HELMHOLTZ
ASSOCIATION

Alliance on Systems Biology

HelmholtzZentrum münchen
German Research Center for Environmental Health

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Probabilistic blind source separation for data with network structures

Katrin Illner

January 2015

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl M12 (Mathematische Modellierung biologischer Systeme)

“Probabilistic blind source separation for data with network structures”

Katrin Illner

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:

Univ.-Prof. Claudia Czado, Ph.D.

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. Fabian J. Theis
2. Univ.-Prof. Dr. Claudia Klüppelberg
3. Univ.-Prof. Dr. Achim Tresch, Universität zu Köln
   (nur schriftliche Beurteilung)

Die Dissertation wurde am 28.01.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 22.05.2015 angenommen.

Acknowledgments

A PhD is quite a long and particularly an intense journey. For me, it was bridging from theoretical mathematics that I studied to the new field of systems biology. During my PhD my scientific work continuously opened up and I am really thankful for chances and support at many levels.

First, I want to thank my supervisor Fabian Theis. Fabian, I really enjoyed being part of your research team and experiencing how it was arising during the last years! Thank you for guiding me in my scientific work, for suggesting collaborations and thank you for the many chances I had to visit conferences and become part of the scientific community. I also want to thank you for supporting my interests in scientific writing – I would love to continue working in this field.

Christiane Fuchs was my team leader and mentor throughout my PhD. Thank you deeply, Christiane, for all the discussions, for the guidance and focus, for realistic time-plans, and for new ideas and motivation during that time! I learnt a lot from your detailed comments and corrections in all my manuscripts. Thank you also for carefully proofreading most parts of this thesis.

I further want to thank my third thesis committee member Achim Tresch who attended my thesis committee meetings and gave guidance and ideas for my future work from an outer perspective.

I had a great collaboration with Jari Miettinen, Klaus Nordhausen, and Hannu Oja from Finland. Thanks for the collaborative work, for the discussions and for your comments and suggestions in our joint papers!

Steffen Sass and Nikola Müller helped me with the biological part of this thesis. I had many discussions with detailed explanations about data and experiments, how to prepare the data and how to interpret the results. Thanks a lot for your support from a bioinformatic perspective!

Furthermore, I want to thank all colleagues/roomies/friends from the institute for a warm working athmosphere, for many discussions, for events after work, and for sharing scientific and private thoughts. I very much enjoyed my time with you at the ICB!

And one last thank from the more private side. The closer this thesis came to its end the more I could really feel the importance of family and close friends in this work. Therefore I want to thank the people who supported me whenever working on this thesis was very intense and hard, frustrating or overextending in its bigness. Thank you for listening to me, for putting my mind at rest and also for distracting me from work from time to time. Most of all, thank you deeply for all your presence during the final stage!

Abstract

Blind source separation (BSS) methods have widely been used to identify meaningful signals from observed multivariate mixtures. Applications range from audio recordings to biomedical signals or images, coding theory and large-scale data from molecular biology. If the data has a specific known structure this can be used to perform a more appropriate signal separation. The structure can be the time axis or a grid and in this thesis we more generally assume *data with network structure*. This means that a variable either initiates the signal or depends on one or multiple predecessors. The definition includes time series but beyond that it describes any regulatory system, for example gene interaction networks. For network data based on a non-trivial network only few BSS methods exist so far. Furthermore, fully probabilistic modeling can provide useful additional information about the mixing process and the source signals. Probabilistic models can, for example, provide a more accurate noise estimation, determine the true number of source signals or evaluate estimates based on their distributions. In this thesis we address all these tasks. In the first part, we investigate BSS methods for weakly stationary time series. Here, we consider limiting distributions of the mixing estimates and propose methods to determine the *mixing pattern*. With this, we can decide which source signals actually contribute to a specific observation and in experiments with different experimental conditions we can decide which source signals are task-related. In the second part, we invent a new probabilistic BSS model for network data where we define the source signals in terms of a Bayesian network. This concept has successfully been used to learn and, thus, model gene interaction networks over the last years. To keep the parameter space small, we define weak stationarity for Bayesian networks; with this, we continue an existing analytical BSS method for network data. We infer the model parameters in an expectation-maximization scheme. Due to the flexible modeling using Bayesian networks we can deal with repeated measurements and missing components and we can compare different possible network structures. We illustrate these strengths in simulations and in an application to gene expression data where systemic inflammation in humans is under investigation. With all this, the thesis contributes to the entirety of probabilistic methods for structured data and provides a wide repertoire of applicabilities based on the probabilistic modeling.

Zusammenfassung

Blind source separation (BSS) Methoden finden breite Anwendung, wenn es darum geht, aus mehrdimensionalen Beobachtungen einzelne aussagekräftige Signale oder *Quellen* herauszufiltern. Das zugrunde liegende Modell ist eine lineares Mischmodell. Die Anwendungsgebiete reichen von Audioaufnahmen, biomedizinischen signal- oder bildgebenden Verfahren bis hin zu Kodierungsmethoden und umfangreichen Datensätzen aus der Molekularbiologie. Wenn den Daten eine bekannte sie generierende Struktur zugrunde liegt, wie zum Beispiel die Zeitachse oder ein Gitter, dann kann diese verwendet werden, um eine aussagekräftigere Quellentrennung zu erreichen. In dieser Arbeit konzentrieren wir uns auf Daten mit Netzwerkstruktur und nehmen an, dass eine Variable entweder das Signal initiiert oder von einer oder mehreren Vorgängervariablen abhängt. Diese Definition schließt Zeitreihen ein, aber beschreibt darüber hinaus jedes regulatorische System wie zum Beispiel Interaktionsnetze von Genen. Für Daten basierend auf einem nicht-trivialen Netzwerk gibt es bisher nur wenige BSS Methoden. Zusätzlich zu Strukturanahmen kann die Formulierung als vollständig probabilistisches Modell weitere nützliche Informationen über die Mischung der Signale und die herausgefilterten Quellen liefern. Probabilistische Modelle können zum Beispiel eine exaktere Schätzung für mögliches Hintergrundrauschen liefern, die tatsächliche Anzahl an Quellen bestimmen oder Schätzungen von Parametern mit Hilfe ihrer Verteilung beurteilen. In dieser Arbeit wenden wir uns allen genannten Bereichen zu. Im ersten Teil untersuchen wir BSS Methoden für schwach stationäre Zeitreihen. Wir benutzen Grenzwertverteilungen der Mischmatrix, um das Mischmuster zu identifizieren. Damit können wir zum Beispiel entscheiden, welche Quellsignale tatsächlich aktiv sind oder in Experimenten bedingungsabhängige Quellen finden. Im zweiten Teil stellen wir eine neue probabilistische BSS Methode vor, bei der wir die Quellsignale mit Hilfe eines Bayesianischen Netzes beschreiben. Dieses Konzept wurde in den letzten Jahren erfolgreich verwendet, um genregulatorische Netzwerkstrukturen zu lernen beziehungsweise zu modellieren. Um die Anzahl an Modellparametern gering zu halten, definieren wir schwache Stationarität für Bayesianische Netze. Damit führen wir die Idee einer bereits existierende analytischen BSS Methode für Netzwerkdaten weiter. Parameterinferenz erfolgt in unserem Ansatz mit Hilfe von Erwartungswertmaximierung. Durch die Verwendung Bayesianischer Netze können wir mit Mehrfachmessungen und fehlenden Komponenten umgehen sowie verschiedene Netzwerkstrukturen vergleichen. Wir demonstrieren die Stärke unseres Modells in Simulationen und in einer Anwendung auf Genexpressionsdaten, bei der systemische Entzündungen bei Menschen untersucht wurden. Insgesamt ergänzt diese Arbeit das Spektrum an probabilistischen BSS Methoden für Daten mit bekannter zugrundeliegender Struktur und liefert ein großes Repertoire an Anwendungen, die nur durch probabilistische Modelle möglich sind.

Contents

$\mathbb{R}^{n\times m}$	$\mathbb{R}$-vector space of all real $n \times m$ matrices, page 20
$\mathbb{S}^{n-1}$	$(n-1)$-dimensional sphere, page 20
$\lVert . \rVert_p$	L_p-norm of a vector, page 19
$\lVert . \rVert_{\mathrm{F}}$	Frobenius norm of a matrix, page 20
I_n	identity matrix with rank n, page 20
$Gl(n)$	general linear group of real $n \times n$ matrices, page 20
$O(n)$	orthogonal group of real $n \times n$ matrices, page 20
$\mathrm{svd}(M)$	orthogonalization of M using the singular value decomposition, page 22
P_X	probability function of X, page 22
F_X	(cumulative) distribution function of X, page 23
f_X	densitiy function of X, page 23
$X \perp\!\!\!\perp Y$	X and Y independent, page 24
$X \perp\!\!\!\perp Y \mid Z$	X and Y conditionally independent given Z, page 24
$E[X]$	expectation of X, page 24
$\mathrm{Var}(X)$	variance of X, page 25
$\mathrm{Cov}(X,Y)$	covariance of X and Y, page 25
$\mathrm{Corr}(X,Y)$	Pearson correlation of X and Y, page 25
$\mathcal{U}[a,b]$	uniform distribution, page 26
$\mathcal{N}(\mu,\Sigma)$	normal distribution, page 26
$\mathcal{X}_k^2$	chi-squared distribution, page 27
$\mathrm{diag}(a_1,\ldots,a_n)$	diagonal matrix with entries $a_1,\ldots,a_n$, page 45
$\mathrm{vec}(A)$	columnwise vectorization of the matrix A, page 75
$ASV(\hat{\boldsymbol{A}})$	asymptotic variance of the mixing estimate $\hat{\boldsymbol{A}}$, page 77
$\widehat{ASV}(\hat{\boldsymbol{A}})$	finite-sample variance of the mixing estimate $\hat{\boldsymbol{A}}$, page 77
$\mathcal{M}(G,q)$	source model based on one graph, page 93
$\mathcal{M}(P_1,\ldots,P_q)$	source model based on multiple pathways, page 105

1

Introduction

The only source of
knowledge is experience.

Albert Einstein

1.1 Probabilistic blind source separation

An often used illustration for blind source separation (BSS) is the several speakers problem or cocktail-party problem. When several people talk at the same time and microphones record the overall noise one wants to identify the voice signal of each person from the mixed records (Figure 1.1). The observations are given by the microphone recordings and the unknown sources are the separate voice signals. In the basic BSS model the observations are generated by a *linear* mixing of the source signals. This scaling (or weighting) of the source signals can for example represent different distances between person and microphone. In general, one assumes that the mixing is *instantaneous*, i. e.

the records are without temporal delay. The BSS objective then is to determine an unknown mixing matrix together with unknown source signals from multiple observed recordings. The various existing BSS models differ in additional assumptions on the source signals and/or the mixing matrix.

