Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autumn Semester 2023 Prof. H.-A. Loeliger

Master's Thesis

On Composite NUV Priors and Hierarchical Models

Luca Iten

Advisor: Prof. H.-A. Loeliger
Co-Advisors: Hugo Aguettaz and Alessio Lukaj

Acknowledgments

This report, as well as the whole project it tries to communicate, has been made possible by the enormous help from the following people. First of all, I would like to thank my main supervisor, Prof. H.-A. Loeliger. His expertise and insightful inputs generated a guided environment of continuous progress, effectively shaping almost all interesting observations I have been able to make. Furthermore, a big thanks goes to my two co-supervisors, Hugo Aguettaz and Alessio Lukaj. They both made me feel very welcome and appreciated in the lab and were always available to help me with all kinds of nonsense a master's student like me can possibly come up with. To all of you,

THANK YOU!

Zürich, 18 March 2024

Luca Iten

Abstract

This report describes two novel NUV-based methods applicable to a great variety of signal-analysis related applications. The first derived algorithm works as a model selector mechanism, effectively estimating which sections of given observations most likely originated from which of the specified models. It makes heavy use of a specially designed One-Hot NUV prior, giving it the desired behaviour. The method is shown to work by applying it to a variety of different problems, most notably to fit a finite amount of potentially unknown constant levels and to construct a Piecewise Constant (PWC) model with a known base level. The second described method provides a way to estimate the potentially evolving covariance matrices of zero-mean Gaussian noise based on a small number of observations. This covariance estimation method relies on a specialised lower-level model, which itself can easily be modelled by a great variety of well known state-space models. Finally, this second method is applied to a hierarchical model, yielding accurate estimates of both, the means and noise covariance matrices for the given observations. Note that all described algorithms have a computational complexity linear in the number of observations, making them a very powerful tool compared to competing methods.

Zusammenfassung

Dieser Bericht beschreibt zwei neue NUV-basierte Methoden, die auf eine Vielzahl von praktischen Anwendungen zur Signalanalyse anwendbar sind. Der erste Algorithmus fungiert als ein Modellauswahlsystem, das schätzt, welche Abschnitte gegebener Beobachtungen wahrscheinlich von welchem der spezifizierten Modelle generiert wurden. Dabei wird insbesondere von einem speziell entworfenen One-Hot NUV-prior Gebrauch gemacht, welches dem System das gewünschte Verhalten verleiht. Die Funktionalität der Methode wird zudem durch die konkrete Anwendung auf eine Vielzahl verschiedener Probleme gezeigt. Insbesondere wird eine endliche Anzahl von konstanten Levels auf die gegebenen Beobachtungen gefittet. Der zweite beschriebene Algorithmus ermöglicht es nun, die sich verändernden Kovarianzmatrizen von Gaussschem Rauschen, basierend auf einer kleinen Anzahl von Beobachtungen, zu schätzen. Diese Methode stützt sich auf ein speziell entworfenes Modell, das in gewisser Weise die zu schätzenden Kovarianzmatrizen beeinflusst und einfach mit Hilfe von bekannten Zustandsraummodellen beschrieben werden kann. Schliesslich wird diese Methode auf ein hierarchisches Modell angewendet. Das Resultat sind genaue Schätzungen sowohl der Mittelwerte als auch der Rauschkovarianzmatrizen für die gegebenen Beobachtungen. Zudem soll hier erwähnt sein, dass alle beschriebenen Algorithmen von linearer Komplexität sind, was sie zu einem sehr leistungsfähigen Werkzeug im Vergleich zu konkurrierenden Methoden macht.

Contents

Chapter 1

Introduction

The general goal of this project was to investigate novel NUV-based approaches to tackle the two subsequently described problem settings. This includes the development, implementation, and simulation of practical algorithms. The most important and interesting findings made in this process are summarized in this report. Furthermore, the thereby developed code is publicly available here, in particular including a Jupyter notebook file transparently reproducing all simulations presented in this report.

In the following, Chapter 2 introduces the fundamental concepts necessary to understand the later chapters. This in particular includes an introduction into statistical models, their respective factor graphs, and the concept of Normals with Unknown Variances (NUVs). Chapter 3 then tackles the first of the two problem settings (stated in the next Section), developing a powerful model selector mechanism. The derivation of this method also includes the development of a novel One-Hot NUV prior, the applications of which are not only limited to this model selector mechanism, but are far more general. This chapter further applies the described mechanism to a variety of problems, showcasing its huge potential. The second problem setting is discussed in Chapter 4. There, a method to estimate potentially evolving covariance matrices of zero-mean Gaussian noise is derived. This method is further applied to a hierarchical model, which is able estimate both, the means and noise covariance matrices of some given observations. Finally, Chapter 5 concludes the report.

1.1 Problem Settings

Problem 1: Fitting Data to Various Models Even though modelling data by a constant probabilistic source is very powerful, this approach is limited to few real-world applications. A more general setting is to assume that various statistical models are interchangeably generating the received data. This point of view however requires a method to estimate which sections of the received data have been generated by which model, effectively creating a metric assigning each observation to a model. Ideally, this metric should have similar properties to the soft weighting factors in Expectation Maximization (EM).

The first step of this project is therefore to develop such a method based on the idea of NUVs. For a starting point, it should be possible to fit the outputs of a finite set of known models to some given data. In other words, the described method should determine which sections of the observations in question can be best described by which model.

Problem 2: Covariance Estimation Another fundamental problem when processing data is the estimation of covariance matrices. This can be important to detect correlations between different measurements, to quantify the uncertainty about an estimation, or simply to characterise observed noise. Doing so turns out to be quite tricky, especially if the covariance matrices in question are assumed to evolve over time.

With this motivation in mind, the second part of the project should focus on the development of a NUV-based method to estimate potentially evolving covariance matrices of additive Gaussian noise. Thereby, the results presented in [1] should be used as a starting point to develop a similar method for the multivariate case.

Chapter 2

Background

This Chapter introduces the most important concepts needed to understand the work presented in this report. This includes some definitions regarding the used notation of statistical models, a list of the most important nodes in factor graphs, and a brief overview of some selected NUV priors. The last Section further derives a complete algorithm to estimate the outputs of a Piecewise Constant (PWC) model, which later works as a basis to tackle more advanced estimation tasks. Note that the content of this Chapter should not be viewed as a complete introduction into the respective topics, but is solely intended to give the reader an overview of the necessary basics to understand the subsequent Chapters.

2.1 Statistical Models and Their Factor Graph Representation

The fundamental concepts of data processing and, in particular, model based estimation are covered by many different sources and are not repeated in this report (a good foundation is, for example, given in [2]). For clarification however, this Section introduces the used notations as well as the corresponding factor graph representations.

2.1.1 Notation

All models investigated in this report can be characterized by some hidden states, their corresponding observations, and optional inputs. The number of hidden states and observations is denoted by N, causing the number of inputs to be $N-1$ (one input in between two subsequent states). To indicate / differentiate between the different samples, all quantities are indexed by a subscript $i \in \{1, \dots, N\}$ (respectively up to $N-1$ for the inputs). Usually, the hidden states are denoted by X_i, the observations by Y_i, and the inputs by U_i. The model investigated in Section 2.3 later exemplifies these definitions.

Next, it is noted that this report only investigates settings where Gaussian message passing algorithms can be used (variation of believe propagation, [3]). Accordingly, all previously mentioned quantities are assumed to be normally distributed. This means that the distribution of each data point at each time index can be perfectly described by either the respective means and covariance matrices, or dual means and precision matrices.

It is further pointed out that the work presented in the following Chapters heavily relies on the notion of factor graphs (in particular Forney factor graphs, [4]). They provide a neat way to communicate the structure of complex statistical models by visualizing their respective probability distribution functions. Furthermore, they can be used to derive powerful Gaussian message passing algorithms ([5], [6]), a fact that has heavily been exploited in this project. Note that in this entire report, the depicted directions of the arrows are reflected in the corresponding mathematical notation. For example, the forward messages through X_i are denoted by $\overrightarrow{m}_{X_i}$ (mean) and $\overrightarrow{V}_{X_i}$ (covariance), where as the backward messages are represented by $\overleftarrow{m}_{X_i}$ and $\overleftarrow{V}_{X_i}$. The resulting posterior estimates are denoted by $\hat{m}_{X_i}$ and $\hat{V}_{X_i}$. As previously mentioned, the same information can also be represented by the dual mean and precision matrix messages, which are denoted in the same way, but m is replaced by ξ and V by W.

2.1.2 Important Nodes in Factor Graphs

The probably most commonly used node in factor graphs is the equality constraint shown in Figure 2.1. It restricts all applied quantities to be strictly equal. The resulting Gaussian messages through this node are given in [6], Table 1.

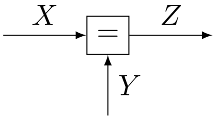


Figure 2.1: Equality constraint, i.e., $X = Y = Z$.

Another node that is often used is the sum of two quantities, the symbol of which is shown in Figure 2.2. It restricts the sum of the two incoming quantities (in the depicted case X and Y) to be equal to the outgoing quantity (labelled by Z). The resulting Gaussian messages through this node are given in [6], Table 2.

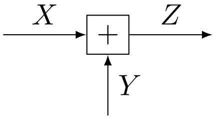


Figure 2.2: Summation constraint, i.e., $X + Y = Z$.

Similar to the sum of two quantities, their product can also be expressed. A factor graph of such a multiplication node is shown in Figure 2.3, where the product of the two incoming quantities (in the depicted case X and Y) is equal to the outgoing quantity (again, labelled by Z). Because the product of two normal distributions is no longer a normal distribution itself, there exist no expressions for the Gaussian messages through this constraint. Furthermore, it is noted that there exists no notion to explicitly state the order of the two ingoing messages. This poses a problem as matrix-matrix multiplications generally are not commutative. For the remainder of this report, this issue is omitted by always assuming the ordering is implicitly fixed by the dimensions of the quantities. This is particularly important for the case discussed in Figure 2.5b.

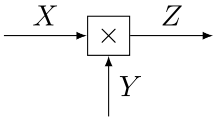


Figure 2.3: Multiplication node, i.e., $XY = Z$ or $YX = Z$.

Observations or, more generally, perfectly known quantities are denoted by small solid-black boxes as shown in Figure 2.4. The resulting Gaussian messages through Y are $\overrightarrow{m}_Y = y$ and $\overrightarrow{V}_Y = \mathbf{0}$, where $\mathbf{0}$ is an all-zero square matrix of appropriate dimension. It illustrates how the covariance matrix in a Gaussian distribution can be interpreted as the uncertainty about the quantity in question, as an observation eradicates all uncertainty (i.e., sets $\overrightarrow{V}_Y$ to zero). Note that the corresponding precision matrix message is not well defined.

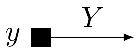
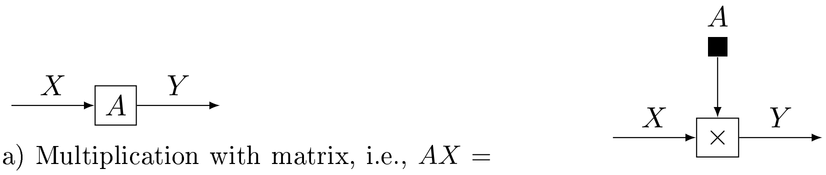


Figure 2.4: Perfectly known quantity.

Finally, there is a neat Gaussian message representation for quantities that are multiplied by a matrix A, i.e., $Y = AX$. The corresponding factor graph representation is shown in Subfigure 2.5a. The resulting Gaussian messages through this node can be found in [6], Table 3. Note that the representation in Subfigure 2.5b is equivalent. Therefore, fixing one of the two ingoing quantities into a multiplication node makes the probability distribution of its resulting outgoing quantity normally distributed. This observation will later be used to perform Gaussian message passing in factor graphs with multiplication nodes.



(a) Multiplication with matrix, i.e., $AX = Y$.

(b) Alternative representation.

2.2 Normals with Unknown Variances (NUVs)

The fundamental idea of NUVs is to represent priors of a Gaussian Random Vector by a multivariate normal distribution, where the variances (i.e., covariance in the multivariate case) are assumed to be random variables themselves with dedicated prior distributions. The priors of these unknown variances are constructed in such a way that maximizing the posterior probabilities of the variances (for fixed estimates of the means) effectively shapes the distribution of the Gaussian vector. In other words, the prior on the Gaussian random vector is expressed by a function which itself is maximised over some engineered parameter. The fundamental idea then is to iteratively improve the estimates of the variances and the means. This method is called Iteratively Reweighed Least Squares (IRWLS). In conclusion, NUVs provide a way to express non-Gaussian priors in a Gaussian way, making it possible to still use Gaussian message passing algorithms in the corresponding system!

In the following, some of the already known and subsequently used NUV priors are described. In particular, Subsections 2.2.1 and 2.2.2 introduce NUV priors that are often used to induce sparsity. Especially the Log-Cost prior is very well investigated due to its neat properties. Subsection 2.2.3 then explains a method to include positivity constraints into models. Note that this is not a complete description of their respective properties, but enables the reader to understand the subsequent derivations in Chapters 3 and 4. For more information regarding NUVs in general, refer to [2], [6], [7], and [8].

2.2.1 Laplace Prior

The Laplace NUV prior describes a method to apply the following prior to a Gaussian random vector X_i,

$$\rho(x_i) = \exp(-\beta||x_i||), \quad x_i \in \mathbb{R}^D \,, \tag{2.1}$$

where $||\cdot||$ denotes the L_2 norm. The value of $\beta \in \mathbb{R}$ can be used to "tune" this prior, i.e., it determines the direction and steepness of the slope in the corresponding cost function (Figure 2.6a). Obviously, Equation (2.1) does not describe a probability density, let alone a Gaussian distribution. For $\beta < 0$, Equation (2.1) is not even bounded. Still, this prior can be implemented by the Laplace NUV prior. According to [1], the AM update rule for the scalar variance is

$$\sigma_i^2 = \frac{||\hat{m}_{X_i}||}{\beta} \,. \tag{2.2}$$

The resulting Gaussian messages generated by this prior are

$$\overrightarrow{m}_{X_i} = \mathbf{0}_D \tag{2.3}$$

$$\overrightarrow{V}_{X_i} = \sigma_i^2 \cdot \mathbf{I}_D \tag{2.4}$$

or

$$\overrightarrow{\xi}_{X_i} = \mathbf{0}_D \tag{2.5}$$

$$\overrightarrow{W}_{X_i} = \frac{1}{\sigma_i^2} \cdot \mathbf{I}_D \,. \tag{2.6}$$

Here, $\mathbf{0}_D$ denotes the all-zero vector and $\mathbf{I}_D$ the identity matrix, both of dimension D. Note that this applies for positive and negative values of β.

2.2.2 Log-Cost Prior

Similar to the Laplace NUV prior, the Log-Cost prior also describes a cost-function growing in the magnitude of its applied random vector X_i. In particular, it implements the following prior

$$\rho(x_i) = \frac{1}{||x_i||^\beta}, \quad x_i \in \mathbb{R}^D . \tag{2.7}$$

According to [1], the EM update rule is

$$\sigma_i^2 = \frac{\mathrm{Tr}\left\{\hat{V}_{X_i}\right\} + ||\hat{m}_{X_i}||^2}{\beta} . \tag{2.8}$$

Here, $\mathrm{Tr}\{\cdot\}$ denotes the trace of the applied matrix. Furthermore, $\beta \in \mathbb{R} \setminus \{0\}$ can again be seen as a "tuning" factor affecting the corresponding cost function (Figure 2.6b). Note that there also exists an AM update rule. However, the work presented in this report always relied on EM update where possible due to its better performance observed in prior work. The resulting Gaussian messages generated by this prior are again

$$\overrightarrow{m}_{X_i} = \mathbf{0}_D \tag{2.9}$$

$$\overrightarrow{V}_{X_i} = \sigma_i^2 \cdot \mathbf{I}_D \tag{2.10}$$

or

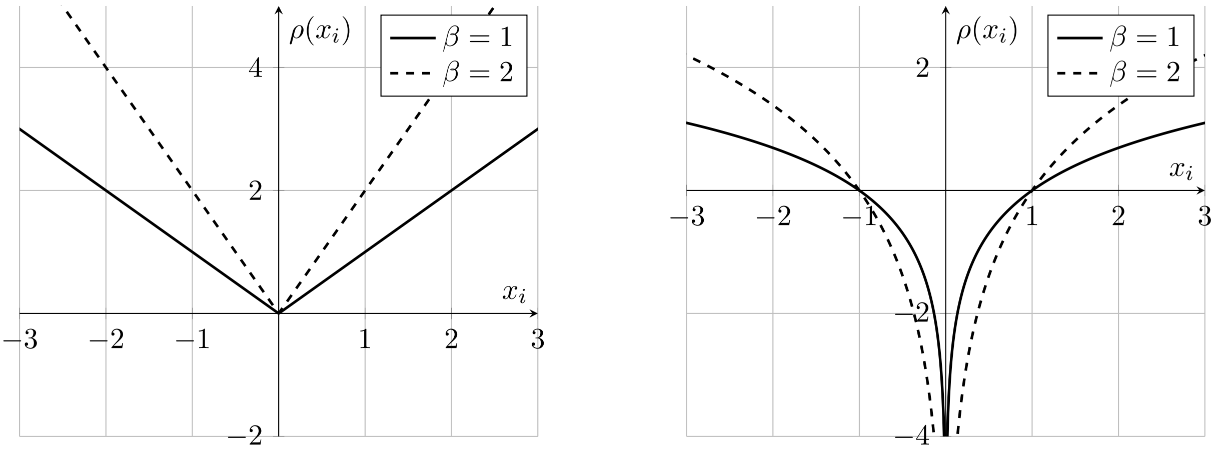
$$\overrightarrow{\xi}_{X_i} = \mathbf{0}_D \tag{2.11}$$

$$\overrightarrow{W}_{X_i} = \frac{1}{\sigma_i^2} \cdot \mathbf{I}_D . \tag{2.12}$$

Note that for the special case where β is chosen equal to the dimension of the applied random vector (i.e., D), this prior is also referred to as the Plain NUV. The motivation behind this name comes from some special properties of the constructed prior for the variance, which reduces to a constant for this particular case.

2.2.3 Positivity Constraint

Many of the investigated models in this report contain quantities that are required (or known) to be strictly positive, i.e., they require some Positivity prior. All of these priors are implemented by the same NUV prior described here (note that there exist other options too). The idea is to apply a Half-space prior (described in [9]) on each element of $X_i \in \mathbb{R}^D$, forcing all of them to be ≥ 0. Figure 2.7 depicts a factor graph of this situation, where the matrices C_d, $d \in \{1, \ldots, D\}$ are row matrices of length D, whose elements are all 0 except for the d-th element, which is 1.



(a) Cost function of the Laplace prior, $\rho(x_i)$ defined in (2.1).

(b) Cost function of the Log-Cost prior, $\rho(x_i)$ defined in (2.7).

Figure 2.6: Plots of different cost functions for the scalar case. A cost function is defined as $-\ln \rho(x_i)$. Note how the cost associated with the Log-Cost prior penalises outliers much less than the Laplace prior (extensive discussion in [2]).

According to [9], the Gaussian messages generated by the ≥ 0 priors are

$$\overleftarrow{m}_{X_{i,d}} = |\hat{m}_{X_{i,d}}| \tag{2.13}$$

$$\overleftarrow{\sigma}_{X_{i,d}} = \frac{|\hat{m}_{X_{i,d}}|}{\beta} . \tag{2.14}$$

Note that these are the scalar messages through the scalar quantities $X_{i,d}$, as it is indicated by the second subscript d. Together with the Gaussian message passing rules through a matrix multiplication node (discussed in Subsection 2.1.2), the resulting messages generated by the Positivity constraint are

$$\overrightarrow{m}_{X_i} = \begin{bmatrix} |\hat{m}_{X_{i,1}}| \\ |\hat{m}_{X_{i,2}}| \\ \vdots \\ |\hat{m}_{X_{i,D}}| \end{bmatrix} \tag{2.15}$$

$$\overrightarrow{V}_{X_i} = \begin{bmatrix} \frac{|\hat{m}_{X_{i,1}}|}{\beta} & 0 & \dots & 0 \\ 0 & \frac{|\hat{m}_{X_{i,2}}|}{\beta} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{|\hat{m}_{X_{i,D}}|}{\beta} \end{bmatrix} \tag{2.16}$$

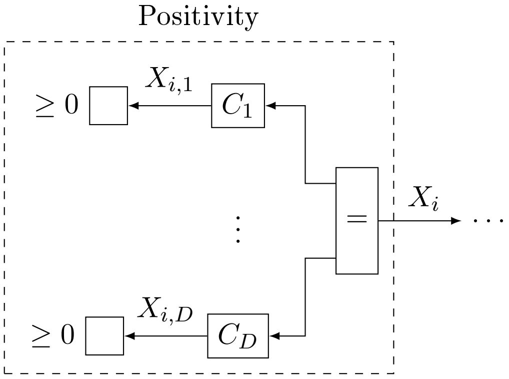


Figure 2.7: Positivity prior, forcing each element of the applied random vector to be ≥ 0.

or

$$\overrightarrow{\xi}_{X_i} = \begin{bmatrix} \beta \\ \beta \\ \vdots \\ \beta \end{bmatrix} \tag{2.17}$$

$$\overrightarrow{W}_{X_i} = \begin{bmatrix} \frac{\beta}{|\hat{m}_{X_{i,1}}|} & 0 & \dots & 0 \\ 0 & \frac{\beta}{|\hat{m}_{X_{i,2}}|} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\beta}{|\hat{m}_{X_{i,D}}|} \end{bmatrix}. \tag{2.18}$$

Again, the parameter $\beta > 0$ works as a tuning factor. In contrast to the tuning factors of the Laplace and Log-Cost prior, this β is chosen to be always strictly positive in this report (choosing it ≤ 0 would be nonsensical in this context).

