjem, "Projection pursuit mixture density estimation", *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 53, no. 11, pp. 4376 – 4383, 2005.

[14] K Honda and H Ichihashi, "Regularized linear fuzzy clustering and probabilistic pca mixture models", *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, vol. 13, no. 4, pp. 508 – 516, 2005.

[15] JW Ma and SQ Fu, "On the correct convergence of the em algorithm for gaussian mixtures", *PATTERN RECOGNITION*, vol. 38, no. 12, pp. 2602 – 2611, 2005.

[16] MT Gan, M Hanmandlu, and AH Tan, "From a gaussian mixture model to additive fuzzy systems", *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, vol. 13, no. 3, pp. 303 – 316, 2005.

[17] E.L. Sutanto and K. Warwick, "Cluster analysis: An intelligent system for the process industries", in *Cybernetics and Systems '94*, Robert Trappl, Ed., Vienna, 1994, vol. 1, pp. 327–344, World Scientific.

[18] Christopher M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, Oxford, UK, 1996.

[19] C. M. Bishop, "Bayesian PCA", *Neural Information Processing Systems*, vol. 11, pp. 382–388, 1998.

[20] M. Kárný, J. Böhm, T. Guy, L. Jirsa, I. Nagy, P. Nedoma, A. Quinn, L. Tesař, R. Patel, and M. Tichý, "ProDaCTool background: theory, algorithms and software", 2002, Deliverable of the ProDaCTool project, version 1, 401 pp.

[21] 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., Prague, September 1998, pp. 77–82, ÚTIA AV ČR.

[22] A. Doucet, "On sequential monte carlo sampling methods for bayesian filtering", 1998.

[23] S. Amari, S. Ikeda, and H. Shimokawa, "Information geometry of α-projection in mean field approximation", in *Advanced Mean Field Methods*, M. Opper and D. Saad, Eds., Cambridge, Massachusetts, 2001, The MIT Press.

[24] V. Šmídl and A. Quinn, *The Variational Bayes Method in Signal Processing*, Springer, 2005.

[25] J. M. Bernardo, "Expected infromation as expected utility", *The Annals of Statistics*, vol. 7, no. 3, pp. 686–690, 1979.

[26] 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., pp. 181–193. Birkhäuser, Boston, 1997.

[27] C. M. Bishop, "Variational principal components", in *Proceedings of the Ninth International Conference on Artificial Neural Networks*, ICANN, 1999.

[28] 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.

[29] 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.

[30] M.E. Tipping and C.M. Bishop, "Probabilistic principal component analysis", *Journal of the Royal Society Series B — Statistical Methodology*, vol. 61, pp. 611–622, 1999.

[31] S. Haykin, *"Neural Networks: A Comprehensive Foundation*, Macmillan, New York, 1994.

[32] 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.

[33] 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.

[34] T.S. Ferguson, "A Bayesian analysis of some nonparametric problems", *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.

[35]

[36] Jian Qing Shi, Roderick Murray-Smith, and Mike Titterington, "Ebayesian regression and classification using mixtures of gaussian processes", *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 4, pp. 265–283, 2002.

[37] M. Kárný, N. Khailova, P. Nedoma, and J. Böhm, "Quantification of prior information revised", *Int. J. Adapt. Control Signal Process.*, vol. 15, pp. 65–84, 2001.

[38] J.O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York, 1985.

[39] V. Šmídl, *The Variational Bayes Approach in Signal Processing*, PhD thesis, Trinity College Dublin, 2004.

[40] S. J. Roberts and W. D. Penny, "Variational Bayes for generalized autoregressive models", *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2245–2257, 2002.

[41] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1972.

[42] P. Nenutil, "Approximate recursive estimation of dynamic mixture models", Tech. Rep., ÚTIA AV ČR, Praha, 2004.

[43] J.S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems", *Journal of the American Statistical Association*, vol. 93, 1998.

[44] M. Kárný, J. Kracík, I. Nagy, and P. Nedoma, "When has estimation reached a steady state? The Bayesian sequential test", *International Journal of Adaptive Control and Signal Processing*, vol. 19, no. 1, pp. 41–60, 2005.

[45] P. Nedoma and J. Andrýsek, "Mixtools. Application Program Interface. User's Guide", Tech. Rep. 2088, ÚTIA AV ČR, Praha, 2003.

[46] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost", *Machine Learning*, vol. 42, no. 3, pp. 287–320, Mar. 2001, also NeuroCOLT Technical Report NC-TR-1998-021.

[47] B. Sch, o Smola, R. Williamson, and P. Bartlett, "New support vector algorithms", 2000.

[48] S. Mika, A. Smola, and B. Schölkopf, "An improved training algorithm for kernel fisher discriminants", in *Proceedings AISTATS 2001.* 2001, Morgan Kaufmann.

[49] J. Andrýsek, "On identification of probabilistic mixture models with dynamic weights", Abstracts of accepted papers of the 6th International Phd Workshop on Systems and Control. Young Generation Viewpoint, October 4-8 2005, http://www-e2.ijs.si/PhDWorkshop/2005.

[50] J. Andrýsek, "Projection based algorithms for estimation of complex models", in *Proceedings of the 5th International PhD Workshop on Systems and Control - a Young Generation Viewpoint*, Budapest, September 2004, pp. 5–10, Hungarian Academy of Sciences.

[51] J. Andrýsek, "Approximate recursive Bayesian estimation of dynamic probabilistic mixtures", in *Multiple Participant Decision Making*, J. Andrýsek, M. Kárný, and J. Kracík, Eds., Adelaide, May 2004, pp. 39–54, Advanced Knowledge International.

[52] J. Andrýsek, "Projection Based Estimation of Dynamic Probabilistic Mixtures", Tech. Rep. 2098, ÚTIA AV ČR, Praha, 2004.

[53] M. Kárný, J. Andrýsek, P. Nedoma, J. Böhm, and T.V. Guy, "On Generalized Factors in Mixture Learning and Prediction", Tech. Rep. 2094, ÚTIA AV ČR, Praha, 2003.

[54] P. Nedoma and J. Andrýsek, *Mixtools Application Program Interface. (Program)*, ÚTIA AV ČR, Praha, 2003.

[55] 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.

[56] A. Quinn, P. Ettler, L. Jirsa, I. Nagy, and P. Nedoma, "Probabilistic advisory systems for data-intensive applications", *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 2, pp. 133–148, 2003.

[57] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus*, Wiley, 2001.

[58] Luc Devroye, *Non-Uniform Random Variate Generation*, Springer, April 1986.