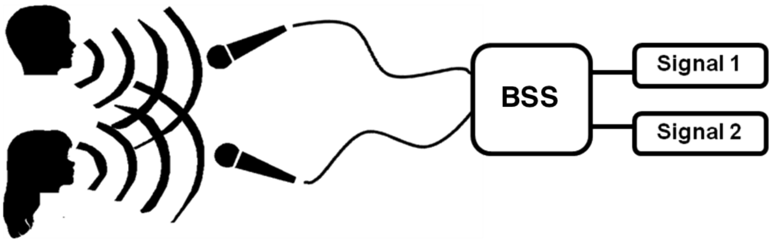

Figure 1.1: Blind source separation, adopted from Eastaway *et al.* (2015). In BSS one assumes multivariate observations (e. g. records from two microphones) and aims to identify the original source signals (e. g. voice signals of two persons). In the basic model we assume that the observations are generated as a linear and instantaneous mixing of the source signals.

Applications of BSS methods not only include audio signals like speech and music (Lee *et al.*, 1999; López *et al.*, 2011); they further range to image processing (Cichocki & Amari, 2002; Zibulevsky, 2003) or coding theory (Lin *et al.*, 2006; Yang *et al.*, 2008). In the latter one is interested in a high-dimensional representation (encryption) of the original signals and thus the number of source signals is larger than the number of observed signals. Furthermore, BSS methods have successfully been applied to biomedical signal and image data. In Joyce *et al.* (2004) the brain's activity is recorded with the electroencephalography (EEG) measuring technique. Using BSS methods artifacts like eye movement or blinking could be separated from the data. Theis *et al.* (2008) performed dimension reduction on high-dimensional functional magnetic resonance imaging data (fMRI) that record the brain's activity. The patients were exposed to a photic stimulus and BSS methods could e. g. identify the component that explains this stimulus. In Theis *et al.* (2010), BSS has further been applied to fluorescence recovery after photobleaching data (FRAP).

In this thesis we focus on large-scale data from molecular biology. Data like gene expression measurements or metabolomic concentrations reflect processes that take place in a cell, a specific tissue or in the whole organism. Usually, several processes are present at the same time and one observes a mixture of these active pathways. Identifying pathways that are related to different experimental conditions or different phenotypes provides meaningful insights. In Lutter *et al.* (2008), for example, independent component analysis (ICA) is applied to gene expression data. From the estimated source signals (i. e. profiles among all genes) the authors selected subsets of genes with highly positive or highly negative values. These sub-modes characterize the respective source signals and one can perform enrichment analysis to relate the signals to pathways from the literature. In a similar approach, Teschendorff *et al.* (2007) identified regulatory modules with different activation patterns dependent on the breast cancer phenotype. Furthermore, Krumsiek *et al.* (2012) applied mean-field ICA to metabolomic data and reconstructed meaningful components of the human blood metabolome.

Including the structure of the data

Time series data, images or data from biological systems have a specific structure. In the first, the time points follow one after the other, in the second, the variables are organized on a grid and in the last the variables are for example given as genes from a gene regulatory network or as metabolomic compounds from a metabolomic pathway. If the structure of the variables is known, it can be used to perform a more appropriate signal separation. In case of time series signals, for example, one wants to identify source signals that contain different temporal information. Here, one commonly assumes that the signals are weakly stationary and one wants to identify source signals that are uncorrelated even when shifted along the time axis. An (un-)mixing estimate can then be derived by (jointly) diagonalizing sample autocovariances at different lags; an estimate of the source signals is given as product of unmixing estimate and observations. BSS methods based on this idea are for example `AMUSE` (Tong *et al.*, 1990), `SOBI` (Belouchrani *et al.*, 1997), and `ACDC` (Yeredor, 2002). The concept has further been generalized to multiple dimensions which makes it applicable to images (Theis *et al.*, 2004b, 2008). Recently, Kowarsch *et al.* (2010) translated the idea of "uncorrelated signals" from time series to networks and provided the (joint) diagonalization algorithm `Grade` for network data. Here, the source signals reflect different signaling information of the network.

More information from probabilistic models

Often, BSS methods exploit descriptive statistics to determine the unknown components of a mixing model. The statistics are used to define a cost function according to the model assumptions; different optimization techniques can then yield an (un-)mixing estimate and with this an estimate of the source signals. Often both steps are approximate – the definition of a cost function and the optimization technique. The independent component model, for example, can be addressed by maximizing the Gaussianity of the signals. A descriptive statistical measure for Gaussianity is the kurtosis and maximization can be achieved by local gradient descent or fixed point approaches (Hyvarinen *et al.*, 2002). We refer to BSS methods that determine an (un-)mixing estimate based on descriptive statistics as *analytical methods*.

Probabilistic methods, in contrast, assume and exploit the (joint) distribution of the random variables and/or the model parameters. The unknown components can, for example, be estimated using a maximum likelihood or Bayesian approach. Such statistically interpretable models bring several benefits – parameter estimates can be evaluated based on their distributions, the (in general unknown) number of source signals can be determined, and hyperparameters or the additional information of parameterpriors can improve the performance. For many BSS models probabilistic versions have been formulated, for example probabilistic PCA (Tipping & Bishop, 1999) and Bayesian ICA (Choudrey & Roberts, 2001). Both define the BSS model as a latent variable model; the former performs parameter inference in an expectation-maximization scheme, the latter uses a variational approach. Furthermore, both approaches use hyperparameters to force single columns of the mixing estimate to zero. With this, the non-zero columns automatically yield the (true) number of source signals; the concept is known as automatic relevance determination. In the main part of this thesis, we focus on BSS methods that provide a probabilistic description of the full model or of parts of the model.

1.2 Probabilistic BSS using graphical models

In this thesis we want to deepen the understanding of separate processes in data with a known network as data-generating structure. Such data can, for example, be gene expression measurements with an underlying gene interaction network. We assume a linear mixing model and explicitly include the structure of the data to perform a more appropriate source separation. With this, we follow the approach of Kowarsch *et al.* (2010) but now provide a fully probabilistic model and go beyond the scope of simple parameter estimation. The benefits of our model are amongst others: We can deal with multiple or missing observed components, we can determine the correct number of source signals using model selection criteria, and we can compare different network structures. Moreover, we can use different networks to describe each source signal individually; this can provide useful insights when possibly active parts of the network are known. Methodically, we combine the task of BSS with the mathematical concept of graphical models. In the following we motivate the choice of graphical models and describe our approach.

In (probabilistic) graphical models the random variables are associated with the nodes of a graph and the dependence between the variables is defined by the graph structure. In the literature, such models have successfully been used to learn and model biological interaction networks. *Gaussian graphical models*, for example, are based on an undirected graph and have been applied to genomic data (Schäfer & Strimmer, 2005) and metabolomic data (Krumsiek *et al.*, 2011). In both studies, an interaction network of the variables was determined using partial correlations. Banerjee *et al.* (2006) continued this concept to learn sparse interaction structures. *Bayesian networks*, in contrast, are based on a directed and acyclic graph. The variables fulfill the Markov property, i. e. a variable given its parents (direct predecessors) is independent of its other predecessors. Friedman *et al.* (2000) learnt Bayesian networks from gene expression measurements to recover gene-interactions. Furthermore, significant subnetworks of interacting genes could be identified in Pe'er *et al.* (2001). We follow these approaches and use a Bayesian network to describe the source signals of our BSS model. In contrast to the above methods, we assume that the network structure is a priori known.

Since Bayesian networks obey the Markov property, the joint distribution of the random variables completely factorizes and the factors are given by the conditional distributions of the variables given their parents. This yields advantages in terms of model complexity and computational costs. To make parameter inference feasible in our model, we assume *(weak) stationarity* of the signals. Weak stationarity is in fact a property of stochastic processes. A stochastic process $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ is weakly stationary if the mean $E[\boldsymbol{s}(t)]$ and the autocovariance

$$\operatorname{Cov}(\boldsymbol{s}(t), \boldsymbol{s}(t-\tau))$$

for any lag $\tau \in \mathbb{Z}$ are independent of the time point $t \in \mathbb{Z}$. Since the autocovariance is identical for any two random variables $\boldsymbol{s}(t)$ and $\boldsymbol{s}(t-\tau)$ we can estimate this quantity from one single observed time series. This is the basis of all joint diagonalization approaches. Kowarsch *et al.* (2010) provided a formulation of (weak) stationarity for networks. Let the variables $\big(\boldsymbol{s}(i)\big)_{i=1}^{N}$ be connected by the edges of a weighted directed graph G with edge weights $\kappa_{ji} \in \mathbb{R}$. In the style of the time-shift $\boldsymbol{s}(t) \to \boldsymbol{s}(t-\tau)$ along the time axis, they introduced a graph-shift $\boldsymbol{s}(i) \to \boldsymbol{s}^G(i)$ along the edges of the graph G (Figure 1.2). Here, $\boldsymbol{s}^G(i) = \sum_{j\in pa(i)} \kappa_{ji}\boldsymbol{s}(j)$ is the (weighted) sum of all parent nodes of $\boldsymbol{s}(i)$ which we index with $pa(i)$. Stationarity of the network means that the graph-delayed covariance

$$\operatorname{Cov}(\boldsymbol{s}(i), \sum_{j\in pa(i)} \kappa_{ji}\boldsymbol{s}(j))$$

is independent of the index i. Thus, each node in the network recieves the same regulatory stimulus from its parent nodes and as before the graph-delayed covariance can be estimated from one set of observations. Since we want to provide a full probabilistic source model we sharpen the above stationarity assumptions and assume that the covariance between any two adjacent nodes is constant up to a known scaling factor given by the edge weight:

$$\operatorname{Cov}(\boldsymbol{s}(i), \boldsymbol{s}(j)) = \frac{1}{\kappa_{ji}|pa(i)|}\, D\ .$$

According to Kowarsch *et al.* (2010), we call D the graph-delayed covariance and we with this we can define all conditional distributions of our Bayesian network. The separation assumption is that D is diagonal, i. e. different source signals are uncorrelated with respect to a shift in the network.