2.3 Piecewise Constant Model

This Section concludes Chapter 2 by applying some of the previously introduced concepts to a practical example. In particular, it tackles the problem of fitting a piecewise constant (PWC) line to some kind of data. This is a very basic task and occurs in many other problems investigated later in this report.

2.3.1 Setup of PWC Model

A factor graph of the investigated PWC model is shown in Figure 2.8. It depicts N hidden states X_i, their noisy observations Y_i, and the inputs U_i. Note the left-most node labelled by $\rho(x_1')$ denoting the prior knowledge about the initial state X_1'. To

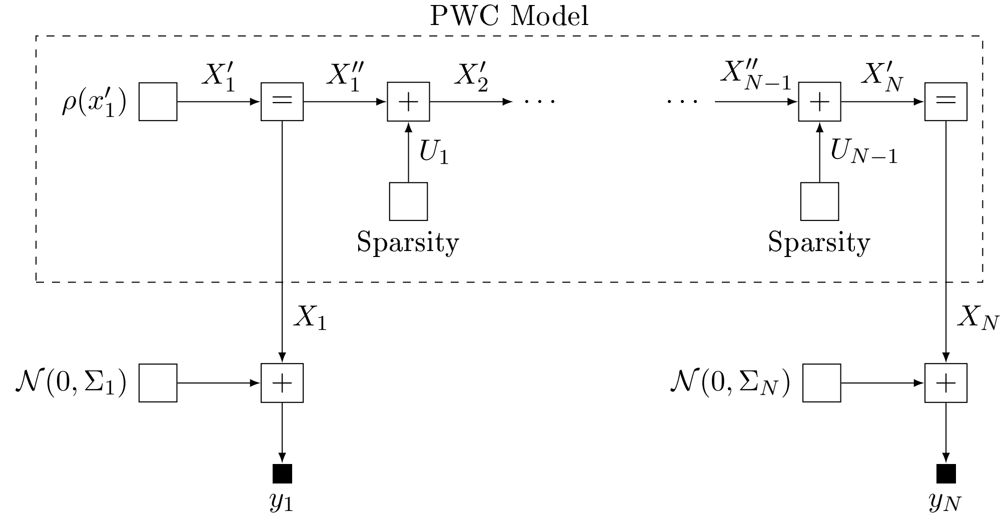


Figure 2.8: Factor graph of PWC model.

achieve the desired PWC effect, a Sparsity NUV prior is applied to each input U_i. To strongly enhance U_i being close to zero without punishing deviations too heavily, a Log-Cost NUV (Subsection 2.2.2) is chosen. Further note that the covariance matrices of the observation noise Σ_i are assumed to be known.

From Figure 2.8 it is obvious that the backward messages through X_i (i.e., the messages passed to the PWC model) are

$$\overleftarrow{m}_{X_i} = y_i \tag{2.19}$$

$$\overleftarrow{V}_{X_i} = \Sigma_i \tag{2.20}$$

or, alternatively,

$$\overleftarrow{W}_{X_i} = \Sigma_i^{-1} \tag{2.21}$$

$$\overleftarrow{\xi}_{X_i} = \Sigma_i^{-1} y_i \,. \tag{2.22}$$

At this point it probably seems unnecessary to express the messages in the latter representation. However, the following Subsections describe how the values of X_i can be estimated in such a PWC model (indicated by the dotted line) irrespective of how the incoming backward message through X_i have been generated. This is useful because depending on their origin, either of the representations can be the more natural choice.

2.3.2 Message Passing with "Conventional" Representation

If the posterior estimates of X_i should be calculated using their "conventional" message representations, forward- / backward- message passing is most efficiently handled by the

Modified Bryson-Frazier (MBF) smoother (described in [6]). Accordingly, the forward messages are recursively computed as

$$\overrightarrow{m}_{X_i''} = \overrightarrow{m}_{X_i'} + \overrightarrow{V}_{X_i'} G_i \Big(\overleftarrow{m}_{X_i} - \overrightarrow{m}_{X_i'} \Big) \tag{2.23}$$

$$\overrightarrow{V}_{X_i''} = \overrightarrow{V}_{X_i'} - \overrightarrow{V}_{X_i'} G_i \overrightarrow{V}_{X_i'} \tag{2.24}$$

$$\overrightarrow{m}_{X_{i+1}'} = \overrightarrow{m}_{X_i''} \tag{2.25}$$

$$\overrightarrow{V}_{X_{i+1}'} = \overrightarrow{V}_{X_i''} + \overrightarrow{V}_{U_i} \,, \tag{2.26}$$

where

$$G_i = \Big(\overleftarrow{V}_{X_i} + \overrightarrow{V}_{X_i'} \Big)^{-1} \tag{2.27}$$

$$F_i = \mathbf{I}_D - \overrightarrow{V}_{X_i'} G_i \,. \tag{2.28}$$

Thereby, $\overrightarrow{m}_{X_1'}$ and $\overrightarrow{V}_{X_1'}$ should be initialized to incorporate any prior knowledge given in $\rho(x_1')$. Next, the backward recursion is computed as

$$\tilde{W}_{X_{i-1}'} = F_{i-1}^{\mathsf{T}} \tilde{W}_{X_i'} F_{i-1} + G_{i-1} \tag{2.29}$$

$$\tilde{\xi}_{X_{i-1}'} = F_{i-1}^{\mathsf{T}} \tilde{\xi}_{X_i'} - G_{i-1} \Big(\overleftarrow{m}_{X_{i-1}} - \overrightarrow{m}_{X_{i-1}'} \Big) \,, \tag{2.30}$$

where $\tilde{W}_{X_N'}$ and $\tilde{\xi}_{X_N'}$ have to be initialized as

$$\tilde{W}_{X_N'} = \Big(\overrightarrow{V}_{X_N'} + \overleftarrow{V}_{X_N} \Big)^{-1} \tag{2.31}$$

$$\tilde{\xi}_{X_N'} = \tilde{W}_{X_N'} \Big(\overrightarrow{m}_{X_N'} - \overleftarrow{m}_{X_N} \Big) \,. \tag{2.32}$$

The posterior estimates of X_i and U_i can then be calculated as

$$\hat{m}_{X_i} = \overrightarrow{m}_{X_i'} - \overrightarrow{V}_{X_i'} \tilde{\xi}_{X_i'} \tag{2.33}$$

$$\hat{V}_{X_i} = \overrightarrow{V}_{X_i'} - \overrightarrow{V}_{X_i'} \tilde{W}_{X_i'} \overrightarrow{V}_{X_i'} \tag{2.34}$$

$$\hat{m}_{U_i} = -\overrightarrow{V}_{U_i} \tilde{\xi}_{X_{i+1}'} \tag{2.35}$$

$$\hat{V}_{U_i} = \overrightarrow{V}_{U_i} - \overrightarrow{V}_{U_i} \tilde{W}_{X_{i+1}'} \overrightarrow{V}_{U_i} \,. \tag{2.36}$$

This concludes one iteration of Iteratively Reweighted Least Squares (IRWLS) as described in [2]. Note that the "reweighting" part happens when computing the messages generated by the sparsifying prior (i.e., the values of $\overrightarrow{V}_{U_i}$), either by the update rules described in Subsection 2.2.1 or 2.2.2. These updates rely on the posterior estimates of U_i, which are then updated in Equations (2.35) and (2.36). IRWLS is usually performed until the estimates are considered to have converged.

2.3.3 Message Passing with "Dual" Representation

This approach is in some sense 'dual' to the algorithm described in the previous Subsection. It applies when the ingoing messages to the PWC model (i.e., backward messages through X_i in Figure 2.8) are given by their dual-mean and precision matrix representations. The following expressions are derived from the Backward Information Filter, forward with Marginals (BIFM) as described in [6]. The backward messages are recursively calculated as

$$
\begin{aligned}
\overleftarrow{\xi}_{X'_i} &= \overleftarrow{\xi}_{X''_i} + \overleftarrow{\xi}_{X_i} && (2.37)\\
\overleftarrow{W}_{X'_i} &= \overleftarrow{W}_{X''_i} + \overleftarrow{W}_{X_i} && (2.38)\\
\overleftarrow{\xi}_{X''_{i-1}} &= \overleftarrow{\xi}_{X'_i} - \overleftarrow{W}_{X'_i}\ddot{h}_i && (2.39)\\
\overleftarrow{W}_{X''_{i-1}} &= \overleftarrow{W}_{X'_i} - \overleftarrow{W}_{X'_i}\ddot{H}_i\overleftarrow{W}_{X'_i}\,, && (2.40)
\end{aligned}
$$

where

$$
\begin{aligned}
\ddot{H}_i &= \left(\overrightarrow{W}_{U_{i-1}} + \overleftarrow{W}_{X'_i}\right)^{-1} && (2.41)\\
\ddot{h}_i &= \ddot{H}_i\overleftarrow{\xi}_{X'_i}\,. && (2.42)
\end{aligned}
$$

$\overleftarrow{\xi}_{X''_N}$ and $\overleftarrow{W}_{X''_N}$ are both initialized to all-zero vectors / matrices. Next, the forward recursion is computed as

$$
\begin{aligned}
\tilde{F}_i &= \mathbf{I}_D - \overleftarrow{W}_{X'_i}\ddot{H}_i && (2.43)\\
\hat{m}_{X_i} &= \tilde{F}_i^{\mathsf{T}}\hat{m}_{X_{i-1}} + \ddot{h}_i && (2.44)\\
\hat{V}_{X_i} &= \tilde{F}_i^{\mathsf{T}}\hat{V}_{X_{i-1}}\tilde{F}_i + \ddot{H}_i\,. && (2.45)
\end{aligned}
$$

To incorporate given prior knowledge, $\hat{m}_{X_1}$ and $\hat{V}_{X_1}$ are initialized as

$$
\begin{aligned}
\hat{V}_{X_1} &= \left(\overrightarrow{W}_{X'_1} + \overleftarrow{W}_{X'_1}\right)^{-1} && (2.46)\\
\hat{m}_{X_1} &= \hat{V}_{X_1}\left(\overrightarrow{\xi}_{X'_1} + \overleftarrow{\xi}_{X'_1}\right), && (2.47)
\end{aligned}
$$

where $\overrightarrow{\xi}_{X'_1}$ and $\overrightarrow{W}_{X'_1}$ directly follow from the values specifying $\rho(x'_1)$ in Figure 2.8. In parallel, the following messages are also computed

$$
\begin{aligned}
\tilde{\xi}_{U_{i-1}} &= \tilde{\xi}_{X'_i} && (2.48)\\
&= \overleftarrow{W}_{X'_i}\hat{m}_{X_i} - \overleftarrow{\xi}_{X'_i} && (2.49)\\
\tilde{W}_{U_{i-1}} &= \overrightarrow{\tilde{W}}_{X'_i} && (2.50)\\
&= \overleftarrow{W}_{X'_i} - \overleftarrow{W}_{X'_i}\hat{V}_{X_i}\overleftarrow{W}_{X'_i}\,. && (2.51)
\end{aligned}
$$

Note that the expressions given in Equations (2.44) and (2.45) already are the posterior estimates of X_i. The posterior estimates of the sparse inputs U_i are calculated as

$$\hat{m}_{U_i} = -\overrightarrow{V}_{U_i}\tilde{\xi}_{U_i} \tag{2.52}$$

$$\hat{V}_{U_i} = \overrightarrow{V}_{U_i} - \overrightarrow{V}_{U_i}\tilde{W}_{U_i}\overrightarrow{V}_{U_i} \,. \tag{2.53}$$

Again, this concludes one iteration of IRWLS. Similar to the previous Subsection, the “reweighting” part also happens when computing the messages generated by the sparsifying prior (i.e., the values of $\overrightarrow{W}_{U_i}$).

Chapter 3

Model Selector

This Chapter tackles the first of the two problem settings stated in the Introduction. It says that for a pre-defined set of models, a method should be developed to determine which sections of the given observations have most likely been generated by which of these models. Such a method has indeed been found, leading to impressive results presented at the end of this Chapter.

Fundamentally, the developed method, from now on referred to as the model selector mechanism, builds on a PWC model (Section 2.3) whose hidden states S_i, $i \in \{1, \ldots, N\}$ indicate which of the specified M models is selected in each time index i. In particular, the M dimensional random vectors S_i have the following three properties:

1. The elements at each time index have to sum up to 1, i.e., $\sum_{m=1}^{M} S_{i,m} = 1$.
2. All its elements lie between 0 and 1, i.e., $S_{i,m} \in [0, 1]$, $m \in \{1, \ldots, M\}$.
3. All-$\{0, 1\}$ solutions should be preferred.

Considering these points, the vectors S_i should ultimately be interpretable as weighting factors, similar to those found in EM algorithms.

In the following, Section 3.1 derives a NUV prior needed to implement the previously listed properties of S_i. Section 3.2 then describes the actual developed model selector mechanism. Finally, Section 3.3 discusses a selection of possible applications.

3.1 One-Hot NUV Prior

This Section describes a novel NUV prior implementing the three previously described properties. Its proposed design is shown in Figure 3.1. Note that the applied Gaussian random vector X_i is of dimension M.

The first desired property (i.e., restricting the sum over all elements of X_i to 1) is achieved by the left most part of the shown factor graph. In particular, $X_i^{\mathbb{1}}$ is multiplied

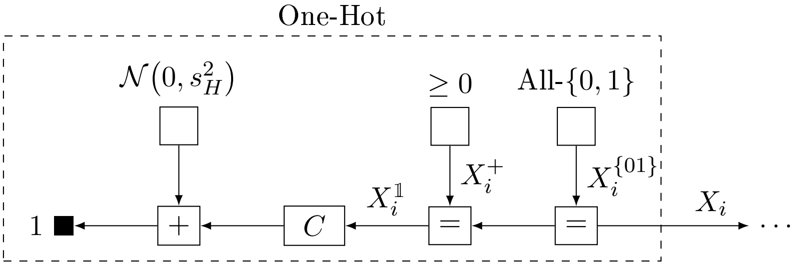


Figure 3.1: Factor Graph of One-Hot prior.

by a row vector $C \triangleq \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$ with its result being fixed to 1, i.e.,

$$CX_i^{\mathbb{1}} = \sum_{m=1}^{M} X_{i,m}^{\mathbb{1}} = 1\,. \tag{3.1}$$

This hard constraint effectively reduces the solution space of X_i to an $M-1$ dimensional hyperplane, which can cause numerical issues in the resulting algorithms (shown by simulations). To ease this effect, a “hypothetical” small noise can be added at this point (zero-mean with small variance $s_H^2 \geq 0$). This is shown to greatly increase the stability of the final algorithm, without decreasing its accuracy notably. To further force all elements of the applied random vector to lie between 0 and 1 (second property) it suffices to apply an element-wise positivity constraint. A NUV prior achieving this has already been described in Subsection 2.2.3. Together with the first property, this indeed restricts all elements to $X_{i,m} \in [0,1]$, $m \in \{1, \dots, M\}$. The third property (emphasise all-$\{0,1\}$ solutions) is arguably the hardest to implement properly. There exist a great variety of possible approaches, a selection of which is discussed in the following Subsections 3.1.1 to 3.1.3. Note that they all differ in performance, especially considering their speed of convergence and their respective tendencies to converge prematurely. Finally, Subsection 3.1.4 states exact expressions for the messages generated by the described One-Hot NUV prior.

3.1.1 Sparsity per Dimension

The idea for this first prior encouraging all-$\{0,1\}$ solutions is to apply a sparsifying NUV prior to each element of X_i separately. This causes the solution to strongly prefer most of its elements to be 0 without penalizing outliers too heavily (i.e., those elements corresponding to selected models). Such a sparsity encouraging prior is the Log-Cost NUV prior as discussed in Subsection 2.2.2. When applied to each element of $X_i^{\{01\}}$ (naming corresponds to Figure 3.1) individually, the generated messages become

$$\overrightarrow{m}_{X_i^{\{01\}}} = \mathbf{0}_M \tag{3.2}$$

$$\overrightarrow{V}_{X_i^{\{01\}}} = \begin{bmatrix} \frac{\hat{m}^2_{X_{i,1}}+\hat{V}_{X_{i,1}}}{\beta} & 0 & \dots & 0 \\ 0 & \frac{\hat{m}^2_{X_{i,2}}+\hat{V}_{X_{i,2}}}{\beta} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\hat{m}^2_{X_{i,M}}+\hat{V}_{X_{i,M}}}{\beta} \end{bmatrix} \tag{3.3}$$

or

$$\overrightarrow{\xi}_{X_i^{\{01\}}} = \mathbf{0}_M \tag{3.4}$$

$$\overrightarrow{W}_{X_i^{\{01\}}} = \begin{bmatrix} \frac{\beta}{\hat{m}^2_{X_{i,1}}+\hat{V}_{X_{i,1}}} & 0 & \dots & 0 \\ 0 & \frac{\beta}{\hat{m}^2_{X_{i,2}}+\hat{V}_{X_{i,2}}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\beta}{\hat{m}^2_{X_{i,M}}+\hat{V}_{X_{i,M}}} \end{bmatrix}. \tag{3.5}$$

Here, $\mathbf{0}_M$ denotes an all-zero vector of length M.

3.1.2 Repulsive Prior at Origin

In this approach, instead of attracting most elements of X_i to 0, all solutions close to the origin are de-emphasized. To do so, either a Laplace or Log-Cost NUV prior can be used, both with negative tuning factor $\beta < 0$. The resulting forward messages through $X_i^{\{01\}}$ therefore become

$$\overrightarrow{m}_{X_i^{\{01\}}} = \mathbf{0}_M \tag{3.6}$$

$$\overrightarrow{V}_{X_i^{\{01\}}} = {\sigma^2}_{\mathrm{RP},i} \cdot \mathbf{I}_M \tag{3.7}$$

or

$$\overrightarrow{\xi}_{X_i^{\{01\}}} = \mathbf{0}_M \tag{3.8}$$

$$\overrightarrow{W}_{X_i^{\{01\}}} = \frac{1}{{\sigma^2}_{\mathrm{RP},i}} \cdot \mathbf{I}_M \,. \tag{3.9}$$

where $\mathbf{I}_M$ denotes the M dimensional identity matrix. For the Laplace prior, ${\sigma^2}_{\mathrm{RP},i}$ is calculated as

$${\sigma^2}_{\mathrm{RP},i} = \frac{||\hat{m}_{X_i}||}{\beta} \,. \tag{3.10}$$

For the Log-Cost prior, it is calculated as

$${\sigma^2}_{\mathrm{RP},i} = \frac{\mathrm{Tr}\left\{\hat{V}_{X_i}\right\} + ||\hat{m}_{X_i}||^2}{\beta} \,. \tag{3.11}$$

Here, $\mathrm{Tr}\{\cdot\}$ denotes the trace and $||\cdot||$ the L_2 norm (further descriptions of these priors can be found in Subsections 2.2.1 and 2.2.2).

Note that the values of ${\sigma^2}_{\mathrm{RP},i}$ are always negative (because $\beta < 0$). Even though they correspond to variances in a statistical model, these negative values do not pose a problem as long as the posterior variances are all non-negative. Therefore, β should be chosen appropriately (i.e., the absolute value of β must be large enough).

3.1.3 Discrete Phase Prior to Target Solutions

A third option to emphasize the desired all-$\{0,1\}$ solutions is to use discrete-phase priors (described in [9]). Thereby, the so-called target vectors t_m, $m \in \{1,\ldots,M\}$ must be chosen equal to the Cartesian bases, i.e.,

$$t_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad t_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad t_M = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \tag{3.12}$$

Accordingly, the forward messages through $X_i^{\{01\}}$ become

$$\overrightarrow{m}_{X_i^{\{01\}}} = M \cdot {\sigma^2}_{\mathrm{DP}} \cdot \sum_{m=1}^{M} \frac{t_m}{\mathrm{Tr}\left\{\hat{V}_{X_i}\right\} + ||\hat{m}_{X_i} - t_m||^2} \tag{3.13}$$

$$\overrightarrow{V}_{X_i^{\{01\}}} = {\sigma^2}_{\mathrm{DP},i} \cdot \mathbf{I}_M \tag{3.14}$$

or

$$\overrightarrow{\xi}_{X_i^{\{01\}}} = M \cdot \sum_{m=1}^{M} \frac{t_m}{\mathrm{Tr}\left\{\hat{V}_{X_i}\right\} + ||\hat{m}_{X_i} - t_m||^2} \tag{3.15}$$

$$\overrightarrow{W}_{X_i^{\{01\}}} = \frac{1}{{\sigma^2}_{\mathrm{DP},i}} \cdot \mathbf{I}_M\,, \tag{3.16}$$

where

$${\sigma^2}_{\mathrm{DP},i} = \frac{1}{M} \cdot \left(\sum_{m=1}^{M} \frac{1}{\mathrm{Tr}\left\{\hat{V}_{X_i}\right\} + ||\hat{m}_{X_i} - t_m||^2} \right)^{-1}. \tag{3.17}$$

Note that this approach does not come with a natural way to tune the NUV updates (similar to β in the previous two Subsections). However, by reducing / increasing all other tuning parameters in the model, this NUV prior can implicitly be tuned too.

3.1.4 Resulting Messages Generated by One-Hot NUV Prior

In the previous Subsections, different possible forward messages through $X_i^{\{01\}}$ have been described. Choosing any of these priors, the resulting forward messages through

X_i (i.e., the messages generated by the One-Hot NUV prior) can easily be expressed by its dual representation as

$$\overrightarrow{\xi}_{X_i} = \frac{1}{s_H^2} C^\mathsf{T} + \overrightarrow{\xi}_{X_i^+} + \overrightarrow{\xi}_{X_i^{\{01\}}} \tag{3.18}$$

$$\overrightarrow{W}_{X_i} = \frac{1}{s_H^2} C^\mathsf{T} C + \overrightarrow{W}_{X_i^+} + \overrightarrow{W}_{X_i^{\{01\}}} \,. \tag{3.19}$$

Expressions for the messages $\overrightarrow{\xi}_{X_i^+}$ and $\overrightarrow{W}_{X_i^+}$ are given in (2.17) and (2.18), respectively. Note that this representation of the messages generated by the One-Hot prior require the "hypothetical" noise variance s_H^2 to be strictly greater than zero!

Alternatively, the same messages can be expressed by their conventional representations as

$$\overrightarrow{m}_{X_i} = \overrightarrow{m}_{X_i'} + \overrightarrow{V}_{X_i'} C^\mathsf{T} G_i \Big(1 - C \overrightarrow{m}_{X_i'}\Big) \tag{3.20}$$

$$\overrightarrow{V}_{X_i} = \overrightarrow{V}_{X_i'} - \overrightarrow{V}_{X_i'} C^\mathsf{T} G_i C \overrightarrow{V}_{X_i'} \,, \tag{3.21}$$

where

$$\overrightarrow{V}_{X_i'} \triangleq \Big(\overrightarrow{W}_{X_i^+} + \overrightarrow{W}_{X_i^{\{01\}}}\Big)^{-1} \tag{3.22}$$

$$\overrightarrow{m}_{X_i'} \triangleq \overrightarrow{V}_{X_i'} \Big(\overrightarrow{\xi}_{X_i^+} + \overrightarrow{\xi}_{X_i^{\{01\}}}\Big) \tag{3.23}$$

$$G_i \triangleq \left(s_H^2 + \sum_{m=1}^{M} \overrightarrow{V}_{X_{i,m}'}\right)^{-1} \,. \tag{3.24}$$

Note how in this representation s_H^2 is explicitly allowed to be equal to zero. However, simulations still showed the algorithm to be much more stable if some small noise is assumed. It is further pointed out that the matrix inversion in Equation (3.22) only involves diagonal matrices, meaning that its computation is not very expensive (reduces to M scalar inversions per matrix). A summary of this novel One-Hot NUV prior can be found in the Appendix, Table A.1.

3.2 Model Selector Mechanism

A factor graph of the developed model selector mechanism is shown in Figure 3.2. It visualizes the interplay between the observations Y_i, the given set of models with outputs $X_{i,m}$, and their inferred weighting factors $S_{i,m}$. The optional second subscript in these quantities specifies the index of their corresponding model. Further note that the weight vectors S_i are modelled PWC, causing the method to emphasize solutions with sparse model changes.

In the following, this general idea is further worked out. Subsection 3.2.1 describes a trick to tackle some immediate issues with the factor graph in Figure 3.2. Next, the functionality of the depicted multiplication nodes is explained in Subsection 3.2.2. Subsection 3.2.3 then derives the actual Gaussian message passing algorithm to estimate the values of S_i in the depicted system. Finally, Subsection 3.2.4 applies the developed method to a simple example, showcasing its enormous potential.

3.2.1 Iterative Estimation of S_i and $X_{i,m}$

The factor graph in Figure 3.2 poses two fundamental problems. Firstly, the interplay between the M models and the model selector mechanism creates loops in the resulting factor graph. This is an issue because message passing algorithms are only guaranteed to converge in loop-free factor graphs, which is therefore no longer given. Secondly, the depicted multiplication nodes restrict the product of two random quantities to a Gaussian random vector (named $R_{i,m}$), meaning that either of the two incoming random vector can not be Gaussian.

Luckily, there exists a simple trick solving both of these issues at the same time. Note that its fundamental idea is adapted from a method used in [1]. Generally, the proposed solution iteratively solves small sub-problems of the whole factor graph, while assuming that the remaining parts of it are fixed. In particular, the following two improvement steps are iteratively performed:

- Improve estimates of weight vectors S_i for fixed model outputs $X_{i,m} = \hat{m}_{X_{i,m}}$.
- Improve estimates of model outputs $X_{i,m}$ for fixed weight vectors $S_i = \hat{m}_{S_i}$

This resolves the issue of loops as the factor graph is effectively split along the fixed quantities. Furthermore, because either one of the ingoing messages into the multiplication node is fixed, the other quantity must be normally distributed (effect explained in Subsection 2.1.2).

3.2.2 Multiplication Nodes in Figure 3.2

The multiplication nodes in Figure 3.2 link the models with their respective weighting factors. Depending on whether the model selector vector element $S_{i,m}$ or the model output $X_{i,m}$ is fixed (iterative estimation explained in previous Subsection 3.2.1), the effect of this multiplication node is different. A close-up of both cases is shown in Figure 3.3, with the former in the left subplot and the latter in the right subplot. In the following, these two perspectives are further explored.

Effect for Model Output Estimation

Subsection 3.2.1 states that when the estimation of the model outputs $X_{i,m}$ is improved, the model selector vectors are fixed to their current mean estimates, i.e., $S_i = \hat{m}_{S_i}$. This

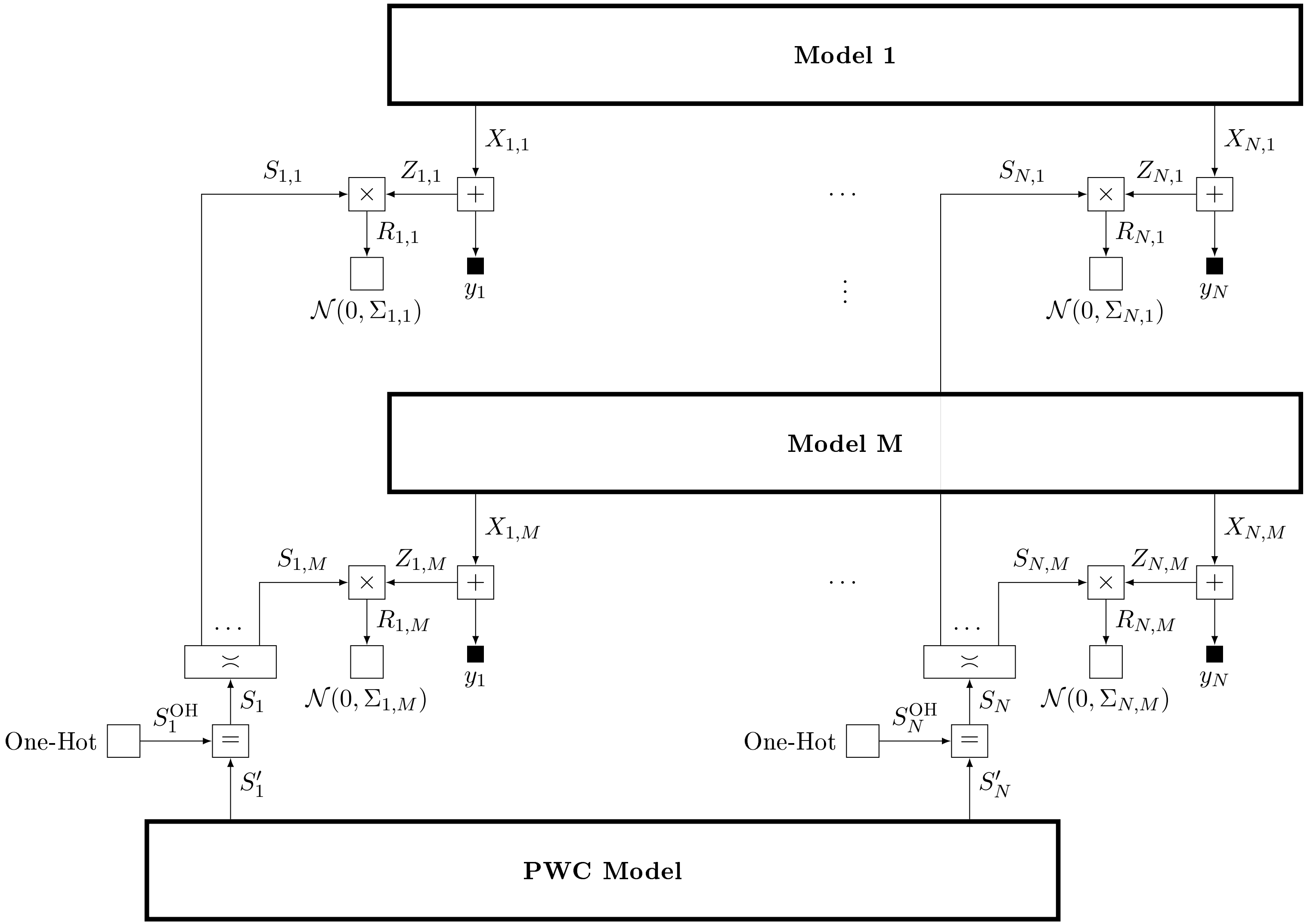


Figure 3.2: Factor graph of model selector mechanism.

scenario is depicted in Subfigure 3.3a. Closing the dashed box labelled by $g(\cdot)$ results in

$$g(x_{i,m}) = \mathcal{N}\big((x_{i,m} - y_i)\hat{m}_{S_{i,m}}; 0, \Sigma_{i,m}\big) \tag{3.25}$$

$$\propto \exp\left(-\frac{1}{2}(x_{i,m} - y_i)^{\mathsf{T}}\left(\hat{m}_{S_{i,m}}^{-2}\Sigma_{i,m}\right)^{-1}(x_{i,m} - y_i)\right). \tag{3.26}$$

According to the distribution in Equation (3.26), the backward messages through $X_{i,m}$ therefore are

$$\overleftarrow{m}_{X_{i,m}} = y_i \tag{3.27}$$

$$\overleftarrow{V}_{X_{i,m}} = \hat{m}_{S_{i,m}}^{-2}\Sigma_{i,m} \tag{3.28}$$

or

$$\overleftarrow{W}_{X_{i,m}} = \hat{m}_{S_{i,m}}^{2}\Sigma_{i,m}^{-1} \tag{3.29}$$

$$\overleftarrow{\xi}_{X_{i,m}} = \overleftarrow{W}_{X_{i,m}} y_i\,, \tag{3.30}$$

where $\Sigma_{i,m}$ is the covariance matrix of the assumed observation noise of model m at time index i. Therefore, the current estimates of the model selector vector element $S_{i,m} = \hat{m}_{S_{i,m}}$ inversely scales the assumed observation noise covariance matrix quadratically. Because the values of $S_{i,m}$ are bounded between 0 and 1, a value close to 1 means that the model sees approximately the assumed observation noise, while a value close to 0 scales the assumed observation noise power towards infinity! Effectively, the latter scaling causes the model to discard the observation at the corresponding time index to estimate any of its outputs.

Effect for Model Selector Vector Estimation

For the improvement of the model selector vector S_i, the model outputs are fixed to its current estimates, i.e., $X_{i,m} = \hat{m}_{X_{i,m}}$. The resulting factor graph representation of a multiplication node is shown in Subfigure 3.3b. There, closing the dashed box $f(\cdot)$ results in

$$f(s_{i,m}) = \mathcal{N}\big(s_{i,m}\hat{m}_{Z_{i,m}}; 0, \Sigma_{i,m}\big) \tag{3.31}$$

$$\propto \exp\left(-\frac{s_{i,m}^2}{2}\hat{m}_{Z_{i,m}}^{\mathsf{T}}\Sigma_{i,m}^{-1}\hat{m}_{Z_{i,m}}\right), \tag{3.32}$$

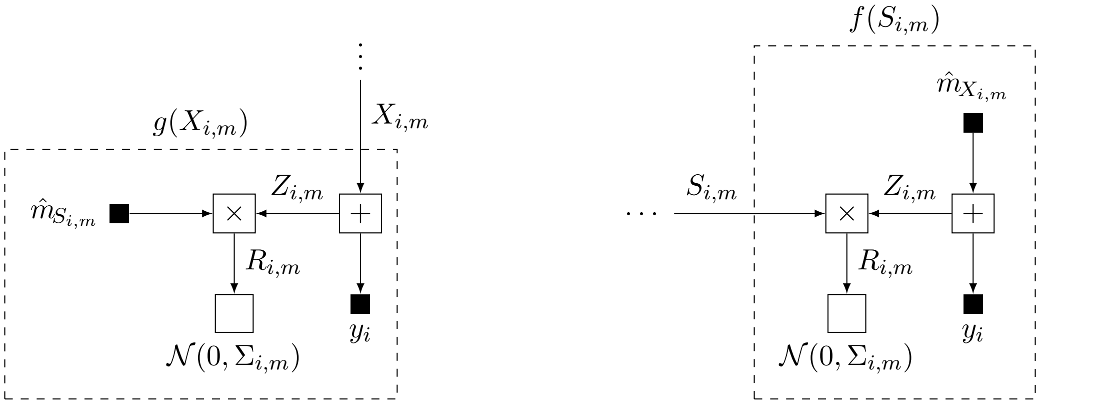
where $\hat{m}_{Z_{i,m}} = y_i - \hat{m}_{X_{i,m}}$. From the expression in (3.32) it becomes clear that $S_{i,m}$ really is normally distributed around zero with a variance of

$$\sigma_{i,m}^2 = \left(\hat{m}_{Z_{i,m}}^{\mathsf{T}}\Sigma_{i,m}^{-1}\hat{m}_{Z_{i,m}}\right)^{-1}. \tag{3.33}$$

Therefore, the Gaussian backward messages through $S_{i,m}$ are

$$\overleftarrow{m}_{S_{i,m}} = 0 \tag{3.34}$$

$$\overleftarrow{V}_{S_{i,m}} = \left(\hat{m}_{Z_{i,m}}^{\mathsf{T}}\Sigma_{i,m}^{-1}\hat{m}_{Z_{i,m}}\right)^{-1} \tag{3.35}$$



(a) Information gained about $X_{i,m}$ for fixed $S_{i,m} = \hat{m}_{S_{i,m}}$.

(b) Information gained about $S_{i,m}$ for fixed $X_{i,m} = \hat{m}_{X_{i,m}}$.

Figure 3.3: Close-up of multiplication node in Figure 3.2.

or

$$\overleftarrow{\xi}_{S_{i,m}} = 0 \tag{3.36}$$

$$\overleftarrow{W}_{S_{i,m}} = \hat{m}_{Z_{i,m}}^{\mathsf{T}} \Sigma_{i,m}^{-1} \hat{m}_{Z_{i,m}} \,. \tag{3.37}$$

In words, this means that closing the box depicted in Figure 3.3b always causes the estimator to believe that $S_{i,m}$ is zero, independent of the actual observations or the considered model. However, the better $\hat{m}_{Z_{i,m}}$ can be explained by the assumed observation noise $\Sigma_{i,m}$, the "less sure" one is that $S_{i,m}$ should actually be 0. Finally, taking into account the One-Hot prior applied to S_i and the influence of the PWC model, one of the considered models will indeed be selected in each time index!

3.2.3 Estimation of S_i

In a first step, the messages passed to the PWC model (i.e., the backward messages through S_i') need to be calculated. Based on the factor graph in Figure 3.2, they can be expressed as

$$\overleftarrow{\xi}_{S_i'} = \overrightarrow{\xi}_{S_i^{\mathrm{OH}}} + \overleftarrow{\xi}_{S_i} \tag{3.38}$$

$$\overleftarrow{W}_{S_i'} = \overrightarrow{W}_{S_i^{\mathrm{OH}}} + \overleftarrow{W}_{S_i} \,. \tag{3.39}$$

Thereby, the messages $\overleftarrow{\xi}_{S_i}$ and $\overleftarrow{W}_{S_i}$ can be constructed as

$$\overleftarrow{\xi}_{S_i} = \mathbf{0}_M \tag{3.40}$$

$$\overleftarrow{W}_{S_i} = \begin{bmatrix} \overleftarrow{W}_{S_{i,1}} & 0 & \dots & 0 \\ 0 & \overleftarrow{W}_{S_{i,2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \overleftarrow{W}_{S_{i,M}} \end{bmatrix}, \tag{3.41}$$

where the expressions for the scalars $\overleftarrow{W}_{S_{i,m}}$ are given in Equation (3.37). Expressions for $\overrightarrow{\xi}_{S_i^{\mathrm{OH}}}$ and $\overrightarrow{W}_{S_i^{\mathrm{OH}}}$ are given in (3.18) and (3.19). However, note that these expressions are only valid if s_H^2 is chosen to be greater than zero. If $s_H^2 = 0$, $\overrightarrow{\xi}_{S_i^{\mathrm{OH}}}$ and $\overrightarrow{W}_{S_i^{\mathrm{OH}}}$ must be constructed from the mean and covariance messages given in (3.20) and (3.21). Note that simulations performed with s_H^2 strictly greater than zero tended to be much more stable, independent of the chosen message representation.

Given the messages calculated in (3.38) and (3.39), one iteration of IRWLS can be performed in the PWC model. This process has extensively been described in Subsection 2.3.3. For the updated estimates of S_i (i.e., $\hat{m}_{S_i}$ and $\hat{V}_{S_i}$), the messages generated by the One-Hot NUV priors change. Accordingly, the messages passed to the PWC model (Equations (3.38) and (3.39)) change too. This whole process constitutes one iteration of IRWLS when estimating S_i.

3.2.4 Trivial Example: Fitting Known Levels

To show-case the power of this described model selector mechanism, the following example is considered. Given are $N = 100$ observations of dimension $D = 1$, originating from $M = 4$ non-equidistant constant models (also referred to as levels). These levels are, at least for this example, perfectly known to the estimator. The observation noise has variance $\sigma^2 = 1$ (compared to a maximum signal magnitude of 4). The goal is therefore to estimate which sections of the given example have been generated by which known level. This example is in some sense "trivial", as the described method will later be used to perform the same regression task on unknown levels too (Subsection 3.3.1).

A summary of the estimation results is shown in Figure 3.4. Thereby, the upper Subplot shows the received data in light-blue, the true transmitted levels in orange, and the estimated levels in green. It is important to understand that the green line is constructed from the known levels, where in each time index the level is chosen for which the corresponding element of S_i is maximal. The lower Subplot shows all elements of S_i, each in a separate colour. This plot visualizes how S_i indeed converges to an all-$\{0, 1\}$ solution.