This source model describes the latent variables in our BSS approach. We further introduce observed variables as linear mixtures of the latent variables. At this point, repeated measurements and missing components can be easily included. The model parameters are then given by the graph-delayed covariance, a mean vector, the mixing matrix and the noise variance. To estimate parameters and source signals we provide an expectation-maximization scheme and call the resulting algorithm `emGrade`. With this statistically interpretable model we bring all gains of probabilistic BSS to the field of network data. In Chapters 6 and 7 we illustrate the power of `emGrade` in simulations and in the application to gene expression data.

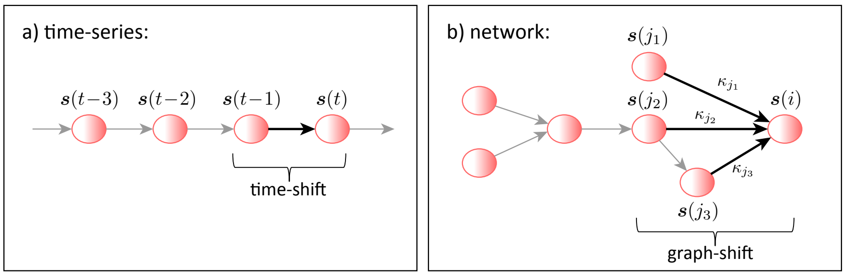

Figure 1.2: Time-shift and graph-shift. In the style of a) the time-shift $s(t) \to \boldsymbol{s}(t-\tau)$ for a node $\boldsymbol{s}(t)$ of a time series, we define b) the graph-shift for the node $s(i)$ in the shown network as $s(i) \to \kappa_{j_1}\boldsymbol{s}(j_1) + \kappa_{j_2}\boldsymbol{s}(j_2) + \kappa_{j_3}\boldsymbol{s}(j_3)$. Thus, the graph-shift of a node is given by the weighted sum of all parents nodes.

1.3 Deep insights on a simple structure

In applications of mixing models we sometimes face the question whether single source signals are present in a specific observation or not. In the several speakers problem, for example, we want to know who is talking into a specific microphone. In experimental scenarios, it can provide new insides if we identify source signals that are “on” in the actual experiment and “off” in a control study. Both questions are related to the shape or pattern of the mixing matrix. In the former example, we want to determine true zero-entries from non-zero estimates. To perform sound decisions we need to know the distributions of the mixing estimates.

The method `emGrade` naturally provides distributions of the source signals due to the expectation maximization scheme. For the parameters, in particular for the mixing matrix, we can only determine approximate confidence intervals. To facilitate sound decisions about the mixing pattern, we consider the joint diagonalization algorithms `SOBIdef` and `SOBIsym` (Miettinen *et al.*, 2014, 2015) for weakly stationary time series. Both algorithms have been published recently – partly in joint work. If we have multiple observations of the same mixing model the (un-)mixing estimates are asymptotically normally distributed under mild conditions. Furthermore, one can explicitly calculate the asymptotic variances of the estimates. Based on these limiting distributions, we invent a family of hypothesis tests to compare linear combinations of the columns of the mixing estimate to a pre-defined vector. With this, we can in particular decide whether mixing columns represent an on/off-shape of the respective source signals. In addition, we consider model selection criteria and define reduced mixing estimates that contain zero-entries – with this we can determine the most appropriate zero-pattern of the true mixing matrix. Thus, we provide methods to deeply analyse the mixing of processes for time-dependent data based on the distributions of the mixing estimates.

1.4 Overview of this thesis

In Chapter 2 we shortly sketch the field of molecular biology. We introduce the central principles in molecular biology, discuss gene regulatory networks and explain measurement techniques that are used to obtain the data in our applications.

In Chapter 3 we provide necessary mathematical basics. We first discuss important matrix decompositions which are used in various BSS approaches. We then introduce stochastic processes and properties relevant for the time series part and we introduce (un-)directed graphical models relevant for the network part. We further explain parameter estimation in probabilistic models with a focus on expectation maximization and we shortly describe model selection criteria.

Chapter 4 gives a review about existing BSS procedures. We introduce three models in more detail – the independent component model, non-negative matrix decomposition, and weakly stationary time series with the method of joint diagonalization of autocovariances. In particular, we review the algorithm `Grade` (Kowarsch *et al.*, 2010) which generalizes concepts from time series analysis to networks; the algorithm is the basis for our probabilistic model in Chapter 6.

In Chapter 5 we begin our survey of probabilistic BSS methods with time series data. We review the algorithms `SOBIdef` (Miettinen *et al.*, 2014) and `SOBIsym` (Miettinen *et al.*, 2015) and explain the limiting distributions of the mixing estimates. To classify the estimation performance we provide an extense comparison to other BSS methods. Based on the limiting distributions we define a family of hypothesis tests and model selection approaches to probabilistically determine the zero-pattern of the true mixing matrix. For validation, we consider matrices with zero and non-zero entries and demonstrate the recovery of the true zero-pattern for different time series models.

Chapter 6 contains the main contribution of this thesis – the probabilistic BSS algorithm `emGrade` for network data. To describe the data we define a source model in terms of a stationary Bayesian network and then expand the network by a linear mixing. We explicitly derive parameter updates for an expectation maximization scheme. As model extension we incorporate repeated observations and missing components and allow different network structures in each source signal. In the last part we evaluate the performance of `emGrade`. We discuss the empirical convergence and compare the algorithm to other BSS methods. Furthermore, we use model selection criteria to determine the true number of source signals and to select the most appropriate structure of the sources in simulations.

In Chapter 7 we apply `emGrade` to publicly available gene expression data. The data consists of a treatment and a control group and we use literature-derived pathways to model the dependencies of the variables. We compare different pathways, investigate the predictive power of our model for missing measurements, and discuss the biological interpretation of the estimated source signals. We further provide a comparison of `emGrade` to other BSS algorithms.

Chapter 8 provides a summary of the main findings and results of this thesis and we propose possible targets for future research.

1.5 Main scientific contributions

The main scientific contributions of this thesis are

- the invention of probabilistic setups to determine the structure of blind source separation mixing matrices (e. g. the position of zero-entries); our approaches are based on the limiting distributions of the algorithms `SOBIdef` and `SOBIsym` for weakly stationary time series,

- a probabilistic description of regulatory data with known interaction structure; here, we define a stationary Bayesian network dependent on a single parameter,
- the invention of the algorithm `emGrade` to separate data with complex network structures; the algorithm combines the task of blind source separation with the mathematical concept of Bayesian networks; we provide explicit formulas for parameter inference using an expectation maximization scheme and introduce extensions of the algorithm assuming repeated observations, missing components and different network structures of each source signal,
- the proof of benefits in the application of `emGrade` to gene expression data; we can identify relevant pathways, predict missing measurement values and provide probe set annotation of single genes on the microarray.

Parts of this thesis are already published in peer-reviewed journals or in conference proceedings. Thus, some chapters are strongly related to and in parts identical with the following publications:

- **K. Illner**, C. Fuchs, F.J. Theis (2014). Bayesian blind source separation for data with network structure. Journal of Computational Biology, 21, 855–865.
- **K. Illner**, C. Fuchs, F.J. Theis (2014). Bayesian blind source separation applied to the lymphocyte pathway. In Proc. 21st International Conference on Computational Statistics (COMPSTAT 2014), 625–632.
- **K. Illner**, C. Fuchs, F.J. Theis (2012). Blind source separation using latent Gaussian graphical models. In Proc. 9th International Workshop on Computational Systems Biology (WCSB 2012), 34–37.
- **K. Illner**, J. Miettinen, C. Fuchs, S. Taskinen, K. Nordhausen, H. Oja, F.J. Theis (2015). Model selection using limiting distributions of second-order blind source separation algorithms. Signal Processing, 113, 95–103.
- J. Miettinen, **K. Illner**, K. Nordhausen, H. Oja, S. Taskinen, F.J. Theis. Separation of uncorrelated stationary time series using autocovariance matrices, *accepted for Journal of Time Series Analysis.*

Whenever a chapter relates to a publication we will explicitly state this at the beginning of the chapter.

2

Molecular biology

In this part we provide useful knowledge from molecular biology and concepts of systems biology. We shortly sketch the mile stones in DNA discovery; we then introduce the transition from DNA information to protein synthesis and discuss transcriptional regulation of genes. The interaction of genes yields the network structures for the blind source separation methods `Grade` (Kowarsch *et al.*, 2010) and `emGrade` in Chapter 6. Since we consider gene expression measurements in our application in Chapter 7 we further introduce microarray technology. More background information about molecular biology can be found in Berg *et al.* (2011) and Alberts *et al.* (2008), concepts from systems biology are described in Walhout *et al.* (2013).

2.1 Basics in molecular biology

Almost the entire genetic information of an organism is memorized in a molecular structure called DNA. The first notice of this capacious molecule was in 1868/69 when Friedrich Miescher performed experiments about leukocytes (Dahm, 2008; Olby, 1974). Since then, pioneering discoveries were reported continuously. Watson & Crick (1953) presented the double helix as model for the molecular structure of the DNA. In 1956, Crick stated for the first time the *dogma of molecular biology* which explains the relationship between DNA, RNA and proteins (Crick, 1970). In April 2003, the Human Genome Project encoded the complete DNA sequence of the human genome (Collins *et al.*, 2003). All these examples account for the today's detailed picture about the molecular basis of life and in the following we spotlight this picture.

The DNA (deoxyribonucleic acid) is a double helix that consists of two strands of nucleotides and is present in almost every cell. The nucleotides are the DNA building blocks; these are complexes of a base (guanine, adenine, thymine or cytosine), a sugar (deoxyribose) and a phosphate group. In eukaryotes, i. e. organisms that contain a nucleus, the DNA is organized on chromosomes which are located in the nucleus. On the basis of the information coded on the DNA cells permanently synthesize proteins in two steps – *transcription* and *translation.* Firstly, DNA is transcribed into single-stranded RNA (ribonucleic acid) by the enzyme RNA polymerase. Different types of RNA exist; messenger RNA (mRNA) directly codes for the protein synthesis and is in a second step translated into functional proteins. Micro RNA (miRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA), in contrast, are non-coding RNAs and regulatorily or enzymatically involved in transcription and translation.

The information about characteristics of an organism is stored in *genes* (Pearson, 2006). Genes are sequences on the DNA that code for proteins or (small and long) non-coding RNAs. The total of all genes is called genome. The human genome, for example, consists of $\sim 25\,000$ protein-coding genes together with numerous genes for non-coding RNAs. The transcriptome further comprises all RNA transcripts and the total of proteins yields the proteome. Figure 2.1 schematically summarizes the relationship between genome, transcriptome and proteome together with the most important regulatory elements.