Interpretation of Results for Trivial Example

Looking at the upper Subplot of Figure 3.4, one can conclude that the proposed method is indeed able to determine which sections of the observations have been generated by which known (i.e., observed) model. In fact, the assignment of the levels is almost perfect, except for a few indices towards the end. Furthermore, it is noted that these excellent

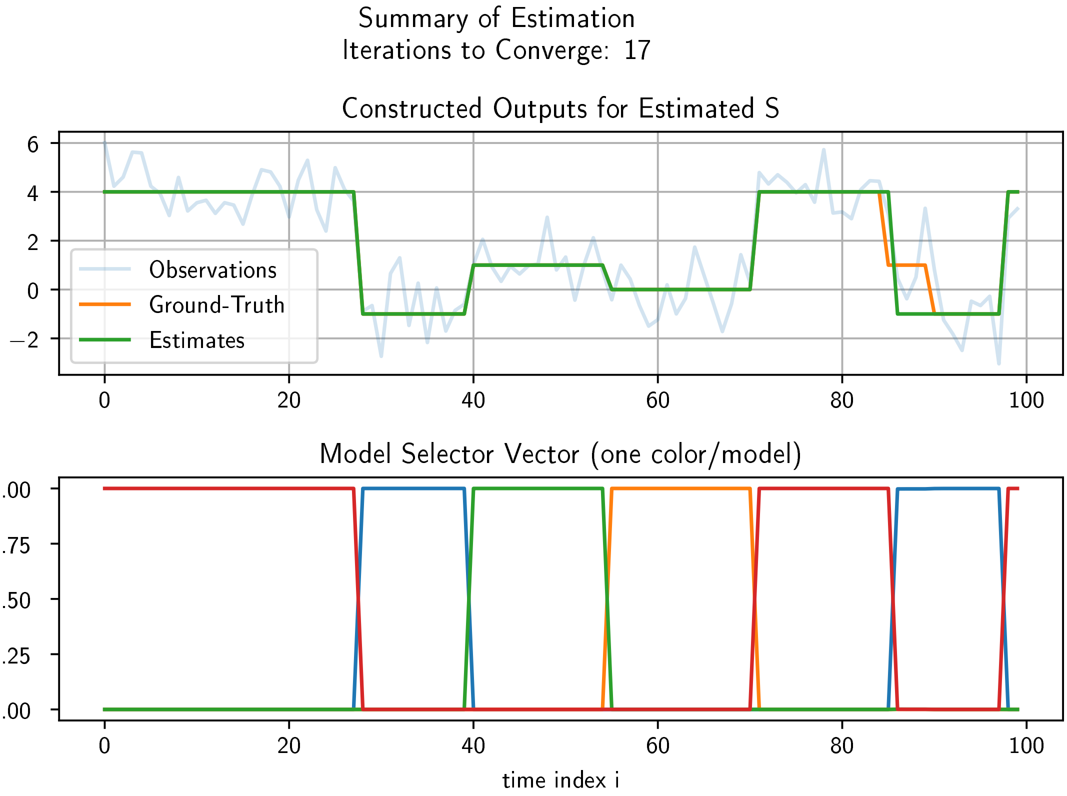


Figure 3.4: Estimation summary of model selector mechanism for the "trivial" example where the levels are known to the estimator. Note that the approach described in Subsection 3.1.2 with the Log-Cost prior is used for the One-Hot NUV.

results are achieved within only 17 iterations of IRWLS, which should be manageable for most practical applications. This is particularly impressive considering that the sparsity encouraging NUV prior has not been specially tuned. In fact, the special case of a Plain NUV (i.e., $\beta = M$, recall that M is the dimension of S_i) is used, which seems to be a good choice for this type of problems. Further note that the approach described in Subsection 3.1.2 with the Log-Cost prior is used for the One-Hot NUV. During simulations, this method yielded the best results. Because of this observation, it has been decided that this approach is used for all simulations presented in this report.

Evolution of Estimated S_i

For the same simulation, Figure 3.5 shows how S_i evolved during its estimation. In particular, the three stages of 1 iteration (dashed), 4 iterations (dotted), and 17 iterations (solid) are depicted. For the last case, the method is considered to have converged, i.e., the found solution is close enough to an all-$\{0, 1\}$ solution. Note how the sum over all elements always stays close to 1 as it is enforced by the One-Hot prior (for reference check Figure 3.1). The smooth shape of the curves in the intermediate estimation stages is caused by the sparsifying prior of the PWC model (described in Section 2.3).

3.3 Applications of Model Selector

This final Section of Chapter 3 discusses some advanced applications of the proposed model selector mechanism. The first investigated model fits a finite number of constant levels to noisy outputs (Subsection 3.3.1). This setting is similar to the one investigated in Subsection 3.2.4, but this time the levels are not assumed to be known. The second investigated model can be seen as an extension to the classical PWC model, but with a known base-level that is regularly revisited (Subsection 3.3.2). Note that for both models the iterative estimation method described in Subsection 3.2.1 is applied. Further note that the factor graph shown in Figure 3.2 still applies. The final Subsection 3.3.3 touches further important aspects of the proposed model selector mechanism and their respective applications.

3.3.1 Fitting Finite Number of Constant Levels

The investigated setting for the Constant-Level-Fitting (CLF) model is similar to the one discussed in Subsection 3.2.4, but with unknown levels. These levels are constant with some optional prior $\rho(x_{0,m})$, which can incorporate prior knowledge and / or initial guesses. For fixed (i.e., assumed to be observed) weight factors S_i, they are therefore represented by the factor graph shown in Figure 3.6.

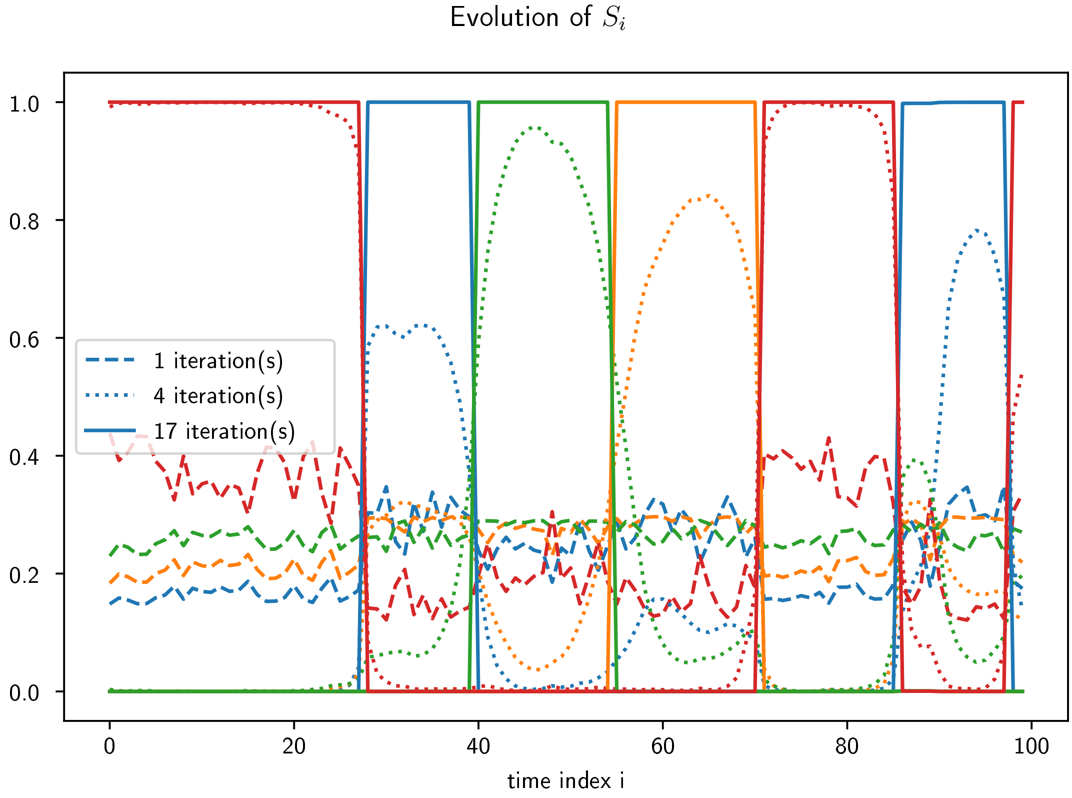


Figure 3.5: Plot of estimated weight factors S_i over time for the “trivial” example.

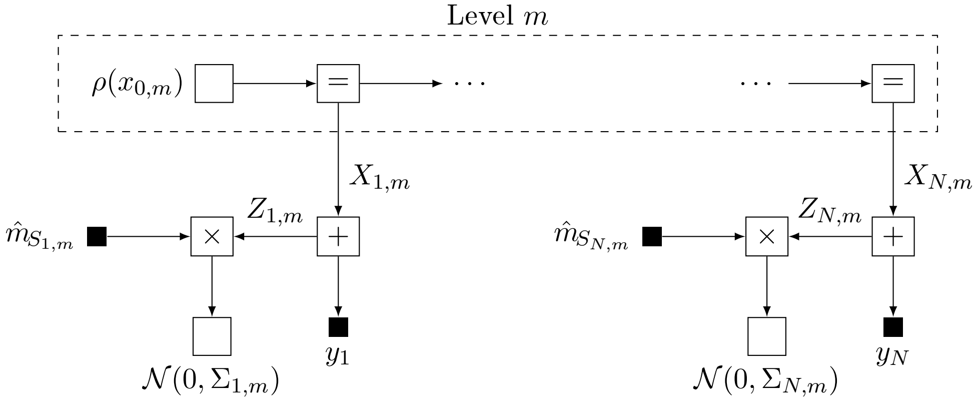


Figure 3.6: Factor graph of constant model (i.e., level) m with some prior $\rho(x_{0,m})$. Note that the current estimates of S_i are fixed and can therefore be treated as observations.

Estimation of Constant Levels

From the factor graph in Figure 3.6, the backward messages through $X_{i,m}$ have been derived in Subsection 3.2.2 as

$$\overleftarrow{W}_{X_{i,m}} = \hat{m}_{S_{i,m}}^2 \Sigma_{i,m}^{-1} \tag{3.42}$$

$$\overleftarrow{\xi}_{X_{i,m}} = \overleftarrow{W}_{X_{i,m}} y_i \, . \tag{3.43}$$

Accordingly, the MAP estimate of the m-th level X_m is

$$\hat{V}_{X_m} = \left(\overrightarrow{W}_{X_{0,m}} + \sum_{i=1}^{N} \overleftarrow{W}_{X_{i,m}} \right)^{-1} \tag{3.44}$$

$$\hat{m}_{X_m} = \hat{V}_{X_m} \left(\overrightarrow{\xi}_{X_{0,m}} + \sum_{i=1}^{N} \overleftarrow{\xi}_{X_{i,m}} \right) , \tag{3.45}$$

where $\overrightarrow{W}_{X_{0,m}}$ and $\overrightarrow{\xi}_{x_{0,m}}$ are specified by the prior $\rho(x_{0,m})$. The mean calculated in Equation (3.45) is used as the estimate of the m-th level. The covariance matrix in (3.44) can be interpreted as an uncertainty measure of this estimation. The latter however is not used for the described algorithm and must therefore not necessarily be saved.

Because the estimates of S_i have not necessarily yet converged to an all-$\{0, 1\}$ solution, the estimate calculated in Equations (3.45) tends to be biased towards the mean over all observations. This in turn will affect the estimation of S_i, causing the convergence speed to decrease drastically. Therefore, alternatively to the former described method, the levels could also be estimated by only considering the samples for which the m-th element of the weight factor is actually the highest one. In other words, the observations can be grouped into M groups, where the m-th group contains all observations with indices in

$$\mathcal{I}_m \triangleq \left\{ i \in \{1, \ldots, N\} \mid \underset{m'}{\operatorname{argmax}} \, S_{i,m'} = m \right\} , \quad m \in \{1, \ldots, M\} \, . \tag{3.46}$$

According to this grouping, the estimated levels are calculated as

$$\hat{V}_{X_m} = \left(\overrightarrow{W}_{X_{0,m}} + \sum_{i \in \mathcal{I}_m} \Sigma_{i,m}^{-1} \right)^{-1} \tag{3.47}$$

$$\hat{m}_{X_m} = \hat{V}_{X_m} \left(\overrightarrow{\xi}_{X_{0,m}} + \sum_{i \in \mathcal{I}_m} \Sigma_{i,m}^{-1} y_i \right) . \tag{3.48}$$

Note that these expressions can be directly derived from (3.44) and (3.45), where the estimated weight factors are set to 1 (i.e., S_i is assumed to have converged to an all-$\{0, 1\}$ solution). In the following discussion, the former approach is referred to as "direct", the later as "selective" level estimation.

Results for CLF Model

This paragraph showcases that the described CLF model is indeed able to fit a finite number of constant levels to some data. In the presented simulations, the estimator is again given the same observations that have already been used in Subsection 3.2.4 (in short: $N = 100$, $D = 1$, $M = 4$, $\sigma^2 = 1$). To estimate the levels, the "direct" approach is used (similar results are achieved with the "selective" approach, but it tends to be less stable). Note that only one iteration of IRWLS is performed per improvement step of S_i.

Figure 3.7 shows a summary of the achieved results. The upper Subplot depicts the given observations in light-blue, the true transmitted levels in orange, and the final estimation in green. Note that this green line is constructed based on the estimations of the weight factors S_i and the estimated levels. The lower Subplot shows the corresponding estimation of all elements of S_i, each in a separate colour. Note that it indeed converged to an all-$\{0, 1\}$ solution.

Looking at these results, it can be concluded that the described CLF model is able to find a good estimate of the underlying ground-truth, comparable to a manual guess. Furthermore, it is pointed out that these nice results are achieved within only 22 iterations, making it applicable to many practical systems. Note that in these considerations the exact observation noise variance $\sigma^2 = 1$ and the correct number of levels $M = 4$ is known to the estimator. However, comparably good results can be achieved for a close estimation of $\sigma^2_{\text{guess}} \approx \sigma^2$ and a large enough guess of $M_{\text{guess}} \geq M$ (further discussed in Subsection 3.3.3).

Evolution of Level Estimates

For the same simulation, Figure 3.8 visualizes how the level estimates evolved during estimation. In the first iteration, their estimates are based on a very poor guess of S_i (check discussion of Figure 3.5). In particular, all elements of S_i are still very close to their initial values $1/M$, causing all level estimates to be close to the average over all observations. They then start to diverge rapidly for ≈ 10 iterations, at which point they already closely approximate the final estimates. During the remaining ≈ 10 iterations, these final estimations are refined, before the algorithm terminates in iteration 22.

3.3.2 PWC Model With Base Level

This second application shifts the focus back to the problem setting of the PWC model (Section 2.3), but this time including some known base layer. This could for example be some default behaviour of a system that is observed most of the time. However, there occur significantly different modes too which should be detected and modelled appropriately. Therefore, a Return-To-Base (RTB) model has been developed. It consists of two models, one of which is perfectly known (base model) and the other is assumed to be PWC. These two models are connected by a model selector estimating which sections of observations have most likely been generated by which model. In the following, the

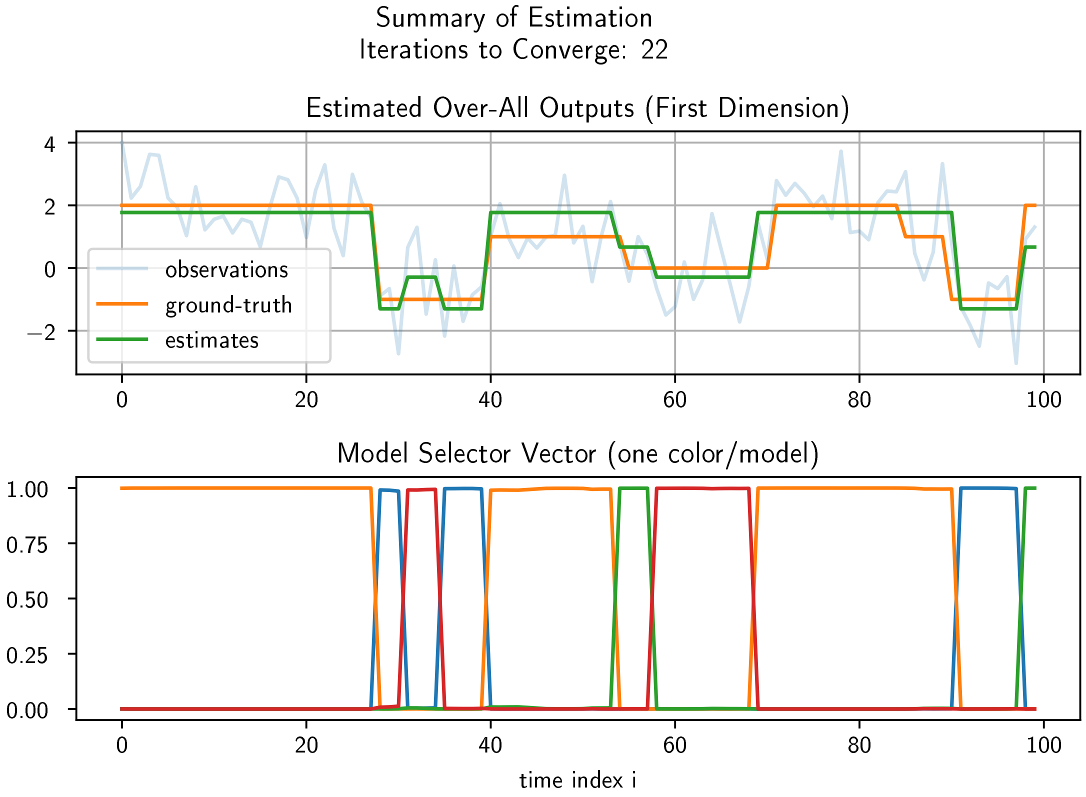


Figure 3.7: Estimation summary of CLF model. Note that the "direct" level estimation approach is used.

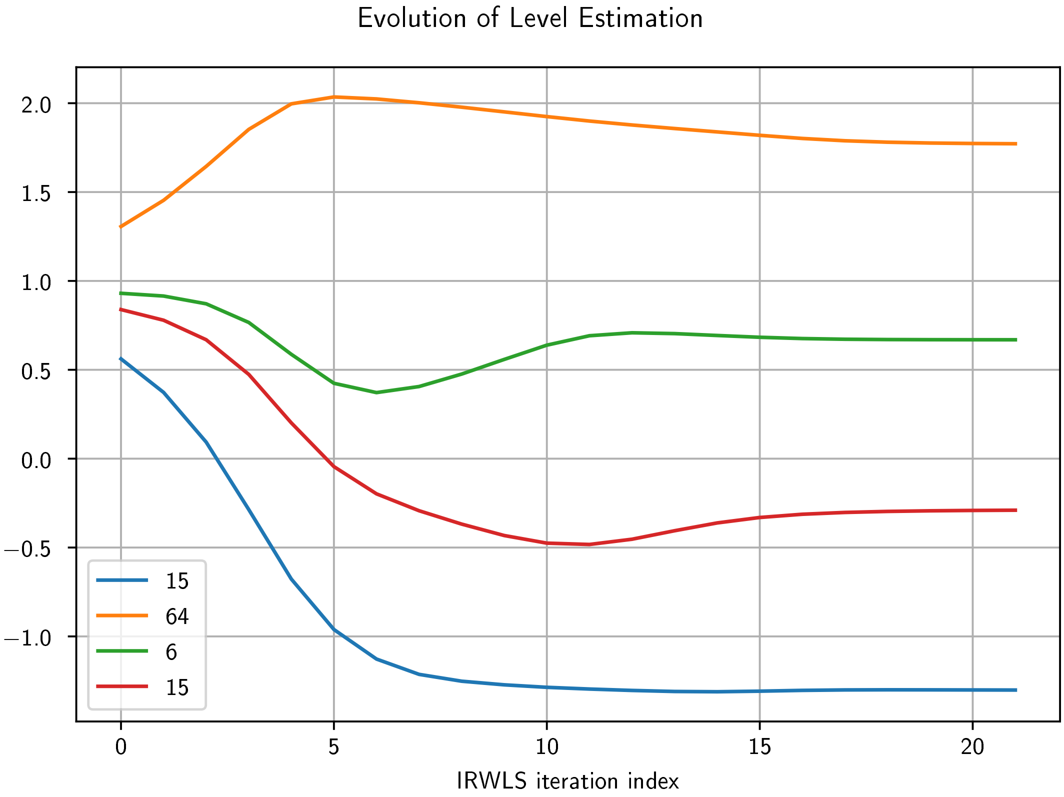


Figure 3.8: Evolution of level estimates in CLF model. The legend denotes how many times each of these levels is used to construct the final estimate (i.e., the green line in the upper Subplot of Figure 3.7).