The complete process of synthesizing functional gene products (i. e. proteins or non-coding RNAs) from the genetic information is called *gene expression.* The spatiotemporal gene expression pattern of a cell reflects its developmental stage, and environmental and pathological conditions. To link genes and gene products to their biological function public ontologies like Gene Ontology (Ashburner *et al.*, 2000) or Kyoto Encyclopedia of Genes and Genomes (Kanehisa *et al.*, 2011) can be used.

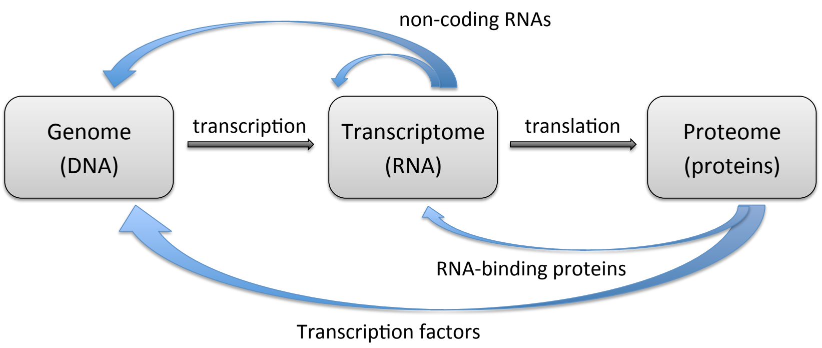

Figure 2.1: Central principle in molecular biology. In the first step DNA is transcribed to mRNA and other non-coding types of RNA. In the second step mRNA is translated to functional proteins. Transcription is mainly regulated by transcription factors and non-coding RNAs; translation, in contrast, is affected by RNA-binding proteins and non-coding RNAs.

2.2 Gene regulation

Cells constantly monitor their environment and react to environmental changes and other external stimuli. Any stimulus triggers a cascade of reactions in the cell. In immunology, for example, the stimuli can be parts of bacteria or bacterial toxin and in response to such antigens the cell generates specific antibody proteins. The most important and effective mechanism in this cellular signaling is the modulation of gene expression. The activation (or inhibition) of specific genes leads in the end to an increased (or decreased) synthesis of functional proteins. In the following we first describe transcriptional gene regulation through transcription factors and then introduce gene regulatory networks as a systematic scheme to summarize gene regulation.

2.2.1 Transcription factors

Transcription factors (TFs) are the key cellular components of gene regulation. They function in activating or inhibiting gene transcription. Some TFs play an important role as master regulators in development, in particular in the formation of organs and tissues (Halder *et al.*, 1995). Other TFs are present throughout lifetime and are, for example, involved in the response to internal or external cellular stimuli. The function of master regulators is in general well-studied. However, the vast majority of TFs have a more subtle effect and their function is not yet completely understood.

Characteristic for TFs is the DNA-binding domain; with this the proteins can bind to the DNA and regulate the expression of their target genes. The counterpart of TFs on the DNA strand are cis-regulatory elements and modules. Both are non-coding DNA sequences with one or multiple TF-binding sites. When TFs bind to these regions the transcription of proximate (upstream) genes is initiated or blocked. One distinguishes between *enhancers* that activate gene expression and *silencers* that inhibit gene expression. If the genes are organized in clusters a single TF can regulate several genes in parallel. Furthermore, a TF can usually bind to multiple DNA sequences and thus affect different genes and a single gene can be regulated by different TFs.

One assumes that 5%-10% of all protein-coding genes of most organisms encode transcription factors (Vaquerizas *et al.*, 2009). However, the total number of TFs in an organism is larger. Firstly, single genes can code for multiple proteins, in particular multiple TFs. The reason for this is alternative splicing. Here, DNA is first transcribed to pre-mRNA; alternative splicing removes exons and, thus, provides different mRNA strands that are used for protein synthesis. Furthermore, TFs can function in a complex with a second TF as homo- or heterodimers; this dimerization yields additional TFs that are not encoded as additional genes (Grove & Walhout, 2008). Another aspect of TFs are post-translational modifications. In response to external stimuli, signaling pathways can in the end lead to phosphorylation and, thus, activation of specific TFs. Many TFs are such functional endpoints of signaling pathways (Yen *et al.*, 2011).

Besides TFs, additional proteins without DNA-binding domain play an important role in gene regulation. In eukaryotes, for example, the DNA is tightly wrapped around histones. Transcription is only possible when this chromosomal structure is released by histone (de-)acetlylases. Further proteins are methylases that release DNA methylation or co-activators/co-repressors that build functional complexes with TFs. Such complexes can amplify a stimulus-induced signal and lead to a more specific cellular response.

2.2.2 Gene regulatory networks

Gene regulatory networks (GRNs) summarize regulatory interactions between TFs and target genes. The nodes of the network are given by TFs and target genes, the directed edges indicate an activating or inhibiting regulatory effect. GRNs can, for example, be inferred from gene expression data where the expression is measured over different experimental conditions or time; respective measurement techniques are introduced in the next section. To infer a GRN, one assumes that genes with similar expression profile are co-regulated by a common TF. The expression profiles of gene set and TFs can be similar or opposite, depending on the regulatory effect of the TFs (Segal *et al.*, 2003). Such large-scale identifications of regulatory interactions have provided GRNs for whole organisms. However, the approach has some drawbacks: The predictions might describe indirect regulations or co-expressions without regulatory consequence. Furthermore, TFs with non-changing expression profile cannot be identified as regulators (Walhout, 2011).

The structure of GRNs provides useful insights into the regulatory mechanisms of biology. Several aspects of the network topology can be investigated. The degree of a node, for example, is the sum of in- or outcoming edges. Like in most real networks, a few TFs function as master regulators, whereas the main portion of TFs only regulates a small set of genes. Hierarchical structures can function as amplification of a stimulus-induced signal; often, TFs on the same layer share static and dynamic properties (Jothi *et al.*, 2009). Furthermore, modularity of the network (i. e. clusters of highly connected genes) can facilitate a fast response to external stimuli. Besides such descriptive network properties, overrepresented network motifs are of special interest. The most important motifs are a) autoregulation, where a TF activates/represses itself, b) cascades with several TFs organized one after the other, c) feedforward loops, where two signals are in parallel and d) feedback loops, where two signals are in parallel but with opposite direction.

2.3 Microarray technology

Several methods have been developed to determine the activity of genes in single cells, cell accumulations or tissues. Technologies either measure the (relative) amount of mRNA or translated proteins. The amount of proteins can, for example, be determined using mass spectrometry or western blots; both methods exploit in the first place the different size or mass of the proteins. In this thesis we concentrate on mRNA measurements. Such data can be collected using microarrays and we describe the technology in the following.

Microarrays determine the relative amount of (known) RNA transcripts in a given sample. The chips consist of numerous spots/probe sets that measure the amount of individual genes or gene transcripts. Each probe set contains copies of a specific DNA sequence which in general matches a short part of the gene. Often, several probe sets for the same gene are embedded; this facilitates a measurement control. To perform microarray analysis, one first synthesizes complementary DNA (cDNA) from the RNA samples by reverse transcription. Thereby, labeled nucleotides are built-in (e. g. fluorophore-labeled nucleotides) and one uses DNA instead of RNA as copy since this is more stable. The cDNA samples are then hybridized with the DNA sequences on the chip and lasers determine the fluorescence intensity.

One distinguishes between one- and two-channel arrays. One-channel arrays determine relative abundances of transcripts from one sample. Two-channel arrays, in contrast, perform competitive hybridization of two samples on the same chip. The samples are marked with different labels (e. g. red and green fluorophores) and represent, for example, treatment vs. control or cancer vs. normal cells. In general, the intensity data is used to perform a ratio-based analysis. With this one can determine genes that are up- or down-regulated between the two samples.

3

Mathematical basics

In this chapter we provide the mathematical background relevant for the contents of this thesis. Among matrix decompositions and basic definitions from probability theory, we introduce the important class of MA(∞)-processes which are the basis for the limiting distributions of mixing matrices in Chapter 5. We further introduce graphical models, i. e. models that are defined with respect to a network structure. Using graphical models we develop the blind source separation method `emGrade` in Chapter 6. Since the parameters will be estimated using expectation maximization, we further introduce parameter inference in probabilistic models and focus on expectation maximization. The chapter concludes with the basics in model selection.

3.1 Vectors, matrices and decompositions

In this part we shortly outline basic properties of matrices and discuss two matrix decompositions and orthogonalization of matrices. The following is especially relevant for the blind source separation approaches in Chapters 4 and 5. Detailed mathematical background can be found in Fischer (2013).

3.1.1 Notations and properties of matrices

Let $\mathbb{R}^n$ denote the standard n-dimensional $\mathbb{R}$-vector space. The p-norm or (L_p-norm) of a vector $x \in \mathbb{R}^n$ is defined as $\|x\|_p = (|x_1|^p + \ldots + |x_n|^p)^{1/p}$. For $p = 2$ we get the Euclidean norm. With this the $(n-1)$-dimensional sphere is given by the subspace $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n \colon \|x\|_2 = 1\}$. Let further $\mathbb{R}^{n \times m}$ denote the $\mathbb{R}$-vector space of all $(n \times m)$-dimensional real matrices. For matrices we consider the Frobenius norm defined as $\|A\|_{\mathrm{F}} = (\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2)^{1/2}$. A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if $A = A'$ where A' denotes the transpose of A. A is orthogonal if $AA' = I_n$ where I_n is the identity matrix. Furthermore, a matrix A is positive definite if and only if $x'Ax > 0$ for all $x \in \mathbb{R}^n$ and A is positive semi-definite if and only if $x'Ax \geq 0$ for all $x \in \mathbb{R}^n$. The general linear group $Gl(n)$ consists of all invertible $n \times n$ matrices and with $\mathcal{O}(n)$ we denote the sub-group of orthogonal matrices.

3.1.2 Eigenvalue decomposition

Let $A \in \mathbb{R}^{n \times n}$ be a quadratic $n \times n$ matrix. A real number $d \in \mathbb{R}$ is an eigenvalue with $v \in \mathbb{R}^n$ a corresponding eigenvector of A if and only if

$$Av = dv \ .$$

If the matrix A is positive definite then all eigenvalues are strictly larger than zero. If A is positive semi-definite then all eigenvalues are larger than or equal to zero. A $n \times n$ matrix can have up to n different eigenvalues, each corresponds to an at least 1-dimensional subspace of eigenvectors. Eigenspaces of different eigenvalues are orthogonal to each other. If eigenvectors of A build a basis of the n-dimensional space $\mathbb{R}^n$ then an eigenvalue decomposition of A exists and is given by

$$A = VDV' \ .$$

Here, $V \in \mathcal{O}(n)$ is an orthogonal matrix, and D is a diagonal matrix containing the eigenvalues of A on its main diagonal. The columns of V are corresponding eigenvectors. If A has n different eigenvalues then the eigenspaces are 1-dimensional. Since the columns of the orthogonal matrix V are normalized, the decomposition VDV' is uniquely determined up to sign and permutation of the columns of V. This uniqueness is important when discussing indeterminacies of blind source separation approaches for weakly stationary time series (Section 4.3.1).

The above decomposition is also known as diagonalization of A. It exists in particular for symmetric matrices. A well-known numerical method to derive the eigenvalues of a matrix is the Jacobi technique. The matrix A is gradually transformed to D using Givens rotations. With these rotations one sets the off-diagonal element with highest absolute value of the current matrix to zero. For the blind source separation algorithm `SOBI` (Belouchrani *et al.*, 1997) this technique is extended to a set of matrices that are jointly diagonalized (Section 4.3.2).

3.1.3 Singular value decomposition

Let now $M \in \mathbb{R}^{n \times m}$ be an arbitrary matrix with rank $r \leq n, m$. The singular value decomposition is given by

$$M = U \Sigma V' \ ,$$

where $U \in \mathcal{O}(n)$ and $V \in \mathcal{O}(m)$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{n \times m}$ has entries $\psi_1 \geq \ldots \geq \psi_r > 0$ on its main diagonal and is zero elsewhere. These entries are the singular values of M. The singular values of a matrix M are related to the eigenvalues of the (symmetric) matrix MM'. The eigenvalue decomposition of MM' can be derived as

$$MM' = (U \Sigma V)(V' \Sigma U) = U \Sigma \Sigma' U' \ .$$

Thus, if the singular values of M are given by $\psi_1, \ldots, \psi_r$ then the eigenvalues of MM' are given by $\psi_1^2, \ldots, \psi_r^2$, and 0 if $r < n$. Moreover, if $M \in \mathbb{R}^{n \times n}$ is symmetric then the eigenvalue decomposition is given by $M = U \Sigma_0 U'$ where Σ_0 contains the eigenvalues $|\psi_1|, \ldots, |\psi_r|$, and possibly 0 on its main diagonal. The reason for this relation is, that if v is an eigenvector of M with eigenvalue d, then v is also an eigenvector of M^2 with eigenvalue d^2.

3.1.4 Orthogonalization of matrices

In blind source separation applications the matrix space is often restricted to orthogonal matrices and we need techniques to transform $A \in Gl(n)$ to an orthogonal matrix. A well-known method is the Gram-Schmidt process. Let a_j denote the jth columns of A. For $j = 1, \ldots, n$ we determine $\tilde{a}_j = (I_n - \sum_{r=1}^{j-1} \tilde{a}_r \tilde{a}_r') a_j$ and normalize $\tilde{a}_j$ to unit variance. The matrix $\tilde{A}$ is then orthogonal.

On the other hand the singular value decomposition has an interesting property in terms of orthogonalizing matrices. If $U\Sigma V'$ is the singular value decomposition of $M \in \mathbb{R}^{n\times n}$ then UV' is the orthogonal matrix closest to M in terms of the Frobenius norm. For quadratic M with singular values $\psi_1, \ldots, \psi_r$ it holds $\|M\|_\mathrm{F} = \sqrt{\psi_1^2 + \ldots + \psi_r^2}$. In Chapter 5 we use the notation $\mathrm{svd}(M) = UV'$ to denote orthogonalization of a matrix using its singular value decomposition.

3.2 Basics in probability theory

In the following we give a short introduction to probability theory and provide the basis for the remaining part of this chapter. Further details can for example be found in Grimmett & Stirzaker (2001).

3.2.1 Distribution and density function

Let $(\Omega, \mathcal{F}, P)$ be a probability space consisting of a non-empty set Ω, a σ-algebra $\mathcal{F} \subseteq \mathcal{P}(\Omega)$, and a probability measure $P\colon \mathcal{F} \to [0,1]$. Let further $(E, \mathcal{E})$ be a measurable space. A *random variable* is then a measurable function $X\colon \Omega \to E$, i. e. for $A \in \mathcal{E}$ it holds $X^{-1}(A) \in \mathcal{F}$. Here, E is the state space of the random variable and if E has dimension higher than 1 X is also called *random vector*. In the following we use lower case letters to denote realizations $x = X(\omega)$ for $\omega \in \Omega$.

A random variable $X\colon \Omega \to E$ on $(\Omega, \mathcal{F}, P)$ naturally yields a probability measure P_X on $(E, \mathcal{E})$. For all $A \in \mathcal{E}$ one defines

$$P_X(A) = P(X^{-1}(A)) \ .$$

P_X is called *probability function of* X and we write $X \sim P_X$. Let now $X_1, \ldots, X_n$ ($n \in \mathbb{N}$) be a finite sequence of random variables with values in E. We define the joint (product) random variable $X = X_1 \otimes \ldots \otimes X_n$ as

$$\begin{aligned} X_1 \otimes \ldots \otimes X_n \colon \Omega &\longrightarrow E \times \ldots \times E \\ \omega &\longmapsto (X_1(\omega), \ldots, X_n(\omega)) \end{aligned}$$

We also use the notation $X = (X_1, \ldots, X_n)$ and the corresponding distribution P_X is the joint distribution of $X_1, \ldots, X_n$.

From now on we consider *continuous* random variables with $(E, \mathcal{E}) = (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ and $\mathcal{B}(\mathbb{R}^n)$ the Borel σ-algebra.

Definition 3.1 (Distribution function)**.** Let $X\colon \Omega \to \mathbb{R}^n$ be a continuous random variable on $(\Omega, \mathcal{F}, P)$. The function

$$\begin{aligned} F_X\colon \mathbb{R}^n &\longrightarrow [0,1] \\ (x_1, \ldots, x_n)' &\longmapsto P_X((-\infty, x_1] \times \ldots \times (-\infty, x_n]) \end{aligned}$$

is the distribution function of X with respect to P. F_X is also called the cumulative distribution function (cdf) of X.

Definition 3.2 (Density function)**.** Let $X\colon \Omega \to \mathbb{R}^n$ be a continuous random variable on $(\Omega, \mathcal{F}, P)$ with distribution function $F_X\colon \mathbb{R}^n \to [0,1]$. If $f\colon \mathbb{R}^n \to [0, \infty)$ is a non-negative Lebesque-integrable function such that

$$F_X(x_1, \ldots, x_n) = \int_{-\infty}^{x_1} \ldots \int_{-\infty}^{x_n} f(u_1, \ldots, u_n)\, \mathrm{d}u_1 \ldots \mathrm{d}u_n$$

for all $(x_1, \ldots, x_n) \in \mathbb{R}^n$, then f is a density function with respect to P. We also call $f = f_X$ a probability density function (pdf) of X.

Let $\tilde{X} = (X_1, \ldots, X_n)\colon \Omega \to \mathbb{R}^n$ be a joint random variable and let $f_{\tilde{X}}$ be a density function. For $X = (X_1, \ldots, X_k)$ and $x \in \mathbb{R}^k$ with $k < n$ the *marginal density function* of X is given by

$$f_X(x) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} f_{\tilde{X}}(x, u_{k+1}, \ldots, u_n)\, \mathrm{d}u_n \ldots \mathrm{d}u_{k+1} \ ,$$

i. e. one integrates over all random variables not assigned to X. The above definition also holds for arbitrary subsets $\{n_1, \ldots, n_k\} \subseteq \{1, \ldots, n\}$ and $X = (X_{n_1}, \ldots, X_{n_k})$.

3.2.2 Conditional distributions and independence

Let now X and Y be continuous random variables with state spaces $\mathbb{R}^n$ and $\mathbb{R}^m$. Let f_X and f_Y be the marginal density functions and $f_{X,Y}$ the joint density function. The *conditional density function* of X given Y is then defined by

$$f_{X|Y}(x \mid Y = y) = \frac{f_X(x)}{f_Y(y)} \ ,$$

for all y with $f_Y(y) > 0$. We also write $f_{X|Y}(x \mid y)$ to keep the notation simple. The variables X and Y are *independent* if and only if the joint density factorizes, i. e.

$$f_{X,Y}(x, y) = f_X(x)\, f_Y(y) \ .$$

For independent variables we use the notation $X \perp\!\!\!\perp Y$. Independent random variables with the same distribution function are called i.i.d. (independent and identically distributed). The definition of independence can be extended to conditional independence. Let therefore Z be an additional random variable with state space $\mathbb{R}^k$ and marginal density function f_Z. The random variables X and Y are *conditionally independent* given Z if

$$f_{X,Y|Z}(x, y \mid Z = z) = f_X(x \mid Z = z)\, f_Y(y \mid Z = z)$$

for all z with $f_Z(z) > 0$. For conditional independence we write $X \perp\!\!\!\perp Y \mid Z$.

3.2.3 Expectation and higher-order moments

Let X be a continuous (univariate) random variable with density function f_X. The expectation of X exists if and only if $\int_{-\infty}^{\infty} |x|\, f_X(x)\, \mathrm{d}x < \infty$ and is then defined as

$$E[X] = \int_{-\infty}^{\infty} x f_X(x)\, \mathrm{d}x \ .$$

We also write $X \in \mathscr{L}^1$ if the expectation of X exists. Let now $X_1, \ldots, X_k$ be continuous (univariate) random variables with joint density function f. For a measurable function $g\colon \mathbb{R} \times \ldots \times \mathbb{R} \to \mathbb{R}^k$ we have

$$E[g(X_1, \ldots, X_k)] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(x_1, \ldots, x_k) f(x_1, \ldots, x_k)\, \mathrm{d}x_1 \ldots \mathrm{d}x_k \ .$$

Thus, the expectation $E[\,.\,]$ is a linear operator in the sense that $E[aX + Y] = aE[X] + E[Y]$ for $a \in \mathbb{R}$. The conditional expectation of X given Y is further defined by

$$E[X \mid Y = y] = \int_{-\infty}^{\infty} x f_{X|Y}(x \mid Y = y)\, \mathrm{d}x \ .$$

In Chapter 6 we also use the notation $E_{X|Y}[\,.\,]$ to denote conditional expectations.