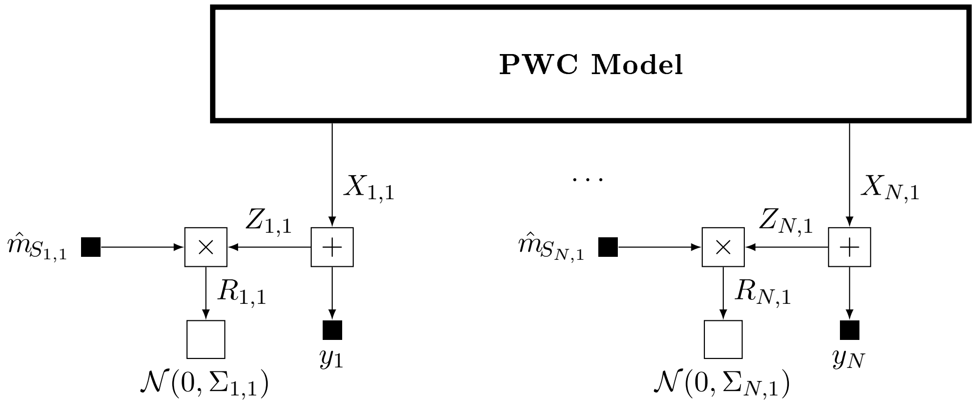


Figure 3.9: Factor graph to improve estimation of $X_{i,1}$ (i.e., outputs of PWC model) for fixed S_i.

outputs of the base layer are denoted by $X_{i,0}$ and those of the PWC model by $X_{i,1}$ ($i \in \{1, \ldots, N\}$ is the time index). Note that this idea can easily be generalized to any evolution model with arbitrary known base-level.

Similar to the CLF model, this RTB model also iteratively improves its estimates of the weight vectors S_i and the model outputs $X_{i,1}$ ($X_{i,0}$ are already assumed to be perfectly known). For the improvement step of $X_{i,1}$, the factor graph shown in Figure 3.9 applies.

Estimation of PWC Outputs in RTB Model

Again, the backward messages into the model can be adapted from the derivation in Subsection 3.2.1 as

$$\overleftarrow{W}_{X_{i,1}} = \hat{m}_{S_{i,1}}^2 \Sigma_{i,1}^{-1} \tag{3.49}$$

$$\overleftarrow{\xi}_{X_{i,1}} = \overleftarrow{W}_{X_{i,1}} y_i \,. \tag{3.50}$$

Given these messages, IRWLS can be performed in the PWC model (Subsection 2.3.3). Note how the precision matrix messages are scaled by the weight factors squared. Therefore, if the model selector mechanism estimates the observation at some index i to have been generated by the base level, the corresponding observation y_i is effectively discarded for the estimation of the PWC outputs.

Preference to Base Level

The discussed RTB model has previously been described as consisting of a known base level and a PWC model, both connected via a model selector. Unfortunately, the straight forward implementation of such a method tends to almost always select the PWC model, as it can approximate the base level arbitrarily close too. Even though the resulting fit of the underlying data still tends to be very good (as one would expect from any PWC model), the ability of distinguishing between the base level and some "special" behaviour is lost. Fortunately, there is a simple solution to tackle this problem.

Subsection 3.2.2 describes how the multiplication nodes in Figure 3.2 connect the model selector mechanism with the models. In particular, it is argued that they determine how well the difference between the (assumed) model output $X_{i,m}$ and the actual observation Y_i can be explained by the assumed observation noise covariance matrix $\Sigma_{i,m}$. From that point of view, scaling the assumed observation noise covariance matrix by a factor $\alpha > 1$ means that the corresponding model is more likely to explain data slightly deviating from the expected values. Therefore, this model is effectively preferred, where α can be used to tune how much it should be preferred. In the following simulations, the (scalar and known) variances of the base level are scaled by $\alpha = 2$.

Results for RTB Model

The setup for the following simulation is quite similar to the previous ones. The estimator is given $N = 100$ observations of dimension $D = 1$. These observations are generated by a PWC ground truth, where the known all-zero base layer is generating the first and last sections. The levels in between are completely unknown. The observation noise is known to be $\sigma^2 = 1$, which is scaled by $\alpha = 2$ for the base layer. A plot of the generated observations is shown in the upper Subplot of Figure 3.10. Thereby, the samples originating from the base layer are depicted in blue, those from the PWC model in red, and the underlying ground truth is shown in orange.

The lower Subplot in Figure 3.10 shows the actual estimated outputs in green. Note that this line is constructed from both, the estimated outputs of the PWC model and the known base layer. In particular, the parts shaded blue are estimated by the model selector mechanism to have been generated by the base layer. Note how this segmentation almost perfectly coincides with the actual origins of the data (indicated by the colour of the observations). Therefore, it can be concluded that the proposed method is indeed able to estimate PWC data with some known base level reasonably well, while also providing a method to distinguish which sections of this data have been generated by the base layer and which not.

3.3.3 Further Important Aspects and Their Applications

To conclude the discussion of the model selector mechanism, this Subsection describes further important aspects of the found method and their respective applications. This discussion is by no means complete, but should be viewed as a basis for further investigations of the method.

Distinguishing Between Different Correlation Patterns

Subsection 3.2.2 discusses the effect of the normal nodes in Figure 3.2. Thereby, for the improvement of the model selector vector estimates, the following backward messages

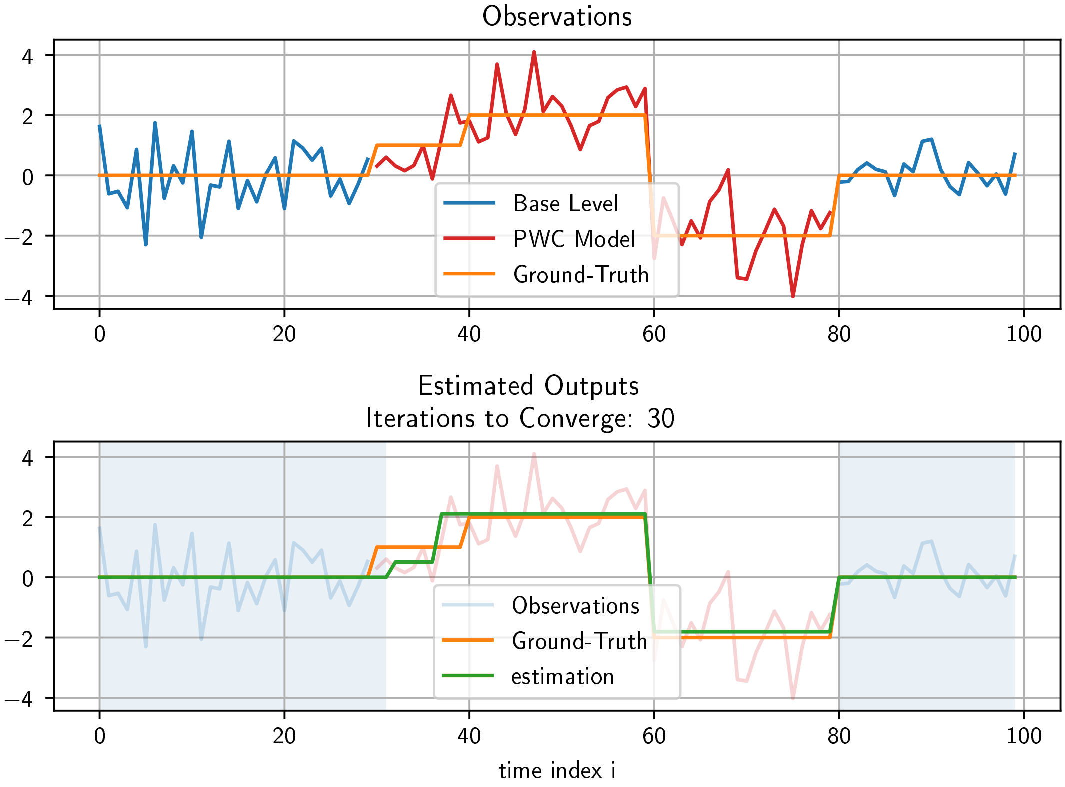


Figure 3.10: Figure describing the estimations of the RTB model. The upper Subplot shows the given observations, the lower Subplot the resulting estimates. Note that the blue shaded areas indicate which sections are estimated to have been generated by the base level.

through $S_{i,m}$ are derived

$$\overleftarrow{\xi}_{S_{i,m}} = 0 \tag{3.51}$$

$$\overleftarrow{W}_{S_{i,m}} = \hat{m}_{Z_{i,m}}^{\intercal} \Sigma_{i,m}^{-1} \hat{m}_{Z_{i,m}} \,, \tag{3.52}$$

where $\hat{m}_{Z_{i,m}} = y_i - \hat{m}_{X_{i,m}}$, and $\Sigma_{i,m}$ is the assumed noise covariance matrix. Therefore, the information passed to the model selector mechanism amounts to how well $\hat{m}_{Z_{i,m}}$ can be described by $\Sigma_{i,m}$.

This simple observation means that the proposed algorithm should be able to distinguish between models with different assumed observation noises, as long as the observation noise covariance matrices differ by more than just a scale factor. To put it differently, the model selector mechanism is able to differentiate between observations originating from model m and m', $m \neq m'$, if their respective assumed observation noise covariance matrices at time index i fulfil

$$\Sigma_{i,m} \neq \gamma \Sigma_{i,m'}, \quad \forall \gamma \in \mathbb{R}_{>0} \,. \tag{3.53}$$

(This is obviously only possible for the multivariate case, i.e., $D \geq 1$). Note that this also works if the two models generate the same mean, i.e., if they only differ in the assumed observation noise!

This effect is demonstrated by the following simulation. Thereby, the estimator is given $N = 100$ observations of dimension $D = 2$. This data is generated by two zero-mean Gaussian noise sources, where the two dimensions either slightly correlate positively or negatively. The given observations are visualized in Figure 3.11. There, the upper Subplot shows the first dimension of the data plotted in blue (model 1) and red (model 2). The lower Subplot shows a scatter plot of the same data, indicating the slight positive (blue, model 1) and negative (red, model 2) correlation between the dimensions. Figure 3.12 shows the corresponding estimates of the model selector mechanism. Thereby, the upper Subplot again depicts the first dimension of the given observations, where now the sections estimated to be generated by model 1 are shaded blue and those by model 2 red, respectively. The lower Subplot shows the corresponding final estimate of S_i, perfectly overlapping with the shading in the upper Subplot. By that, it can be concluded that the model selection is again fairly accurate, as all model changes are detected with an error of only a few samples in each change.

Discarding Unused Models

For the presented results in Subsection 3.3.1, the exact number of models M is given to the estimator. However, in real-world scenarios, this number is often not exactly known. To go even further, estimating this number could be one of the main goals in some applications. Does this lack of knowledge about M render the proposed method impractical for most applications? The answer to this question is no. The proposed algorithm is perfectly capable of discarding unused models, effectively finding the optimal amount of models M_{opt} needed to fit the given data. However, note that the initial guess M_{init} must be large enough, i.e., $M_{\text{init}} \geq M_{\text{opt}}$.

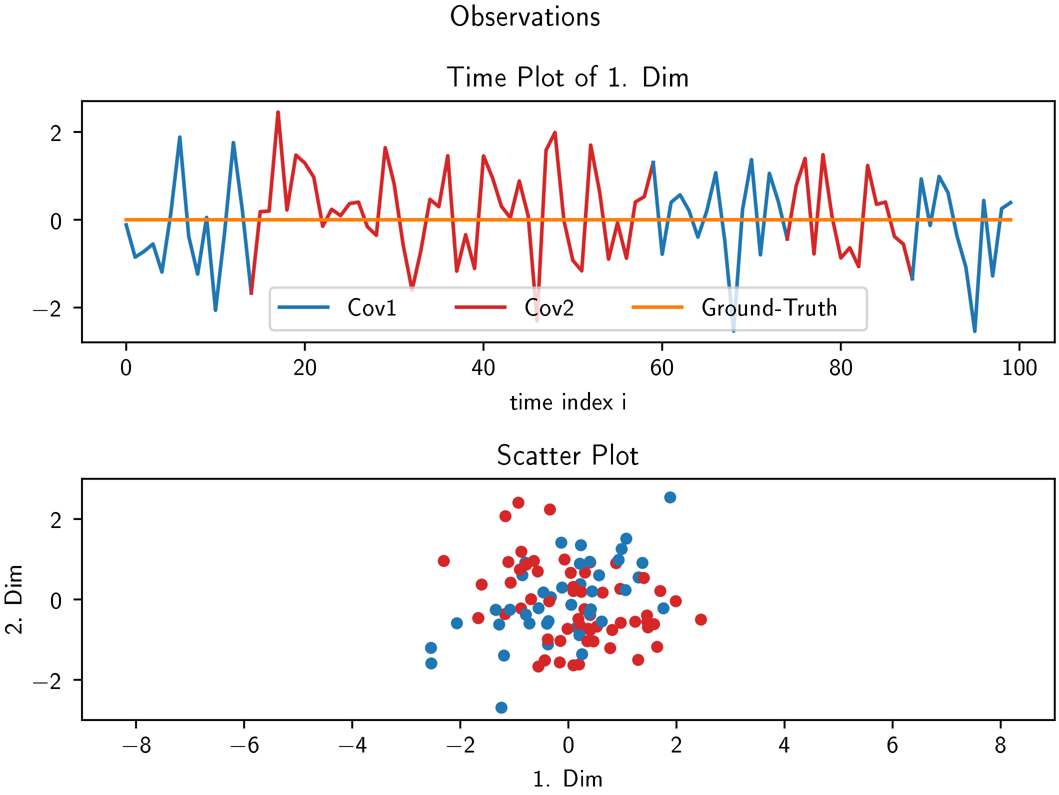


Figure 3.11: Generated observations for model selector distinguishing between different correlation patterns.

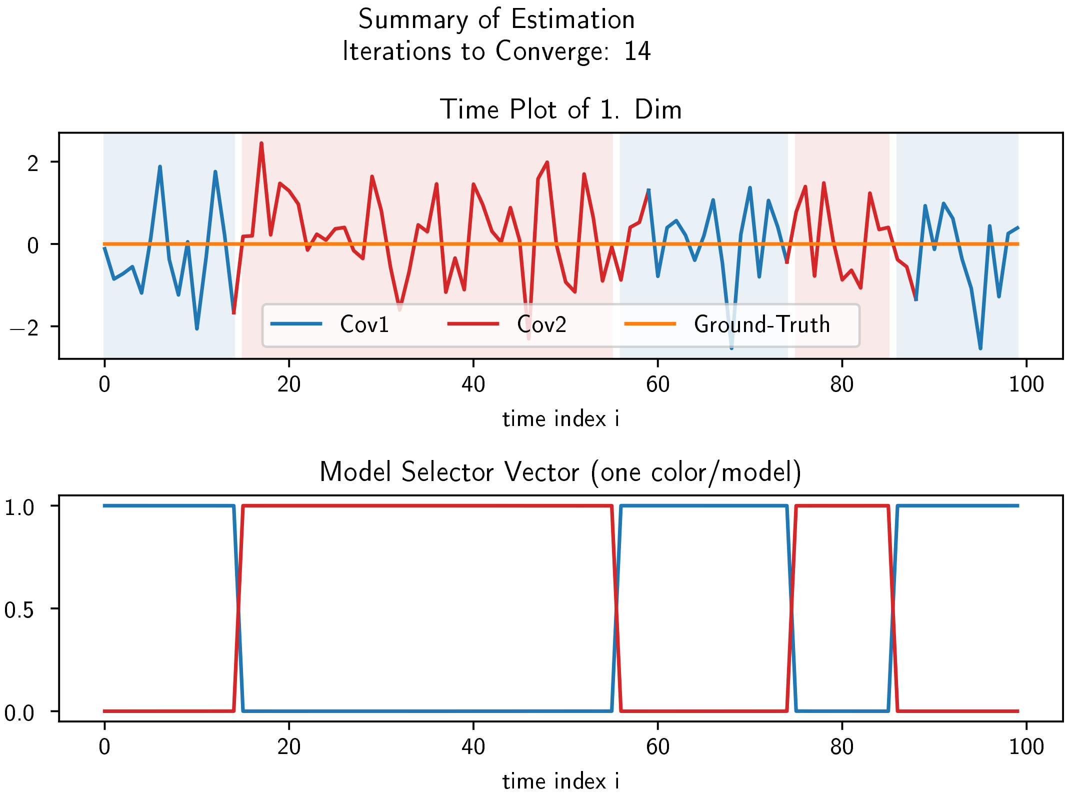


Figure 3.12: Estimation summary of the model selector distinguishing between different correlation patterns. Note that the shadings in the upper Subplot indicate the final estimations, i.e., it perfectly corresponds to the estimated S_i in the lower Subplot.

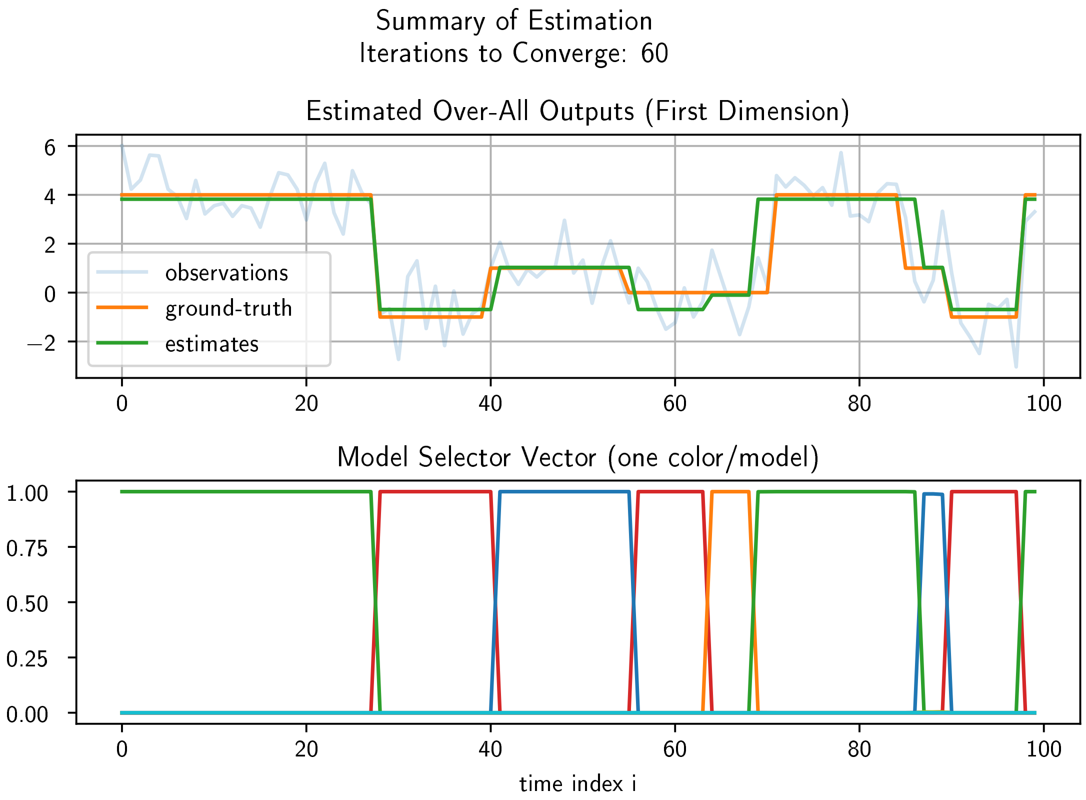


Figure 3.13: Estimation summary of CLF model with $M = 20$.

This effect is demonstrated in Figures 3.13 and 3.14. The given observations are again exactly the same as those in Subsection 3.3.1 (i.e., $N = 100$, $D = 1$, $M = 4$, and $\sigma^2 = 1$), but this time the estimator initially assumes $M_{\text{init}} = 20$ models. The resulting estimate is very accurate (much more accurate than with $M = 4$), however, the computations were much more expensive as more levels had to be estimated and more iterations had to be performed. Further note that the remaining number of levels is exactly 4 (indicated by the solid lines in Figure 3.14), i.e., the method has found the exact number of needed models to fit the given data. This excellent result has been achieved with a $\beta = 3.0$. A model is considered to be unused (i.e., it is discarded) if its weight factors are never the highest in any time index.

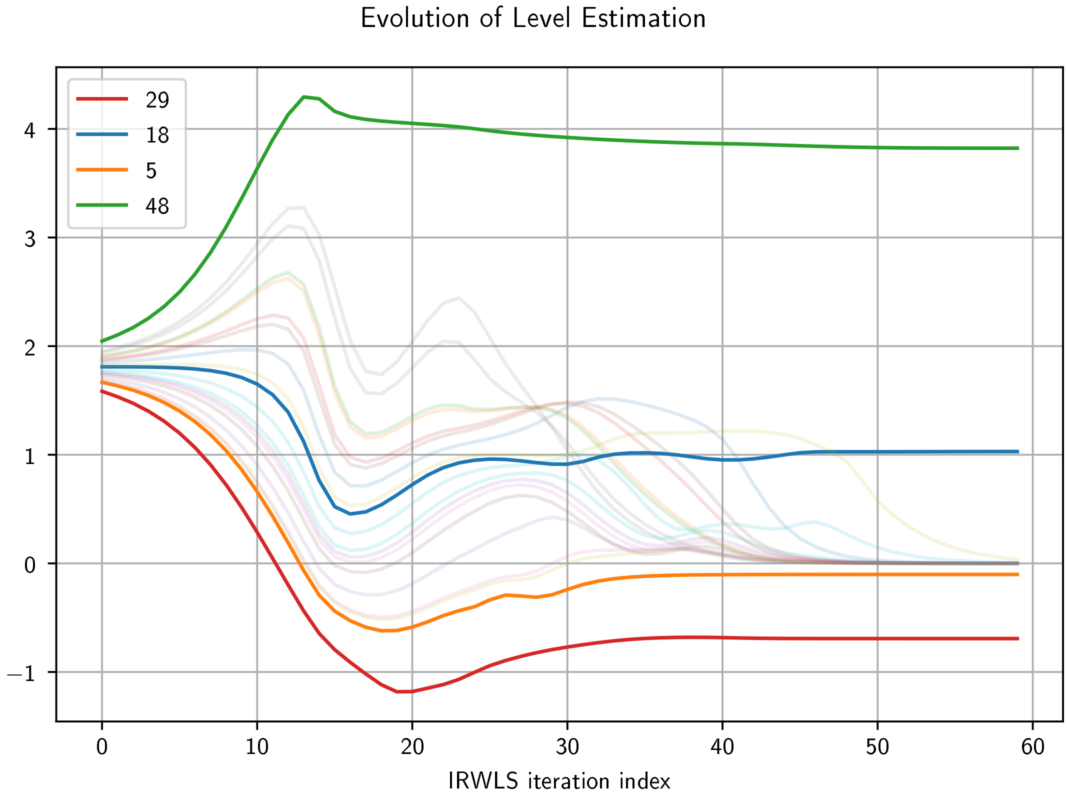


Figure 3.14: Evolution of level estimates in CLF model with $M = 20$. Note that the shaded lines indicate the discarded models. The legend denotes how often each level is used in the final estimation.

Chapter 4

Covariance Estimation

An important topic in model based estimation theory that has not yet been touched in this report is the estimation of covariance matrices. This can be useful to detect correlations between different measurements, to quantify the uncertainty of an estimation, or simply to describe some unknown observation noise. The latter has so far always been assumed to be perfectly known, but this is obviously almost never the case in practical applications (note that very coarse estimates are usually good enough for the algorithms described in Chapter 3).

This Chapter now describes a method to estimate the potentially evolving covariance matrices of zero-mean observation noise of general dimension. It builds on the work presented in [1], which describes a similar approach restricted to the scalar case. In particular, Section 4.1 develops a method to calculate exact Gaussian message passes through Normal nodes, whose covariance matrices somehow depend on a lower-level model. Given these findings, Section 4.2 then describes a method to estimate evolving covariance matrices of zero-mean observation noise. Finally, Section 4.3 applies the derived algorithm to hierarchical models.

4.1 Gaussian Message Passing Through Normal Nodes

The setup for this Section is the following. Given is an observation $y \in \mathbb{R}^D$, its known ground-truth (i.e., mean) $x \in \mathbb{R}^D$, and an additive Gaussian noise source. The positive (semi-)definite covariance matrix describing this noise is denoted by $\Sigma \in \mathbb{R}^{D\times D}$. Furthermore, it is assumed that Σ somehow depends on a lower-level quantity $J \in \mathbb{R}^{\tilde{D}}$, which itself is a Gaussian RV. The mapping between these two quantities is denoted by $g(\cdot)$, i.e.,

$$g : \mathbb{R}^{\tilde{D}} \to \mathbb{R}^{D\times D}\, ;\, J \mapsto g(J) = \Sigma\,. \tag{4.1}$$

A factor graph representing this very general setup is shown in Figure 4.1.

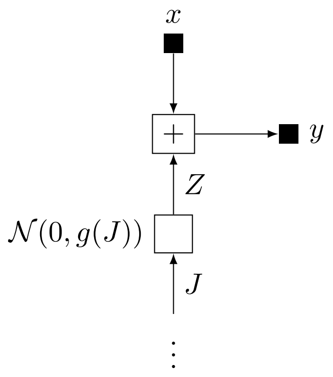


Figure 4.1: Factor graph of general setup for Gaussian message passes through normal nodes.

4.1.1 Multiplicative Factor Graph Representation of Normal Nodes

In the factor graph shown in Figure 4.1, Z is a multivariate normally distributed random vector of zero mean and covariance matrix $g(J) = \Sigma$. It is well known that Z can be expressed by

$$Z = AU, \quad A \in \mathbb{R}^{D \times D}, \quad Z, U \in \mathbb{R}^{D}\,, \tag{4.2}$$

where A is the Cholesky factor of Σ and U is the standard normal random vector, i.e.,

$$\Sigma = AA^{\intercal} \tag{4.3}$$

$$U \sim \mathcal{N}(0, \mathbf{I}_D)\,. \tag{4.4}$$

Given the expression in (4.2), the setup shown in Figure 4.1 is equivalent to the factor graph representation depicted in Figure 4.2a. Note that the vertical dots between A and J subsume some unknown transformation $\tilde{g} : \mathbb{R}^{\tilde{D}} \rightarrow \mathbb{R}^{D \times D}$, which, according to Equation (4.3), directly relates to the transformation described in (4.1). The detailed nature of either of these functions is not important at this point.

Comparing Figure 4.2a to the setup in 4.1, it can be seen that Z is assumed to be perfectly known (i.e., observed) for the considered case. Furthermore, Subsection 2.1.2 describes how Gaussian message passing is possible through multiplication nodes if one of the ingoing messages is assumed to be perfectly known. Therefore, Gaussian message passing through this multiplication node would be possible if the direction of Z was inverted. In an effort to do so, the following observation is made (similar idea is presented in [1] for the scalar case)

$$\mathcal{N}(Z; 0, g(J)) = \int_{-\infty}^{\infty} \mathcal{N}(U; 0, \mathbf{I}_D)\delta(AU - Z)\mathrm{d}U \tag{4.5}$$

$$= \frac{1}{|\det(A)|} \int_{-\infty}^{\infty} \mathcal{N}(U; 0, \mathbf{I}_D)\delta\big(U - A^{-1}Z\big)\mathrm{d}U \tag{4.6}$$

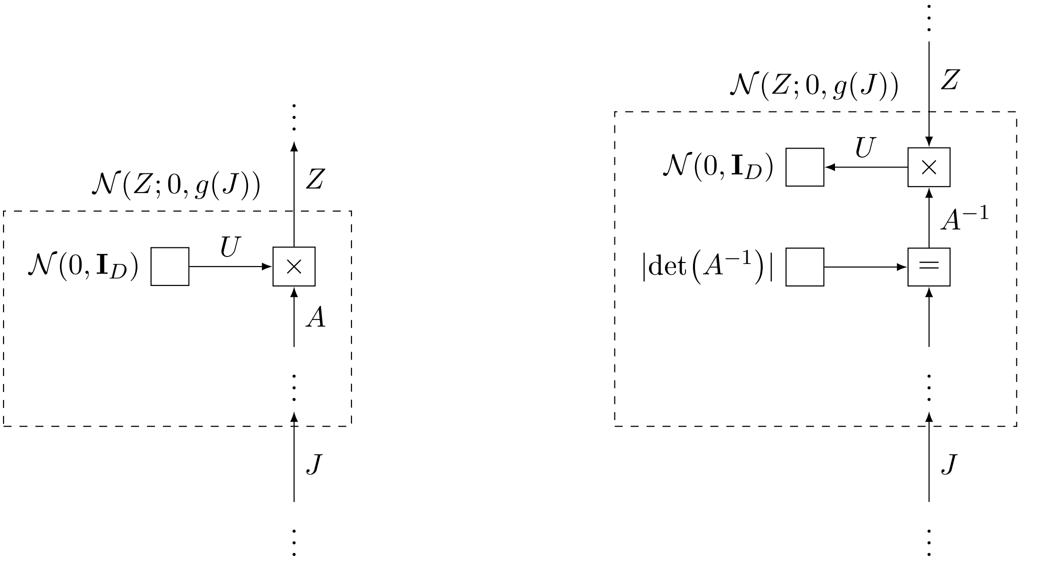
$$= |\det\big(A^{-1}\big)| \int_{-\infty}^{\infty} \mathcal{N}(U; 0, \mathbf{I}_D)\delta\big(U - A^{-1}Z\big)\mathrm{d}U\,. \tag{4.7}$$

Here, $\delta(\cdot)$ denotes the Dirac-delta function. The equality in (4.6) follows from Proposition 2.3 in [10]. The equality in (4.7) used the fact that

$$1 = \left|\det\left(AA^{-1}\right)\right| = \left|\det(A)\det\left(A^{-1}\right)\right| = \left|\det(A)\right| \left|\det\left(A^{-1}\right)\right| \tag{4.8}$$

$$\Rightarrow \left|\det\left(A^{-1}\right)\right| = \frac{1}{\left|\det(A)\right|}\,. \tag{4.9}$$

Given the expression in (4.7), it can be concluded that the factor graph representation shown in Figure 4.2b is indeed equal to the one in 4.2a.



(a) Based on Equation (4.2). (b) Alternative representation.

Figure 4.2: Multiplicative factor graph representations of normal node.

At this point, the following three facts are noted:

- For fixed Z, the elements of A^{-1} are jointly normal distributed.
- Because A^{-1} is the inverse of an upper-triangular matrix A (property of Cholesky factors), it is an upper-triangular matrix itself. Therefore, it has at most $\tilde{D} = D(D+1)/2$ non-zero elements, which are in the following denoted by

$$A^{-1} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1D} \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \tilde{a}_{DD} \end{bmatrix}. \tag{4.10}$$

- The absolute value node in Figure 4.2b can simply be computed as

$$|\det(A^{-1})| = \prod_{d=1}^{D} |\tilde{a}_{dd}|\,. \tag{4.11}$$

This is particularly helpful as there exists a NUV representation for scalar absolute values, namely the Laplace NUV prior with $\beta = -1$ (described in Subsection 2.2.1).

4.1.2 Definition of J

Initially, it has been stated that J is seen as a lower-level quantity that describes the noise covariance matrix Σ. Subsection 4.1.1 then derives a manageable factor graph representation of normal nodes in terms of the inverse Cholseky factor of Σ, denoted by A^{-1}. In this representation, the elements of A^{-1} are seen to be jointly normal distributed. Considering all these facts, it appears straightforward to define the lower-level quantity J as the stacked non-zero elements of A^{-1}, i.e.,

$$J \triangleq \begin{bmatrix} \tilde{a}_{11} \; \tilde{a}_{22} \; \ldots \; \tilde{a}_{\mathrm{DD}} \big| \tilde{a}_{12} \; \tilde{a}_{23} \; \ldots \; \tilde{a}_{\mathrm{D\text{-}1D}} \big| \ldots \big| \tilde{a}_{\mathrm{1D}} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{\tilde{D}} . \tag{4.12}$$

Thereby, the dimension $\tilde{D}$ is equal to the number of non-zero elements in an upper-triangular matrix of dimension D, i.e.,

$$\tilde{D} = \frac{D(D+1)}{2} . \tag{4.13}$$

Note that J as defined in (4.12) perfectly describes Σ. Its direct relation to A^{-1} motivates its name Vectorized Inverse Cholesky Factor (VICF).

Given this definition of J, the setup in Figure 4.1 can alternatively be expressed by the factor graph in Figure 4.3. Thereby, the matrix $\tilde{Z}$ is constructed from the observed $Z = \begin{bmatrix} z_1 & z_2 & \ldots & z_D \end{bmatrix}^{\mathsf{T}}$ as

$$\tilde{Z} \triangleq \left[\begin{array}{ccccc|cccc|c|c} z_1 & 0 & \ldots & 0 & 0 & z_2 & 0 & \ldots & 0 & & z_{\mathrm{D}} \\ 0 & z_2 & \ldots & 0 & 0 & 0 & z_3 & \ldots & 0 & & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & \ldots & z_{\mathrm{D\text{-}1}} & 0 & 0 & 0 & \ldots & z_{\mathrm{D}} & & 0 \\ 0 & 0 & \ldots & 0 & z_{\mathrm{D}} & 0 & 0 & \ldots & 0 & & 0 \end{array} \right] \tag{4.14}$$

$$= \begin{bmatrix} \tilde{Z}_1 & \tilde{Z}_2 & \ldots & \tilde{Z}_{\mathrm{D}} \end{bmatrix} \in \mathbb{R}^{D \times D(D+1)/2} , \tag{4.15}$$

where

$$\tilde{Z}_d \triangleq \left[\begin{array}{cccc} z_{\mathrm{d}} & 0 & \ldots & 0 \\ 0 & z_{\mathrm{d+1}} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & z_{\mathrm{D}} \\ \hline 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 \end{array} \right] \in \mathbb{R}^{D \times (D-d+1)} . \tag{4.16}$$

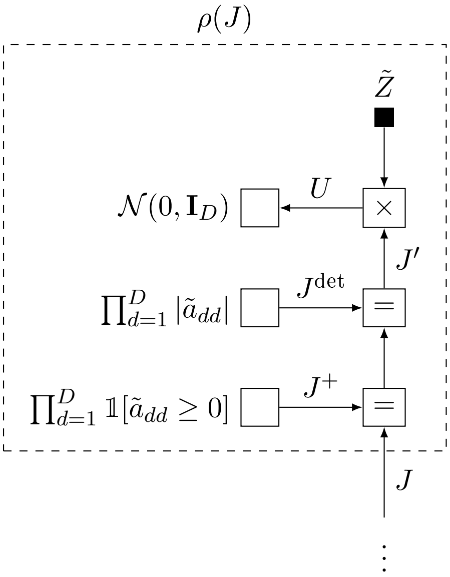

$\rho(J)$
$\tilde{Z}$
$\mathcal{N}(0, \mathbf{I}_D)$
U
$\times$
J'
$\prod_{d=1}^{D} |\tilde{a}_{dd}|$
J^{det}
$=$
$\prod_{d=1}^{D} \mathbb{1}[\tilde{a}_{dd} \geq 0]$
J^{+}
$=$
J

Figure 4.3: Alternative factor graph representation of the setup in 4.1, based on the definition of J and $\tilde{Z}$.

Note that the structure of $\tilde{Z}$ is a direct consequence of the definition of J, as the following equality must hold

$$U = A^{-1}Z = \tilde{Z}J\,. \tag{4.17}$$

As already mentioned previously, the prior on J^{det} in Figure 4.3 can be implemented by a Laplace NUV prior with $\beta = -1$. The additional positivity prior on the first D elements of J (i.e., the prior on J^{+}) is needed to find any non-trivial solution of J (same argument has been made in [1]).

4.1.3 Resulting Gaussian Messages Passes Through J

Given the factor graph in Figure 4.3, it is now possible to find expressions for the Gaussian messages passed through the normal node in 4.1.

First of all, it is noted that the two NUV priors generating the messages trough J^{det} and J^{+} are only applied to the first D elements of J. The corresponding messages per dimension are

$$\overrightarrow{\xi}_{J_d^{\text{det}}} = 0 \tag{4.18}$$

$$\overrightarrow{W}_{J_d^{\text{det}}} = -\frac{1}{\hat{m}_{J_d}^2} \tag{4.19}$$

$$\overrightarrow{\xi}_{J_d^{+}} = \beta \tag{4.20}$$

$$\overrightarrow{W}_{J_d^{+}} = \frac{\beta}{|\hat{m}_{J_d}|}\,, \tag{4.21}$$

where $\hat{m}_{J_d}$, $d \in \{1, \ldots, D\}$ denotes the d-th element of the current mean estimate of J. Note that these NUV priors are further explained in Subsections 2.2.2 and 2.2.3, respectively. The backward messages through the multiplication node are

$$\overleftarrow{\xi}_{J'} = \mathbf{0}_{\tilde{D}} \tag{4.22}$$

$$\overleftarrow{W}_{J'} = \tilde{Z}^\mathsf{T} \tilde{Z}\,, \tag{4.23}$$

where $\mathbf{0}_{\tilde{D}}$ denotes an all-zero vector of dimension $\tilde{D}$.

Accordingly, the backward messages through J (i.e., the resulting Gaussian message passes) are

$$\overleftarrow{\xi}_{J} = \sum_{d=1}^{D} C_d \cdot \beta \tag{4.24}$$

$$\overleftarrow{W}_{J} = \tilde{Z}^\mathsf{T} \tilde{Z} + \sum_{d=1}^{D} C_d C_d^\mathsf{T} \cdot \frac{\beta |\hat{m}_{J_d}| - 1}{\hat{m}_{J_d}^2}\,, \tag{4.25}$$

where C_d is a column-vector of dimension $\tilde{D}$, whose values are all zero except for its d-th value being 1. Note that, depending on the value of β in (4.25), the resulting precision messages can have negative values on the diagonal. This poses no problem as long as the resulting posterior variance estimates of J are positive. Accordingly, β should be chosen high enough.

4.2 Covariance Matrix Estimation Model

The previous Section 4.1 derives exact expressions for the Gaussian message passes through normal nodes. Thereby, it is assumed that the covariance matrices of these normal nodes depend on the VICF, usually denoted by J. J is a Gaussian random vector, which means that its behaviour can be modelled by any known evolution model for Gaussian random vectors. A factor graph describing this setup is shown in Figure 4.4. Based on it, the following Subsections describe a method to estimate the evolving covariance matrices of zero-mean Gaussian noise.

4.2.1 Estimation of VICF

The general setup for the estimation of $J_i \in \mathbb{R}^{\tilde{D}}$ is given in Figure 4.4. Thereby, both the observations $y_i \in \mathbb{R}^D$ and their corresponding ground-truths (i.e., means) $x_i \in \mathbb{R}^D$ are assumed to be known. This in particular also includes the setting of pure zero-mean Gaussian noise (i.e., $x_i = 0$, $\forall i \in \{1, \ldots, N\}$). The corresponding values of $Z_i \in \mathbb{R}^D$ are therefore completely known and can be calculated as

$$Z_i = y_i - x_i = \begin{bmatrix} z_{i,1} & z_{i,2} & \ldots & z_{i,D} \end{bmatrix}^\mathsf{T}. \tag{4.26}$$

Accordingly, the $\tilde{Z}_i$ matrices can be constructed following the definition in Equation 4.14. Note that the time indices i are omitted in this definition.

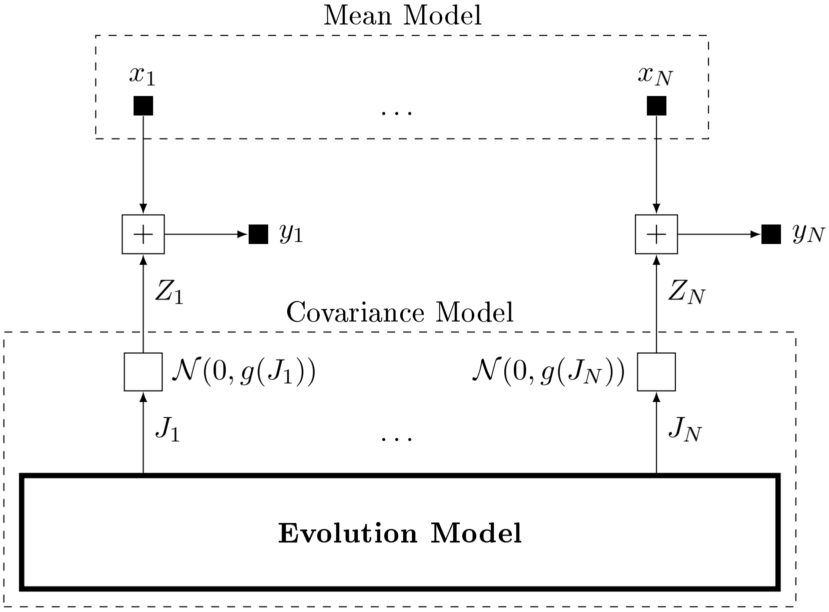


Figure 4.4: Factor graph for covariance estimation model.