Beside the expectation, also higher-order moments are used to describe random variables. For X with $X^r \in \mathscr{L}^1$ the nth central moment is given by $E[(X - E[X])^n]$ and the nth raw moment by $E[X^n]$. We also write $X \in \mathscr{L}^n$ and it holds $\mathscr{L}^n \subset \mathscr{L}^m$ for $m < n$. The variance of $X \in \mathscr{L}^2$ is the second central moment, i. e.

$$\operatorname{Var}(X) = E[(X - E[X])^2] \; .$$

The covariance generalizes the idea of a variance. For variables $X, Y \in \mathscr{L}^2$ the covariance is defined by

$$\begin{aligned} \operatorname{Cov}(X, Y) &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY] - E[X]E[Y] \; . \end{aligned}$$

If X and Y are independent then the covariance is zero. The Pearson correlation of the dependence between two random variables and is defined by

$$\operatorname{Corr}(X, Y) = \frac{\operatorname{Cov}(X, Y)}{\sqrt{\operatorname{Var}(X)}\sqrt{\operatorname{Var}(Y)}} \; ,$$

for $\operatorname{Var}(X), \operatorname{Var}(Y) \neq 0$. The third- and fourth-order central moments normalized with the factor $1/\sqrt{\operatorname{Var}(X)^n}$ for $n = 3, 4$ are known as skewness and flatness. The latter can be used to measure Gaussianity of a probability distribution. Gaussianity means the closeness of a distribution to the Gaussian distribution and one defines the kurtosis of a (zero-mean) random variable $X \in \mathscr{L}^4$ as

$$\operatorname{kurt}(X) = E[X]^4 - 3\,E[X^2]^2 \; .$$

If X is Gaussian then the kurtosis is zero. Thus, this quantity provides a measure of Gaussianity. We come back to this definition in Section 4.2 where we discuss independent component analysis.

3.2.4 Important distributions

In this part we explicitly state the density functions of some important distributions. For the following we define the *indicator function* $\mathbb{1}_B$ on a non-empty measureable set $B \subset \mathbb{R}^n$ as

$$\begin{aligned} \mathbb{1}_B \colon \mathbb{R}^n &\to \mathbb{R} \\ x &\mapsto \begin{cases} 1, & \textit{if } x \in B \; , \\ 0, & \textit{otherwise} \; . \end{cases} \end{aligned}$$

Example 3.1 (Uniform distribution). A random variable $X\colon \Omega \to \mathbb{R}^n$ is *uniformly distributed on* $B \subset \mathbb{R}^n$ if the density function for $x \in \mathbb{R}^n$ exists and is of the form

$$f_X(x) = \frac{1}{\lambda^n(B)} \mathbb{1}_B(x) \ .$$

Here, λ^n is the n-dimensional Lebesgue measure and $\mathbb{1}_B$ is as defined above. We write $X \sim \mathcal{U}(B)$. For univariate X uniformly distributed on an interval $[a, b] \subset \mathbb{R}$ we directly write $X \sim \mathcal{U}[a, b]$.

Example 3.2 (Normal distribution). A random variable $X\colon \Omega \to \mathbb{R}^n$ is multivariate *normally distributed* (or *Gaussian*) if the density function exists for $x \in \mathbb{R}^n$ and is of the form

$$f_X(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\Big(-\frac{1}{2}(x-\mu)' \, \Sigma^{-1}(x-\mu)\Big) \ ,$$

where $\mu \in \mathbb{R}^n$ is the mean vector and $\Sigma \in \mathbb{R}^{n \times n}$ is the symmetric positive semidefinite covariance matrix. We also write $X \sim \mathcal{N}(\mu, \Sigma)$.

Let now $X = \left(\begin{smallmatrix} X_1 \\ X_2 \end{smallmatrix}\right)$, $\mu = \left(\begin{smallmatrix} \mu_1 \\ \mu_2 \end{smallmatrix}\right)$ and $\Sigma = \left(\begin{smallmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{smallmatrix}\right)$. Then $X_1 \mid X_2$ is again normally distributed with mean and covariance given by

$$\begin{aligned} \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(X_2 - \mu_2) \ , \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \ . \end{aligned}$$

If $X\colon \Omega \to \mathbb{R}$ is univariate normally distributed then the density function simplifies to

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(x-\mu)^2}{2\sigma^2}\Big) \ ,$$

for $x \in \mathbb{R}$ and we call $\sigma > 0$ the standard deviation. In this case we have $X \sim \mathcal{N}(\mu, \sigma^2)$. If the natural logarithm of X, i. e. $\ln(X)$, is normally distributed then X is lognormally distributed which we define in the following.

Example 3.3 (Lognormal distribution). A univariate random variable $X\colon \Omega \to \mathbb{R}$ is *lognormally distributed* if the density function exists for $x \in \mathbb{R}$ and is of the form

$$f_X(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(\ln(x)-\mu)^2}{2\sigma^2}\Big) \, \mathbb{1}_{(0,\infty)}(x) \ ,$$

where $\mu \in \mathbb{R}$, $\sigma > 0$ and $\mathbb{1}_{(0,\infty)}$ is the indicator function on $(0, \infty) \subset \mathbb{R}$. We write $X \sim \mathcal{LN}(\mu, \sigma^2)$.

Example 3.4 (Gamma distribution). A univariate random variable is *gamma distributed* if the density function exists for $x \in \mathbb{R}$ and is of the form

$$f_X(x) = \frac{1}{\lambda^k \Gamma(k)}\, x^{k-1} \exp\Big(-\frac{x}{\lambda}\Big)\ \mathbb{1}_{(0,\infty)}(x)\ ,$$

with shape parameter $k > 0$ and scale parameter $\lambda > 0$. The gamma function $\Gamma(\,.\,)$ is defined by $\Gamma(k) = \int_0^\infty t^{k-1} \exp(-t)\,\mathrm{d}t$ and we write $X \sim \Gamma(k, \lambda)$.

Example 3.5 (Chi-squared distribution). A univariate random variable is *chi-squared distributed* if the density function exists for $x \in \mathbb{R}$ and is of the form

$$f_X(x) = \frac{1}{2^{k/2}\Gamma(k/2)}\, x^{k/2-1} \exp\Big(-\frac{x}{2}\Big)\ \mathbb{1}_{[0,\infty)}(x)\ .$$

Here, $k \in \mathbb{N} \setminus \{0\}$ is a positive integer and denotes the degrees of freedom and gamma function $\Gamma(\,.\,)$ like before. We write $X \sim \mathcal{X}_k^2$.

The chi-squared distribution is a special case of the gamma distribution. If X is chi-squared distributed with k degrees of freedom then X is gamma distributed with shape parameter $k/2$ and scale parameter 2. We come back to this distribution when defining test statistics in Chapter 5.

3.3 Stochastic processes

In this part we define stochastic processes and introduce some important time series models. We get back to these models in Chapter 5 where we derive limiting distributions of blind source separation mixing matrices. A detailed discussion of time series is given in Brockwell & Davis (2009).

A *stochastic process* is a family of random variables $\{X_t\}_{t\in T}$ defined on a the same probability space $(\Omega, \mathcal{F}, P)$ and with arbitrary index set T. A realization of the process is given as $\{x_t\}_{t\in T}$ where $x_t = X_t(\omega_t)$ for some $\omega_t \in \Omega$. Let now $\{X_t\}_{t\in T}$ be a stochastic process with $\mathrm{Var}(X_t) < \infty$ for all $t \in T$. The *autocovariance* function $\gamma_X(.\,,.)$ of the process is defined by

$$\gamma_X(s,t) = \mathrm{Cov}(X_s, X_t) = E[(X_s - E[X_s])(X_t - E[X_t])]\ ,$$

for all $s, t \in T$. In time series analysis one usually assumes $\mathbb{Z}$ or $\mathbb{R}$ (or subsets) as index set. Since any observed time series consists of countably many time points we restrict ourselves to $T = \mathbb{Z}$ from now on.

Definition 3.3 (Weak stationarity)**.** A time series $\{X_t\}_{t\in\mathbb{Z}}$ is weakly stationary if for all $s, t \in \mathbb{Z}$ it holds

(i) $E[(X_t)^2] < \infty$

(ii) $E[X_t] = \mu$

(iii) $\operatorname{Cov}(X_t, X_{t+\tau}) = \operatorname{Cov}(X_s, X_{s+\tau})$, for all $\tau \in \mathbb{Z}$

Thus, for a weakly stationary process the autocovariance only depends on the lag $\tau \in \mathbb{Z}$. A weakly stationary process is also known as wide-sense stationary. In contrast, a process is *strongly stationary* if the joint distributions of $(X_{t_1}, \dots, X_{t_k})'$ and $(X_{t_1+\tau}, \dots, X_{t_k+\tau})'$ are identical for all $t_1, \dots, t_k, \tau \in \mathbb{Z}$.

3.3.1 Time series models

In this part we introduce time series models that will be used to generate the data in Chapter 5. We formulate all models as univariate stochastic processes but the generalization to multiple dimensions is straight-forward. Firstly, let $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ be a stochastic process with $E[\varepsilon_t] = 0$ and autocovariance at lag $\tau \in \mathbb{Z}$ is given by

$$\operatorname{Cov}(\varepsilon_t, \varepsilon_{t+\tau}) = \begin{cases} \sigma^2 & \text{for } \tau = 0 \ , \\ 0 & \text{for } \tau \neq 0 \ . \end{cases}$$

The process is weakly stationary and is known as *white noise.* If ε_t is further Gaussian distributed we refer to the process as white Gaussian noise. With this, we define three important classes of time series models.

Example 3.6 (MA(∞)-process). The process $\{X_t\}_{t\in\mathbb{Z}}$ is a moving-average process if

$$X_t = \sum_{j=1}^{\infty} \psi_j \varepsilon_{t-j} + \varepsilon_t \ ,$$

with coefficients $\psi_j \in \mathbb{R}$ and $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ a white noise process.

If $\psi_j = 0$ for $j > q$ the process is a moving-average process or order q and we use the notation MA(q)-process. Let for the coefficients hold $\sum_{j=0}^{\infty} |\psi_j| < \infty$ with $\psi_0 = 1$. The expectation is then given by $E[X_t] = 0$ and the autocovariance at lag $\tau \in \mathbb{Z}$ is of the form $\sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+\tau}$. Since $E[\varepsilon_t]$ and $\mathrm{Cov}(\varepsilon_t, \varepsilon_{t+\tau})$ are independent of the index $t \in \mathbb{Z}$ the process is weakly stationary. MA(∞)-processes represent a wide range of time series models; every zero-mean weakly stationary process that is non-deterministic can be uniquely expressed as a sum of an MA(∞)-process and a deterministic part. This result is known as Wold's decomposition (Brockwell & Davis, 2009, p.187). In particular, AR- and ARMA-processes, which we introduce in the following, can be solely expressed as MA(∞)-processes with no deterministic component.

Example 3.7 (AR(p)-process). The process $\{X_t\}_{t\in\mathbb{Z}}$ is autoregressive of order p if

$$X_t = \sum_{j=1}^{p} \varphi_j X_{t-j} + \varepsilon_t \ ,$$

with coefficients $\varphi_1, \ldots, \varphi_p \in \mathbb{R}$ and $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ a white noise process.

The coefficients ψ_j can be chosen such that the process is weakly stationary; details are provided in Brockwell & Davis (2009). Finally, ARMA-processes combine autoregressive and moving-average processes and are defined as follows.

Example 3.8 (ARMA(p, q)-process). The process $\{X_t\}_{t\in\mathbb{Z}}$ is an autoregressive moving-average process if

$$X_t = \sum_{j=1}^{p} \varphi_j X_{t-j} + \sum_{i=1}^{q} \psi_i \varepsilon_{t-i} + \varepsilon_t \ ,$$

with coefficients $\varphi_j, \psi_i \in \mathbb{R}$ and $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ a white noise process.

3.4 Probabilistic graphical models

In probabilistic graphical models the random variables are represented by the nodes of a graph – the edges of the graph reflect dependencies and independencies between the variables. One distinguishes between undirected and directed models. For both, Markov properties of the random variables are defined differently. In the following we introduce these concepts. A detailed discussion on graphical models with proofs of all statements is given in Lauritzen (1996). A review on applications of graphical models in systems biology can be found in Friedman (2004).

3.4.1 Graphs and notations

We first define undirected and directed graphs and discuss some terminologies. $G^u = (V, E^u)$ is an *undirected* graph if V is a set of nodes and $E^u \subseteq \{\{i, j\} \mid i, j \in V\}$ is a set of undirected edges. The neighbours $\mathrm{ne}(i)$ of a node i are given by all nodes connected with i. Let now $A, B, C \subset V$ be disjoint subsets. C separates A from B if all paths from a node A to a node B contain one node in C. For a subset $A \subseteq V$ the induced subgraph of A is given by the nodes in A and all edges from the original graph that connect nodes in A. A (sub-)graph is complete if all nodes are joined by an edge. If any two nodes of a (sub-)graph are complete by an edge then the graph is a clique. We also use the terms complete and clique for subsets $A \subseteq V$ that induce such subgraphs. Figure 3.1a illustrates the above terminologies for undirected graphs.

A *directed* graph $G^d = (V, E^d)$, in contrast, is given by a set of nodes V and a set of directed edges $E^d \subseteq V \times V$. Since we mainly focus on directed graphs in this thesis we omit superscripts and only write G and E whenever the graph is directed. The parents $\mathrm{pa}(i)$ of a node i are given by all nodes with an edge pointing towards i. If $\mathrm{pa}(i) = \emptyset$ then i is called a root node of the graph. The anchestors $\mathrm{an}(i)$ are all nodes with a directed path leading to i (and with no path backwards). Conversely, all paths from node i are leading towards the descendants $\mathrm{de}(i)$ of i. The non-descendants are given by $\mathrm{nd}(i) = V \setminus (\mathrm{de(i)} \cup \{i\})$ and a set $A \subseteq V$ is anchestral if $\mathrm{an}(a) \subseteq A$ for all $a \in A$. The graph G is acyclic if it contains no directed cycles. In an acyclic graph the nodes can be ordered such that $V = \{1, \ldots, N\}$ and for each node all parent nodes are predecessors with respect to that ordering. Finally, the moral graph G^m is the undirected equivalent of G; it arises from G by adding edges between parent nodes and deleting directions. The terminologies for directed graphs are shown in Figure 3.1b.

3.4.2 Markov random fields

Let now $G^u = (V, E^u)$ be an undirected graph and let $(X_i)_{i \in V}$ be random variables indexed according to the nodes V and with values in $(\mathcal{X}_i)_{i \in V}$. Thus, each variable X_i is

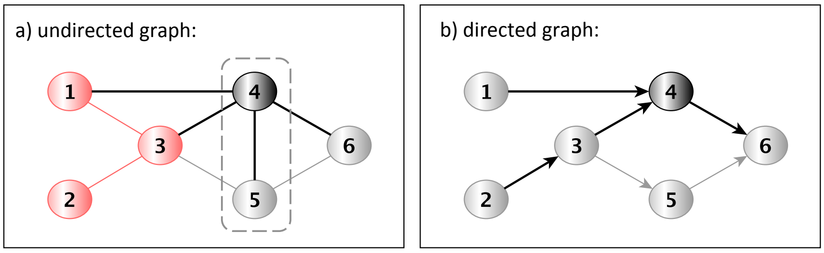

Figure 3.1: Terminologies of graphs. a) shows an undirected complete graph. The neighbours of node 4 are given by $\mathrm{ne}(4) = \{1, 3, 6\}$, the subset $C = \{4, 5\}$ separates $A = \{1, 2, 3\}$ from $B = \{6\}$ and the induced subgraph of A is shown in red. b) shows a directed acyclic graph. The root nodes are given by nodes 1 and 2. The parents of node 4 are given by $\mathrm{pa}(4) = \{1, 3\}$, the anchesters are given by $\mathrm{an}(4) = \{1, 2, 3\}$, and $\mathrm{de}(4) = \{6\}$ are the descendants. The non-descendants of node 4 are $\mathrm{ne}(4) = \{1, 3, 6\}$. Furthermore, the graph in a) is the moral graph of the graph in b).

associated with a node $i \in V$. For a subset $A \subseteq V$ let $X_A = (X_a)_{a \in A}$ denote the joint random variable with values in $\mathcal{X}_A = \times_{a \in A} \mathcal{X}_a$. For such an undirected graphical model various Markov properties are defined.

The variables $(X_i)_{i \in V}$ have the *global Markov property* with respect to G^u if

$$\text{(G)} \qquad X_A \perp\!\!\!\perp X_B \mid X_C \ ,$$

where $A, B, C \subset V$ are disjoint subsets and C separates A from B. This means, that X_A and X_B are conditionally independent given X_C (Section 3.2.2). The random variables X have the *local Markov property* with respect to G^u if for all $i \in V$ one has

$$\text{(L)} \qquad X_i \perp\!\!\!\perp X_{V \setminus (\mathrm{ne}(i) \cup \{i\})} \mid X_{\mathrm{ne}(i)} \ ,$$

where $\mathrm{ne}(i)$ is the set of neighbors of node i. And last, the variables have the *pairwise Markov property* with respect to G^u if for any pair (i, j) of non-adjacent nodes it holds

$$\text{(P)} \qquad X_i \perp\!\!\!\perp X_j \mid X_{V \setminus \{i,j\}} \ .$$

Thus, two random variables – with the respective nodes non-adjacent – are conditionally independent given all other variables. For any distribution of the random variables the global Markov property implies the local property and this further implies the pairwise property, i. e. (G) ⇒ (L) ⇒ (P).

Let now P_X be the probability distribution of $X = X_V$. P_X *factorizes* according to the undirected graph G^u if for all complete subsets $A \subseteq V$ there exist non-negative functions $\psi_A \colon \mathcal{X}_A \to [0, \infty)$ together with a product measure $\mu = \otimes_{i \in V} \mu_i$ on $\mathcal{X} = \mathcal{X}_V$ such that X has density f with respect to μ where

$$f(x) = \prod_{A \text{ complete}} \psi_A(x_A) ,$$

for all $x = (x_i)_{i \in V}$ in $\mathcal{X}$ and $x_A = (x_a)_{a \in A}$. The functions ψ_A are not uniquely defined (there is, for example, arbitrariness in the choice of μ) and one can further restrict the connected subsets to cliques

$$f(x) = \prod_{C \text{ clique}} \psi_C(x_C) .$$

We call a probability distribution that factorizes according to an undirected graph a *Markov random field.* One can show, that the factorization implies the global Markov property (and with this the local and pairwise Markov property). The following theorem provides the opposite implication and can be found in Lauritzen (1996, p.36).

Theorem 3.1 (Hammersley and Clifford)**.** *A probability distribution P_X with positive and continuous density f with respect to some product measure μ satisfies the pairwise Markov property with respect to an undirected graph G if and only if it factorizes according to G^u.*

Markov random fields show an interesting applicability if we assume Gaussian random variables. Let therefore $X = (X_i)_{i \in V}$ be a continuous multivariate variable that is distributed as $\mathcal{N}_{|V|}(\mu, \Sigma)$. Let the covariance matrix Σ be regular, i. e. the inverse exists. In this case, conditional independence in form of the pairwise Markov property can be easily observed. The following corollary is according to Lauritzen (1996, p.129).

Corollary 3.1. *Let $X \sim \mathcal{N}_{|V|}(\mu, \Sigma)$ be normally distributed with regular covariance matrix Σ. Then for $i, j \in V$ with $i \neq j$ it holds*

$$X_i \perp\!\!\!\perp X_j \mid X_{V \setminus \{i,j\}} \quad \Leftrightarrow \quad k_{ij} = 0 ,$$

where k_{ij} is the respective entry of the concentration matrix $K = \Sigma^{-1}$.

In a *Gaussian graphical model* we now assume that the multivariate Gaussian variable X fulfills the pairwise Markov property with respect to an undirected graph G^u. Since the density is positive and continuous the distribution factorizes and the global and local Markov property hold. According to the above corollary, the graph structure directly relates to the zero-entries of the concentration matrix; if an entry $k_{ij} \in K$ equals zero then the respective nodes $i, j \in V$ are non-adjacent in the graph. Based on this idea, informative graph-structures have successfully been learnt from multivariate data sets (Krumsiek *et al.*, 2011; Schäfer & Strimmer, 2005).