With the results of (4.26) and some initial values for $\hat{m}_{J_i}$ (i.e., mean estimates of J_i), the backward messages given to the evolution model can be calculated according to Equations (4.24) and (4.25). Note that the time indices are again omitted. Based on these ingoing messages to the evolution model, the corresponding estimates of J_i can be calculated (if J_i is assumed to be PWC, the method in Subsection 2.3.3 can be applied). This whole process, including the calculation of the ingoing messages and the updated estimates of J_i, constitutes one iteration of IRWLS in the context of Figure 4.4. Note that for the next iteration, the ingoing messages passed to the evolution model must be recomputed based on the new estimates of $J_i = \hat{m}_{J_i}$.

Initial Values of $\hat{m}_{J_i}$

Initializing the values of $\hat{m}_{J_i}$ requires a bit more thought than in previously discussed algorithms. The reason for that is the implicit meaning of the mean values of J_i. According to the definition in (4.12), the elements of J_i relate to the inverse of the Cholesky factor of the corresponding noise covariance matrix Σ_i. Therefore, initially setting all values of $\hat{m}_{J_i}$ simply to zero can cause numerical problems (observed during simulations). It is instead proposed to set the first D values of $\hat{m}_{J_i}$ to 1 and the remaining values to 0. This corresponds to initially assuming that the observation noise is i.i.d. with unit variance, i.e.,

$$\Sigma_{i,\text{init}} = \mathbf{I}_D \, . \tag{4.27}$$

D	$\tilde{D}$	$\tilde{D}^2$
1	1	1
3	6	36
10	55	3025
20	210	44100
30	465	216225
⋮	⋮	⋮

Table 4.1: Table listing D, $\tilde{D}$, and $\tilde{D}^2$.

Dimension of VICF and the Limitations it Poses

Equation (4.13) states that the dimension of J_i (denoted by $\tilde{D}$) grows quadratically in the dimension of the observations (denoted by D). Because J_i itself is a jointly normal distributed random vector, its estimation requires handling matrices of dimension $\tilde{D} \times \tilde{D}$. The number of elements of these matrices therefore grows quartically in D, obviously limiting the practical applications of the described method to cases of small dimension. For comparison, Table 4.1 lists some choices for D, the corresponding $\tilde{D}$, and the resulting number of elements describing the covariance matrices of J_i (i.e., $\tilde{D}^2$). Note that simulations have shown the method to work for up to a dimension of $D = 10$.

4.2.2 Recovering Noise Covariance Matrices from VICF

Recall the definitions of the noise covariance matrices

$$\Sigma_i = A_i A_i^{\mathsf{T}} , \tag{4.28}$$

their inverse Cholesky factors

$$A_i^{-1} = \begin{bmatrix} \tilde{a}_{i,11} & \tilde{a}_{i,12} & \dots & \tilde{a}_{i,1D} \\ 0 & \tilde{a}_{i,22} & \dots & \tilde{a}_{i,2D} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \tilde{a}_{i,DD} \end{bmatrix} , \tag{4.29}$$

and the VICF

$$J_i = \left[\tilde{a}_{i,11} \; \tilde{a}_{i,22} \; \dots \; \tilde{a}_{i,\mathrm{DD}} \middle| \tilde{a}_{i,12} \; \tilde{a}_{i,23} \; \dots \; \tilde{a}_{i,\mathrm{D\text{-}1D}} \middle| \dots \middle| \tilde{a}_{i,\mathrm{1D}} \right]^{\mathsf{T}} . \tag{4.30}$$

Therefore, given the mean estimates of J_i as $\hat{m}_{J_i}$, the estimates of A_i^{-1} can be constructed as

$$\hat{A}_i^{-1} = \begin{bmatrix} \hat{m}_{J_{i,1}} & \hat{m}_{J_{i,D+1}} & \dots & \hat{m}_{J_{i,D(D+1)/2}} \\ 0 & \hat{m}_{J_{i,2}} & \dots & \hat{m}_{J_{i,D(D+1)/2-1}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{m}_{J_{i,D}} \end{bmatrix} . \tag{4.31}$$

With these reconstructed $\hat{A}_i^{-1}$, the estimated noise covariance matrices $\hat{\Sigma}_i$ can be calculated as

$$\hat{\Sigma}_i = \hat{A}_i \hat{A}_i^{\mathsf{T}} \,, \tag{4.32}$$

where $\hat{A}_i$ is recovered as

$$\hat{A}_i = \left(\hat{A}_i^{-1}\right)^{-1} . \tag{4.33}$$

Note that the inversion in Equation (4.33) is computationally cheap as $\hat{A}_i^{-1}$ is an upper-triangular matrix. However, this inversion can be omitted entirely if, instead of the noise covariance matrices, its precision matrices are estimated. In this case, $\hat{\Sigma}_i^{-1}$ can be directly calculated from $\hat{A}_i^{-1}$ as

$$\hat{\Sigma}_i^{-1} = \left(\hat{A}_i^{-1}\right)^{\mathsf{T}} \hat{A}_i^{-1} . \tag{4.34}$$

These results as well as the expressions for the backward messages through J_i are summarized in the Appendix, Table A.2.

4.2.3 Results for Estimating PWC Covariance Matrices of Zero-Mean Gaussian Noise

This Subsection puts the derived method to the test by estimating the PWC covariance matrices of zero-mean Gaussian noise. For the discussed simulation, $N = 100$ observations of dimension $D = 3$ are considered. They are all generated by a zero-mean normal distribution whose covariance matrices change twice. For the first 30 observations, i.i.d. noise with unit variance is used, i.e.,

$$\Sigma_i = \mathbf{I}_D, \quad i \in \{1, \ldots, 30\} . \tag{4.35}$$

For the next 40 observations, the noise level is increased by a factor of 10. Furthermore, the first and second dimension correlate perfectly, i.e.,

$$\Sigma_i = \begin{bmatrix} 10 & 10 & 0 \\ 10 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad i \in \{31, \ldots, 70\} . \tag{4.36}$$

Finally, in the last 30 observations, the second and third dimension correlate negatively, i.e.,

$$\Sigma_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad i \in \{71, \ldots, 100\} . \tag{4.37}$$

Figure 4.5 shows the generated observations. Thereby, the upper Subplot depicts the first dimension of all observations in chronological order. Note that the three different covariance matrices used to generate the data are indicated by three different colours. The lower three Subplots show scatter plots of the observations, visualizing the correlation between the dimensions. The used colour scheme matches the one used in the upper Subplot.

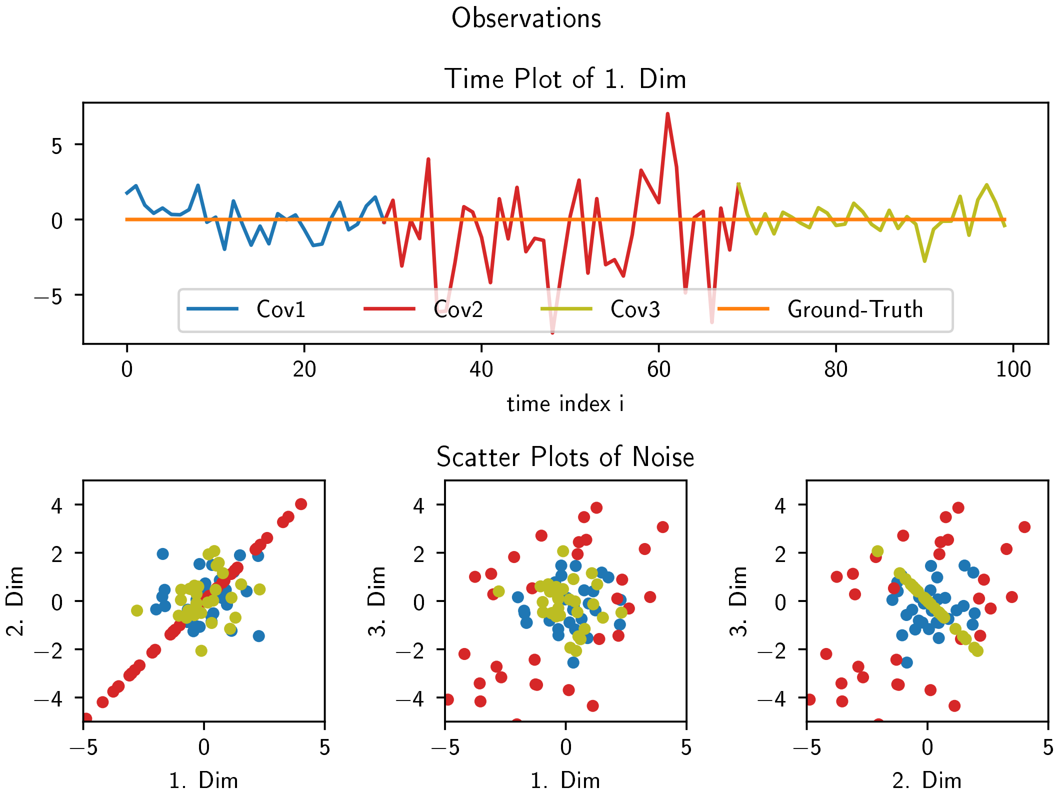


Figure 4.5: Generated observations for covariance estimation model.

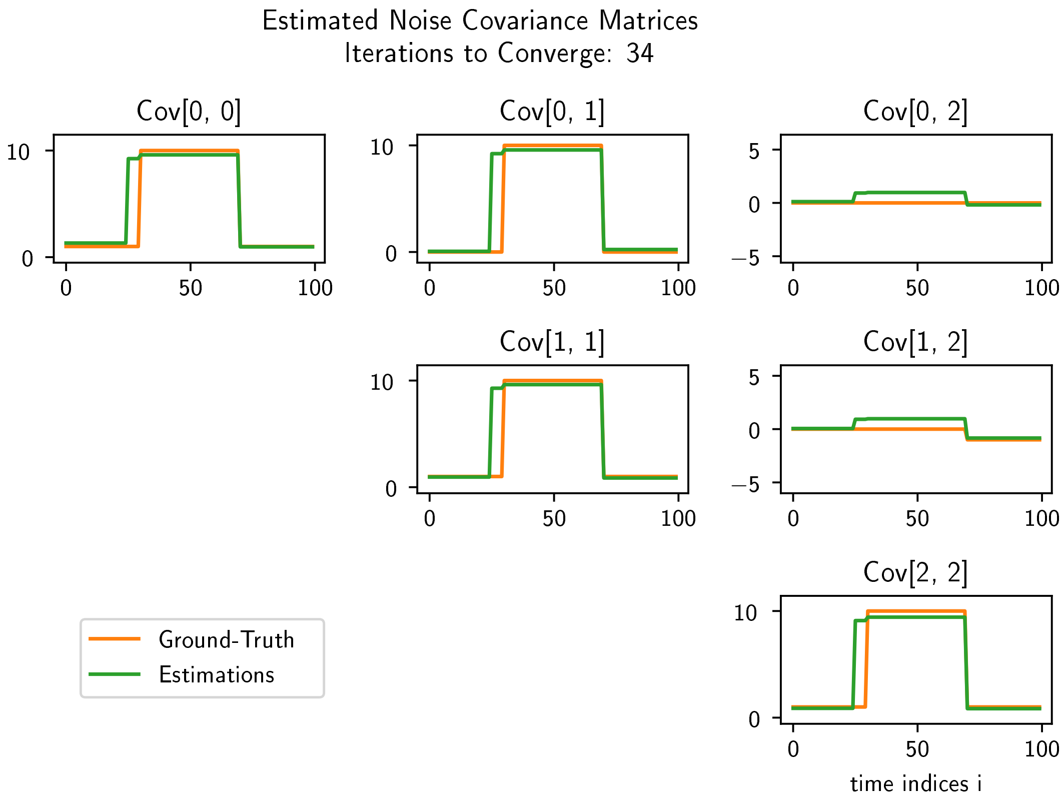


Figure 4.6: Estimated PWC covariance matrices from covariance estimation model.

Figure 4.6 shows the estimated covariance matrices given the previously described observations. In particular, the estimated values of each element of Σ_i are depicted in separate Subplots. The spacial ordering of these Subplots corresponds to the position of the depicted element in the matrix. Note that only the elements on the upper-half (i.e., upper-triangular) are shown because Σ_i is symmetric, meaning that the lower half would simply be a mirrored copy of the upper half.

Interpretation of Results for Covariance Estimation

Figure 4.6 shows the estimated values of Σ_i, where each element of it is plotted in a separate subplot. Note that the placing of these subplots correspond to the placing of the element in the actual covariance matrix. The lower half of it is omitted because Σ is symmetric, i.e., these plots would simply be mirrored copies of the plots in the upper half. Thereby, the estimates are shown in green and the underlying ground-truth in orange. It is observed that the estimates are fairly accurate (considering that only ≈ 30 observations are given per covariance matrix). In particular, the estimator seems to be able to detect changes in the observation noise covariance matrices and calculate the new values accurately. In conclusion, the results in Figure 4.6 suggest that the proposed method works very well in the simulated case.

4.3 Application of Covariance Estimator to Hierarchical Models

The previously developed method to estimate covariance matrices of zero-mean Gaussian noise is used in this Section to build a model estimating both the mean and noise covariance matrices of given observations. This application is inspired by an example made in [1] and demonstrates the power of the found method. Such two-folded models are in some sense hierarchical, as a directly accessible model describes the mean of the observations (upper-level) and some hidden model the observed noise (lower-level). A factor graph visualizing this is shown in Figure 4.7. Note how both evolving quantities (i.e., the means and VICFs) are expressed by general evolution models, e.g., PWC models.

Performing estimation tasks in Figure 4.7 causes similar problems as those discussed in Subsection 3.2.1. Namely, the developed covariance estimator relies on known (i.e., observed) means X_i and the resulting factor graph is no longer loop-free. The proposed solution to these problems is again an iterative approach. In particular, the following two estimation improvement steps are performed iteratively.

- Improve estimates of VICFs J_i for fixed means $X_i = \hat{m}_{X_i}$.
- Improve estimates of means X_i for fixed observation noise precision matrices $\hat{\Sigma}_i = g(\hat{m}_{J_i})$ (expression for $g(\cdot)$ is given in Equation (4.32)).

Note that the means X_i can also be estimated based on $\hat{\Sigma}_i^{-1}$, which can be calculated from $\hat{m}_{J_i}$ according to Equation (4.34). This approach is generally less computationally expensive.

4.3.1 PWC Hierarchical Model

In this example, both evolution models in Figure 4.7 are assumed to be PWC (Section 2.3). Accordingly, a similar setup as discussed in Subsection 4.2.3 is used, with the only difference being that the means evolve PWC over time too (i.e., $N = 100$, $D = 3$, noise covariance matrices as in Equations (4.35) to (4.37)). A summary of the generated observations is shown in Figure 4.8.

Estimations in PWC Hierarchical Model

Plots of the resulting mean and noise covariance matrix estimates are shown in Figures 4.9 and 4.10, respectively. In both plots, the final estimates are depicted in green and the underlying ground-truths in orange. Furthermore, Figure 4.9 also shows the results that are achieved with a simple PWC mean estimation model, where i.i.d. noise with unit variance is assumed for all observations (green dashed line). This shows how

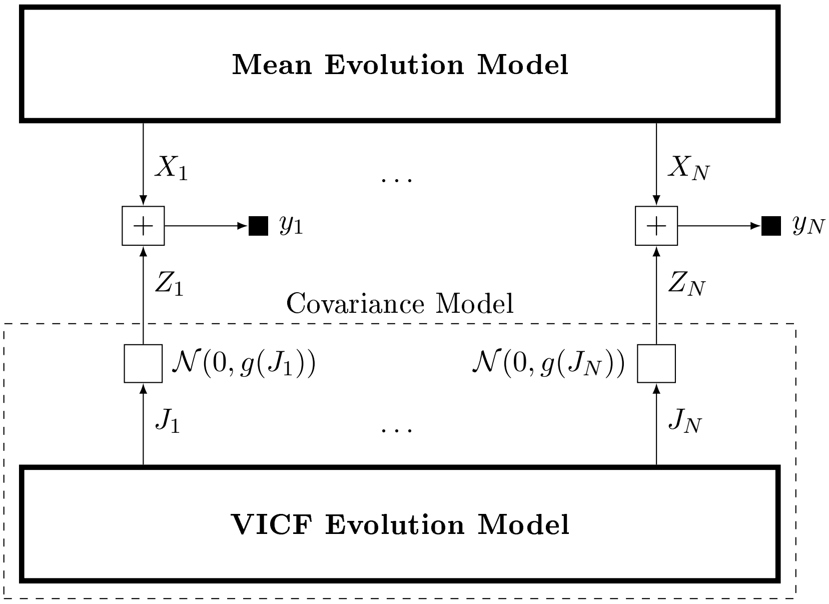


Figure 4.7: Factor graph of hierarchical model.

the estimation of the noise covariance matrices is crucial in cases where its behaviour evolves over time, as the performance of the mean estimator obviously decreases when its assumption about Σ_i is too far off (occurs in Figure 4.9, observations 30 to 70). The estimation of the noise covariance matrices are comparably good as those achieved for zero-mean Gaussian noise (Subsection 4.2.3), which is expected as the mean estimates are almost perfect. Over all, the presented results suggest that the proposed method indeed works in the presented simulation.

4.3.2 Connecting Multiple Hierarchical Models with Model Selector Mechanism

A step forward from this point on could be to connect multiple hierarchical models (as in Figure 4.7) by the model selector mechanism described in the previous Chapter 3. This could, for example, be useful to model PWC data with known base levels, where the added Gaussian noise needs to be estimated too. A sketch of such a model is shown in Figure 4.11. Unfortunately, there hasn't been time during this project to further investigate this idea. However, this could be one of the immediate next steps if the investigation of this topic should be continued.

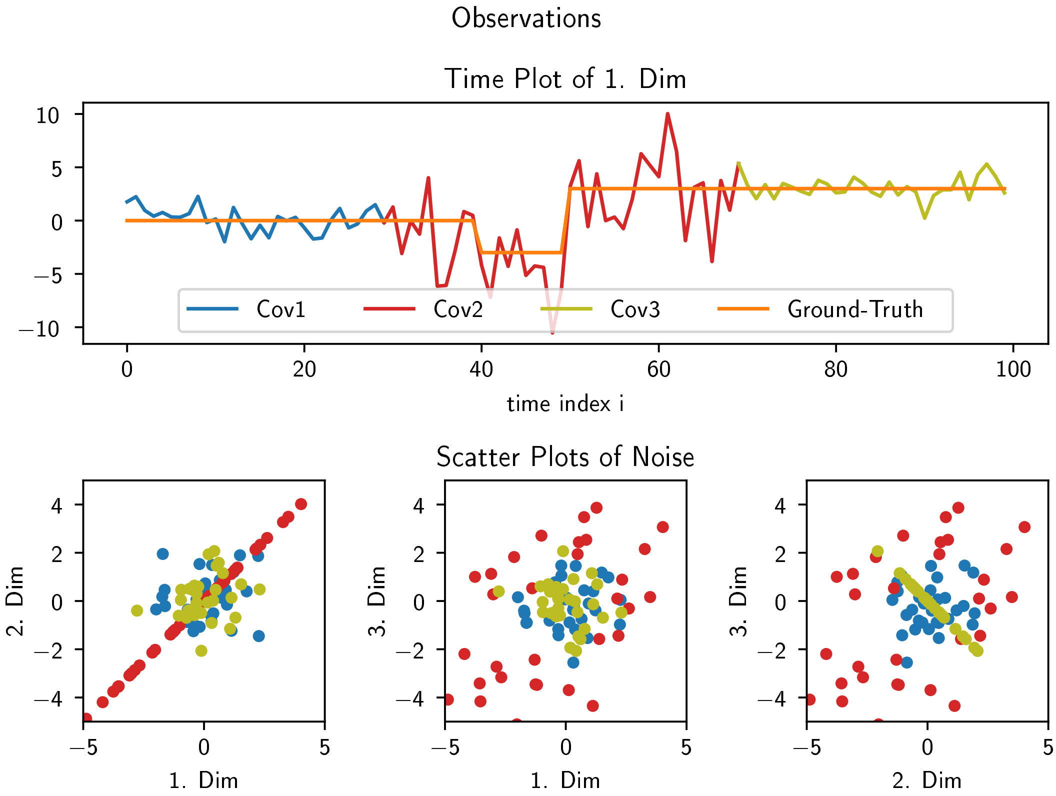


Figure 4.8: Generated observations for PWC hierarchical model.

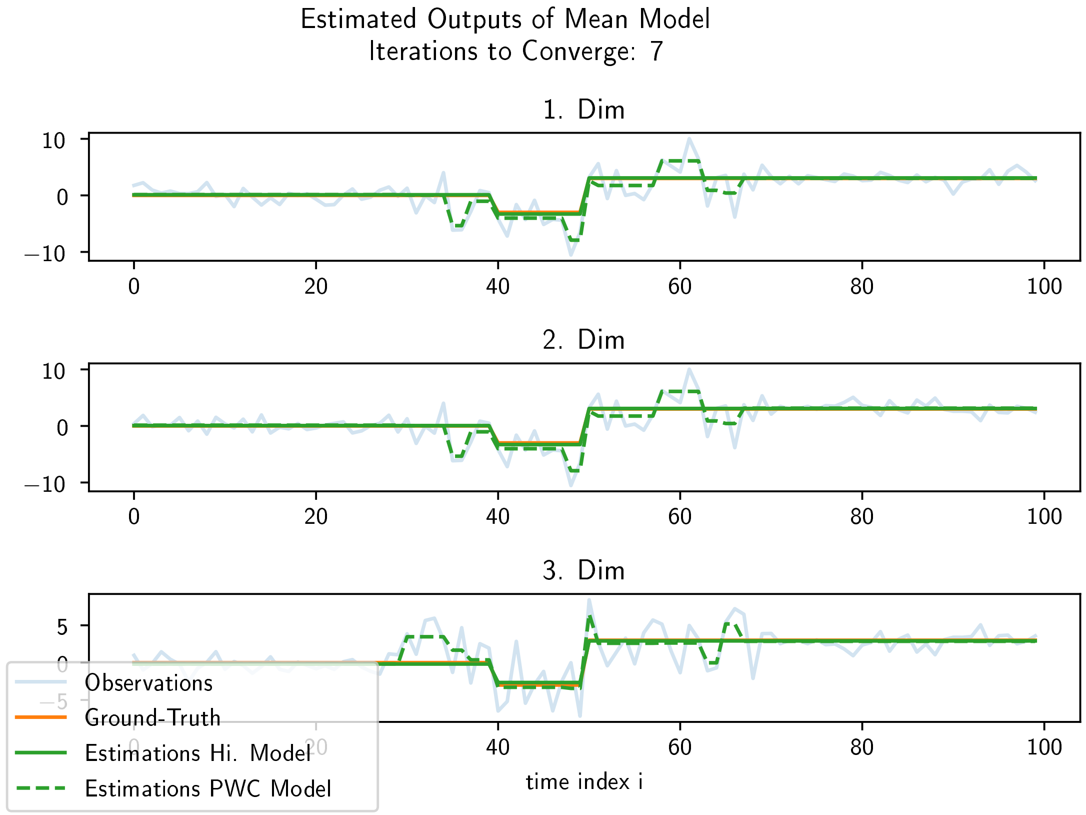


Figure 4.9: Estimated means in PWC hierarchical model (green solid lines). For comparison, the resulting estimates achieved with a simple PWC model are showed too (green dahed lines).

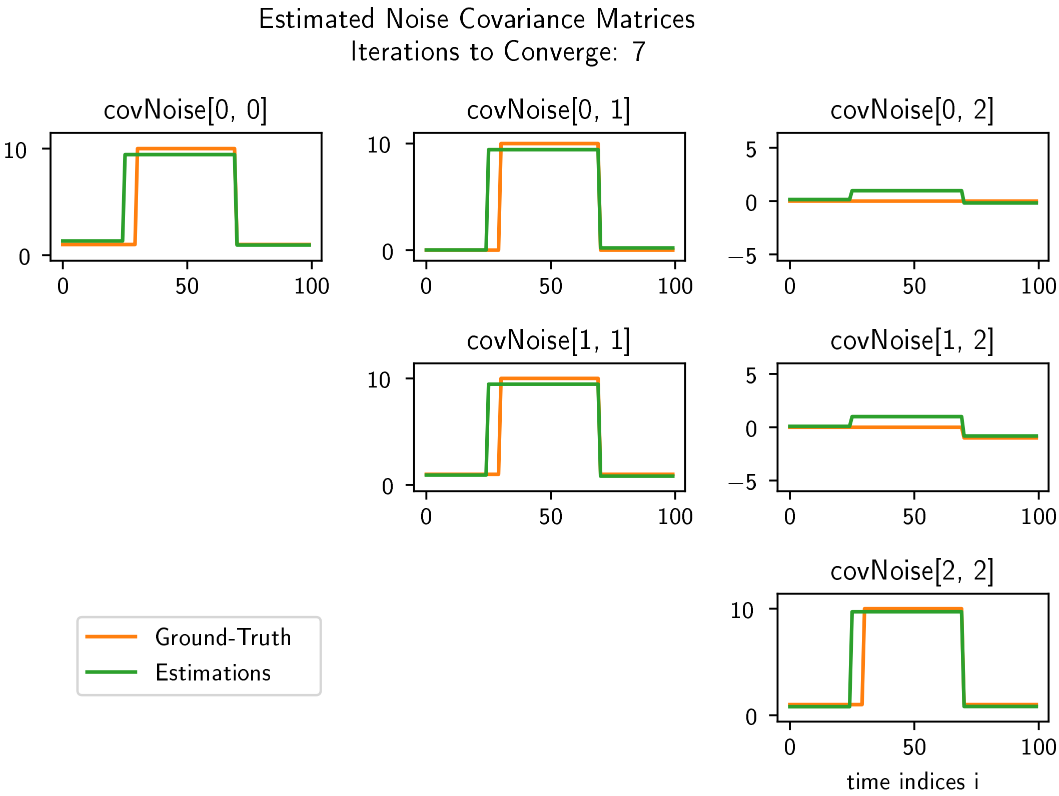


Figure 4.10: Estimated noise covariance matrices in PWC hierarchical model.

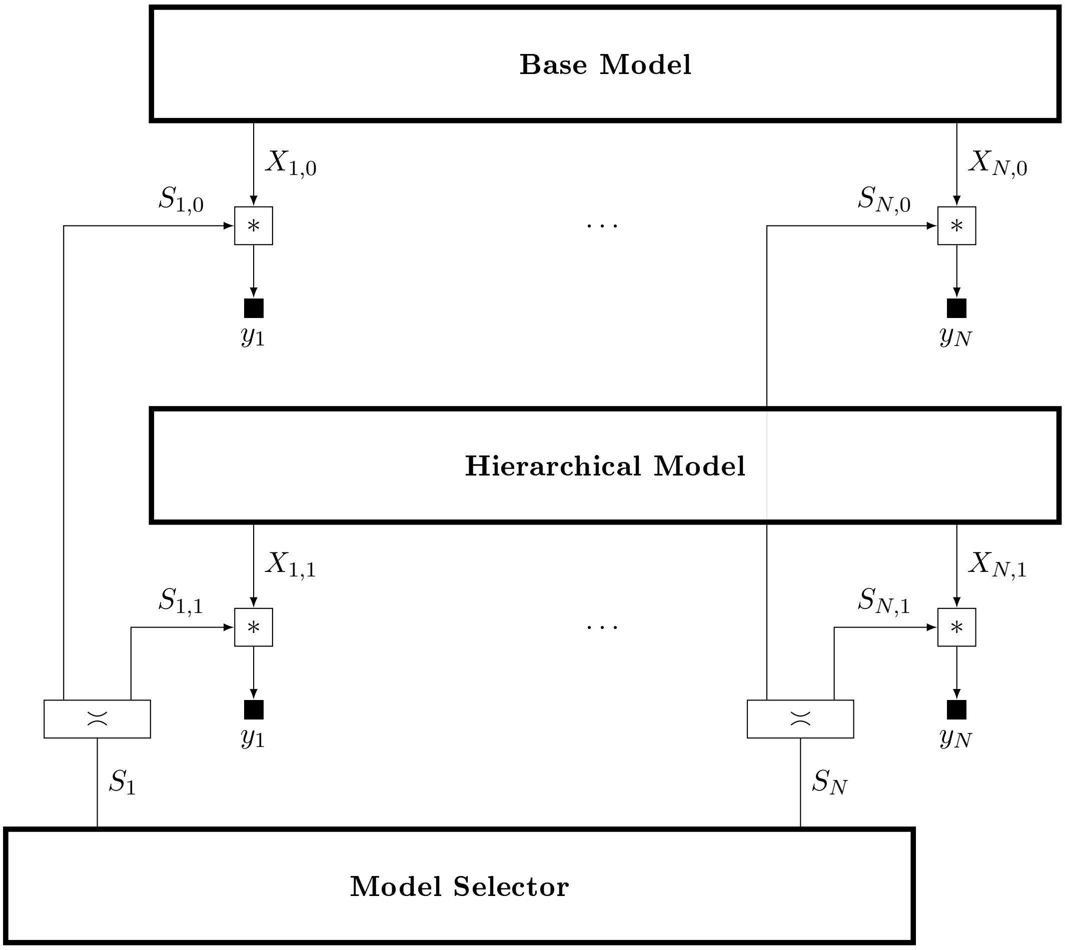


Figure 4.11: Factor graph sketch of RTB model with unknown noise. The $*$ boxes denote some unspecified operation, somehow connecting the models with the selector mechanism.

Chapter 5

Conclusion

This report discusses two novel NUV-based approaches to the problem settings stated in the Introduction. The first method is explained in Chapter 3, describing an algorithm to estimate which sections of a set of given observations have been generated by which specified model. This model selector mechanism (factor graph given in Figure 3.2) is built around the powerful One-Hot NUV prior (Section 3.1), shaping the estimated model selector vector in the desired way (summarized in Table A.1). Furthermore, Section 3.3 discusses a variety of advanced applications for the model selector mechanism. There, the proposed method is shown to work for the respective applications by well-documented simulations, showcasing its enormous potential and easy-to-use nature.

The second discussed method tackles the problem of estimating potentially evolving covariance matrices of zero-mean Gaussian noise. It is extensively documented in Chapter 4, pointing out many different nuances of the developed algorithm. One particularly interesting aspect is the definition of the Vectorized Inverse Cholesky Factor (VICF), based on which exact expressions for Gaussian messages passed through normal nodes are derived (summarized in Table A.2). These findings are then used to create a method able to estimate basically any kind of evolving noise covariance matrices (as long as the evolution of the VICF can be processed). As a proof of concept, the found algorithm is applied to estimate the PWC covariance matrices of zero-mean Gaussian noise in the three-dimensional case, resulting in a very accurate estimate (Figure 4.6). Furthermore, the proposed method is also applied to a hierarchical model, again showcasing very good performance (Figures 4.9 and 4.10).

In conclusion, it can be clearly stated that the two fundamental methods proposed in this report yield very promising results. In fact, they both are shown to be valid approaches to the initially stated problem settings. However, there remain a lot of open questions that need to be answered and some burning issues that must be resolved. But the taken path promises great potential and should definitely be further pursued!

Appendix A

Summary of Newly Proposed Composite NUV Priors

The following tables briefly summarise all important equations to calculate the messages generated by the One-Hot (Table A.1) and Normal Node (Table A.2) NUV priors. The latter also describes how the resulting estimates of the underlying noise covariance matrices can be recovered from the estimated VICF.

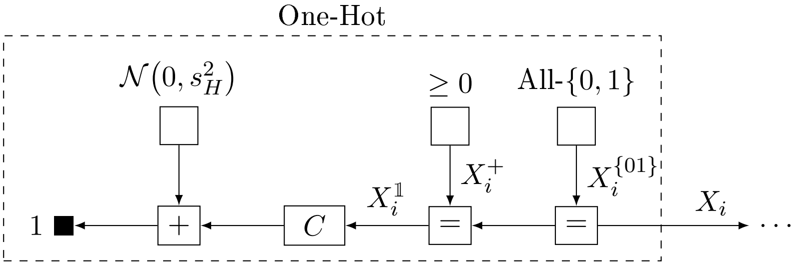
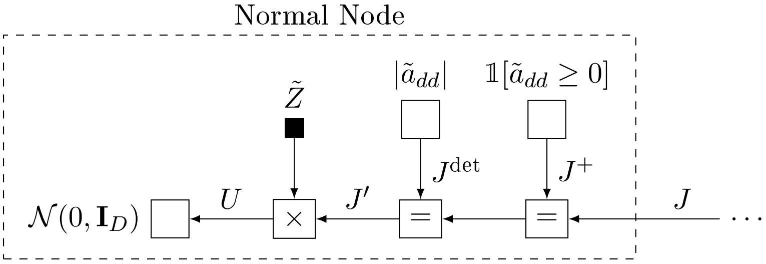
 One-Hot $\mathcal{N}(0, s_H^2)$ ≥ 0 All-$\{0,1\}$ X_i^+ $X_i^{\{01\}}$ $X_i^{\mathbb{1}}$ X_i 1 $+$ C $=$ $=$ $\cdots$	
$\begin{aligned} \overrightarrow{\xi}_{X_i} &= s_H^{-2} C^\mathsf{T} + \overrightarrow{\xi}_{X_i^+} + \overrightarrow{\xi}_{X_i^{\{01\}}} \\ \overrightarrow{W}_{X_i} &= s_H^{-2} C^\mathsf{T} C + \overrightarrow{W}_{X_i^+} + \overrightarrow{W}_{X_i^{\{01\}}}\,, \end{aligned}$	$\begin{aligned} \overrightarrow{m}_{X_i} &= \overrightarrow{m}_{X_i'} + \overrightarrow{V}_{X_i'} C^\mathsf{T} G_i \Big(1 - C \overrightarrow{m}_{X_i'}\Big) \\ \overrightarrow{V}_{X_i} &= \overrightarrow{V}_{X_i'} - \overrightarrow{V}_{X_i'} C^\mathsf{T} G_i C \overrightarrow{V}_{X_i'}\,, \end{aligned}$ with $\begin{aligned} \overrightarrow{V}_{X_i'} &\triangleq \Big(\overrightarrow{W}_{X_i^+} + \overrightarrow{W}_{X_i^{\{01\}}}\Big)^{-1} \\ \overrightarrow{m}_{X_i'} &\triangleq \overrightarrow{V}_{X_i'} \Big(\overrightarrow{\xi}_{X_i^+} + \overrightarrow{\xi}_{X_i^{\{01\}}}\Big) \\ G_i &\triangleq \left(s_H^2 + \sum_{m=1}^{M} \overrightarrow{V}_{X_{i,m}'}\right)^{-1}, \end{aligned}$
where $s_H^2 > 0$ (for right-hand side $s_H^2 \geq 0$) and $C \triangleq \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^\mathsf{T} \in \mathbb{R}^D$.	
$\overrightarrow{\xi}_{X_i^+} = \begin{bmatrix}\beta & \beta & \dots & \beta\end{bmatrix}^\mathsf{T}$ $\overrightarrow{W}_{X_i^+} = \begin{bmatrix} \frac{\beta}{\lvert\hat{m}_{X_{i,1}}\rvert} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\beta}{\lvert\hat{m}_{X_{i,D}}\rvert} \end{bmatrix}$	$\overrightarrow{m}_{X_i^+} = \mathbf{0}_M$ $\overrightarrow{V}_{X_i^+} = \begin{bmatrix} \frac{\lvert\hat{m}_{X_{i,1}}\rvert}{\beta} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\lvert\hat{m}_{X_{i,D}}\rvert}{\beta} \end{bmatrix}$
$\begin{aligned} \overrightarrow{\xi}_{X_i^{\{01\}}} &= \mathbf{0}_M \\ \overrightarrow{W}_{X_i^{\{01\}}} &= \sigma^{-2}_{\mathrm{RP},i} \cdot \mathbf{I}_M\,, \end{aligned}$	$\begin{aligned} \overrightarrow{m}_{X_i^{\{01\}}} &= \mathbf{0}_M \\ \overrightarrow{V}_{X_i^{\{01\}}} &= \sigma^{2}_{\mathrm{RP},i} \cdot \mathbf{I}_M\,, \end{aligned}$
where $\sigma^2_{\mathrm{RP},i} = \frac{\mathrm{Tr}\{\hat{V}_{X_i}\} + \lvert\lvert\hat{m}_{X_i}\rvert\rvert^2}{\beta}$.	

Table A.1: Table summarizing computation of messages generated by One-Hot NUV prior, where all-$\{0,1\}$ solutions are emphasized with a repulsive Log-Cost prior at the origin.



where $d \in \{0, \dots, D\}$

Backward messages through J are calculated as

$$\overleftarrow{\xi}_J = \sum_{d=1}^{D} C_d \cdot \beta$$

$$\overleftarrow{W}_J = \tilde{Z}^{\mathsf{T}} \tilde{Z} + \sum_{d=1}^{D} C_d C_d^{\mathsf{T}} \cdot \frac{\beta |\hat{m}_{J_d}| - 1}{\hat{m}_{J_d}^2},$$

where

$$\tilde{Z} \triangleq \left[\begin{array}{ccccc|cccc|c|c} z_1 & 0 & \dots & 0 & 0 & z_2 & 0 & \dots & 0 & & z_{\mathrm{D}} \\ 0 & z_2 & \dots & 0 & 0 & 0 & z_3 & \dots & 0 & & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & \dots & z_{\mathrm{D\text{-}1}} & 0 & 0 & 0 & \dots & z_{\mathrm{D}} & & 0 \\ 0 & 0 & \dots & 0 & z_{\mathrm{D}} & 0 & 0 & \dots & 0 & & 0 \end{array}\right]$$

$$z_d = y_d - x_d, \quad d \in \{1, \dots, D\},$$

and with all elements of C_d being 0, except for its d-th element being 1.

$\hat{\Sigma}$ can be recovered from estimated $\hat{J} = \hat{m}_J$ as

$$\hat{A}_i^{-1} = \begin{bmatrix} \hat{m}_{J_{i,1}} & \hat{m}_{J_{i,D+1}} & \dots & \hat{m}_{J_{i,D(D+1)/2}} \\ 0 & \hat{m}_{J_{i,2}} & \dots & \hat{m}_{J_{i,D(D+1)/2-1}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{m}_{J_{i,D}} \end{bmatrix}$$

resulting in

$$\hat{\Sigma}_i = \hat{A}_i \hat{A}_i^{\mathsf{T}}$$

$$\hat{\Sigma}_i^{-1} = \left(\hat{A}_i^{-1}\right)^{\mathsf{T}} \hat{A}_i^{-1}, \quad \hat{A}_i \;= \left(\hat{A}_i^{-1}\right)^{-1}$$

Table A.2: Table summarizing computation of messages generated by Normal Node NUV prior as well as the method to recover the estimated noise covariance matrices.

Bibliography

[1] H.-A. Loeliger, "On nup priors and gaussian message passing," in *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, D. Comminiello and M. Scarpiniti, Eds. Piscataway, NJ: IEEE, 2023, Conference Paper, p. 10285859, 33rd IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2023); Conference Location: Rome, Italy; Conference Date: September 17-20, 2023.

[2] ——, "Lecture notes for model-based estimation and signal analysis," 2023.

[3] J. Pearl, "Reverend bayes on inference engines: a distributed hierarchical approach," in *Proceedings of the Second AAAI Conference on Artificial Intelligence*, ser. AAAI'82. AAAI Press, 1982, p. 133–136.

[4] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[5] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[6] H.-A. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn, and N. Zalmai, "On sparsity by nuv-em, gaussian message passing, and kalman smoothing," 2016.

[7] R. Keusch and H.-A. Loeliger, "Half-space and box constraints as nuv priors: First results," 2021.

[8] ——, "A binarizing nuv prior and its use for m-level control and digital-to-analog conversion," 2021.

[9] R. Keusch, "Composite nuv priors and applications," Doctoral Thesis, ETH Zurich, Zurich, 2022.

[10] L. Zhang, "Dirac delta function of matrix argument," *International Journal of Theoretical Physics*, vol. 60, no. 7, p. 2445–2472, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1007/s10773-020-04598-8

List of Figures

List of Tables

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

ON COMPOSITE NUV PRIORS AND HIERARCHICAL MODELS

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
ITEN	LUCA

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
ZURICH, 15.03.24	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.