3.4.3 Bayesian networks

Let now $G = (V, E)$ be a directed acyclic graph and let $(X_i)_{i \in V}$ be random variables with values in $(\mathcal{X}_i)_{i \in V}$. Let again $X_A = (X_a)_{a \in A}$ denote the joint random variable with values in $\mathcal{X}_\mathcal{A} = \times_{a \in A} \mathcal{X}_a$ for any subset $A \subseteq V$. The Markov properties for directed graphical models are defined as follows. The variables $(X_i)_{i \in V}$ have the *directed global Markov property* if

$$\text{(DG)} \quad X_A \perp\!\!\!\perp X_B \mid X_C \ ,$$

where A and B are separated by C in $(G_{\mathrm{An}(A \cup B \cup C)})^m$ which is the moral graph of the smallest anchestral set containing $A \cup B \cup C$. The random variables X have the *directed local Markov property* with respect to G if for all $i \in V$ one has

$$\text{(DL)} \quad X_i \perp\!\!\!\perp X_{\mathrm{nd}(i)} \mid X_{\mathrm{pa}(i)} \ .$$

Thus, a variable is conditionally independent of its non-descendants given its parents. And last, the variables have the *directed pairwise Markov property* with respect to G if for any pair (i, j) of non-adjacent nodes it holds

$$\text{(DP)} \quad X_i \perp\!\!\!\perp X_j \mid X_{\mathrm{nd}(i) \setminus \{j\}} \ .$$

The directed local and the directed global Markov property are equivalent, i. e. (DL) $\Leftrightarrow$ (DG). Furthermore, the directed local implies the directed pairwise property (DL) $\Rightarrow$ (DP). The converse is in general not true.

Let P_X be the probability distribution of $X = X_V$. P_X *factories recursively* according to the directed acyclic graph G if for all $i \in V$ there exist non-negative functions (or kernels) $k^i \colon \mathcal{X}_i \times \mathcal{X}_{\mathrm{pa}(i)} \to [0, \infty)$ together with a product measure $\mu = \otimes_{i \in V} \mu_i$ on $\mathcal{X} = \mathcal{X}_V$ such that

$$\int k^i(y_i, x_{\mathrm{pa}(i)}) \mu_i \, \mathrm{d}(y_i) = 1 \ ,$$

and X has density f with respect to μ where

$$f(x) = \prod_{i \in V} k^i(x_i, x_{\mathrm{pa}(i)}) \ .$$

One can easily show that the kernels $k^i(\,.\,, x_{\mathrm{pa}(i)})$ are densities of the conditional distributions X_i given $X_{\mathrm{pa}(i)} = x_{\mathrm{pa}(i)}$. We call a probability distribution that factorizes recursively according to G a *Bayesian network*; the distribution is defined by the conditional distributions of a node given its direct predecessors. The following theorem is according to Lauritzen (1996, p.51).

Theorem 3.2. *Let P_X be a probability distribution of X such that the density is given with respect to some product measure μ. Then P_X satisfies the directed global Markov property with respect to a directed acyclic graph G if and only if it factorizes recursively according to G.*

3.5 Parameter inference in probabilistic models

In this part we introduce parameter inference using the maximum likelihood estimate and the maximum a posteriori estimate. We further discuss expectation maximization as an interative scheme to determine these estimates; details can be found in McLachlan & Krishnan (2007). The section concludes with model selection approaches to determine the most appropriate probabilistic model for given data. More background about parameter inference and model selection are given in Held (2008).

3.5.1 Maximum likelihood estimate

Let X be a (multi-dimensional) random variable and x a realization. We also call x *observations* of X. We further assume that the density function $f = f_X$ exists and that it depends on a parameter $\theta \in \Theta$ where Θ denotes the parameter space. Here, θ might in fact be a multi-dimensional parameter vector. The aim is to draw conclusions about

θ based on the observations x. Therefore, we consider the *likelihood function* which is for fixed observations x given as

$$L(\theta) = f(x;\theta) \ , \quad \theta \in \Theta \ .$$

We are then interested in parameters θ such that the observations x become most likely, i. e. we want to maximize the likelihood function with respect to θ. This maximum likelihood estimate (MLE) is given by

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_{\theta\in\Theta} L(\theta) \ .$$

Often, one considers the log-likelihood function $\ell(\theta) = \ln f(x;\theta)$ instead of $L(\theta)$. Since the logarithm is a strictly monotonically increasing function, $\hat{\theta}^{\mathrm{ML}}$ is the same estimate in both cases.

Let now $X_1, \ldots, X_n$ be independent and identically distributed random variables with the same density function f. We consider the joint random variable $X = (X_1, \ldots, X_n)$ together with a realization $x = (x_1, \ldots, x_n)$. The likelihood function of x then factorizes as

$$L(\theta) = f(x;\theta) = \prod_{i=1}^{n} f(x_i;\theta) \ .$$

The corresponding log-likelihood function decomposes into a sum $\ell(\theta) = \ln f(x;\theta) = \sum_{i=1}^{n} f(x_i;\theta)$, accordingly. Thus, the log-likelihood function is often more convenient regarding complexity.

Score function and Fisher information

Standard errors for the maximum likelihood estimate can be calculated using the *Fisher information*. Let therefore the gradient vector and the negative second-order partial derivates of the log-likelihood function be given by

$$\begin{aligned} S(x;\theta) &= \partial \ln L(\theta)/\partial\theta \ , \\ H(x;\theta) &= -\partial^2 \ln L(\theta)/\partial\theta\,\partial\theta' \ . \end{aligned}$$

The first is known as score statistics, the latter is the Hessian matrix or the observed information matrix. The expected Fisher information matrix $\mathcal{I}(\theta)$ is then given by

$$\mathcal{I}(\theta) = E_\theta[S(X;\theta)S'(X;\theta)] = -E_\theta[H(X;\theta)] \ ,$$

where E_θ denotes expectation using the parameters θ. The asymptotic covariance matrix of the MLE is equal to the inverse of $\mathcal{I}(\theta)$ and can be approximated by $\mathcal{I}(\hat{\theta})$. Thus, the standard error of $\hat{\theta}_i$ is given by

$$SE(\hat{\theta}_i) \approx \sqrt{\mathcal{I}^{-1}(\hat{\theta})_{ii}} \approx \sqrt{H^{-1}(\hat{\theta})_{ii}}\ ,$$

where it is common to replace the expected by the observed information matrix. Furthermore, the approximate confidence interval at level $1-\alpha$ is

$$\big[\,\hat{\theta}_i \pm z_{(1-\alpha)/2}(\sqrt{H^{-1}(\hat{\theta};X)}_{ii})\,\big]\ ,$$

where $z_{(1-\alpha)/2}$ denotes the $(1-\alpha)/2$-quantile of the normal distribution.

3.5.2 Bayesian inference

In contrast to classical parameter inference from the last section, we now think of θ as a random variable with a density function. We are then interested in its posterior distribution, i. e. the distribution of the parameters θ given the observations x. The statistical background is Bayes' Theorem.

Theorem 3.3 (Bayes' Theorem)**.** *Let X and Y be continuous random variables and let x and y be realizations. Then*

$$f_{Y|X}(y \mid x) = \frac{f_{X|Y}(x \mid y)\, f_Y(y)}{\int f_{X|Y}(x \mid y)\, f_Y(y)\,\mathrm{d}y}\ .$$

From this, the *posterior distribution* of the parameters θ given observations x directly follows. For easier notation we drop the indexing of the different density functions and thus get

$$f(\theta \mid x) = \frac{f(x \mid \theta)\, f(\theta)}{\int f(x \mid \theta)\, f(\theta)\,\mathrm{d}\theta}\ .$$

The factor $f(x \mid \theta)$ is the likelihood function which we denoted by $L(\theta) = f(x;\theta)$. Since we now explicitly condition on the random variable θ we adapted the notation. The factor $f(x)$ is the prior distribution of the parameters. We also use the notation $\pi(\theta)$. The denominator is known as marginal and it holds $\int f(x \mid \theta)\, f(\theta)\,\mathrm{d}\theta = \int f(x,\theta)\,\mathrm{d}\theta = f(x)$. This factor is a normalizing constant and assures that $\int f(\theta \mid x)\,\mathrm{d}\theta = 1$. Thus, the posterior distribution of the parameters is proportional to the product of likelihood and prior

$$f(\theta \mid x) \propto f(x \mid \theta)\, f(\theta)\ .$$

For parameter maximization it often is sufficient to consider the r.h.s. The maximum a posteriori estimate (MAP) of θ is finally given as

$$\hat{\theta}^{\mathrm{MAP}} = \arg\max_{\theta \in \Theta} f(\theta \mid x)\ .$$

3.5.3 Expectation maximization

The expectation maximization (EM) algorithm is an iterative scheme to compute the maximum likelihood estimate. It can be easily extended to also provide the maximum a posteriori estimate. Expectation maximization is applicable whenever parameter inference from a complete data set would be straightforward – but the observed data is incomplete. This means that either data points are missing, or that the model contains additional variables that cannot be observed. The idea then is to replace the unobserved variables by their conditional expectation given observed variables and model parameters.

Let X denote observed random variables and let S denote unobserved (or unobservable) random variables. Given observations x, we aim to maximize the data log-likelihood $\ell(\theta) = \ln f(x;\theta)$ with respect to the parameters $\theta \in \Theta$. In contrast to $\ell(\theta)$, we assume that the *complete* data log-likelihood

$$\ell_c(\theta) = \ln f_c(x, s; \theta)$$

is easy to handle. Here, s is some unknown realization of S and with this also $\ell_c(\theta)$ is unknown. Thus, we further replace $\ell_c(\theta)$ by its conditional expectation given s, and we assume that the current parameter estimate is $\theta^{(k)}$:

$$\mathcal{Q}(\theta; \theta^{(k)}) = E_{\theta^{(k)}}[\, \ell_c(\theta) \mid s\,]\ .$$

Expectation maximization (EM) is now an iterative scheme. Given the current parameter estimates $\theta^{(k)}$, we determine $\mathcal{Q}(\theta; \theta^{(k)})$ in the E-step. In the M-step we maximize

$\mathcal{Q}(\theta;\theta^{(k)})$ with respect to the parameters θ and derive

$$\theta^{(k+1)} = \arg\max_{\theta\in\Theta} \mathcal{Q}(\theta;\theta^{(k)}) \ .$$

Both steps are alternated until convergence of the likelihood function, i. e. until one has $\ell(\theta^{(k+1)}) - \ell(\theta^{(k)}) < \epsilon$ for some threshold $\epsilon > 0$. Alternatively, a sufficiently small change in the parameters can be considered as stop criterion (Abbi *et al.*, 2008).

Maximum a posteriori estimation

We now assume that we have some prior density $f(\theta)$ of the parameters. We then aim to maximize the (incomplete) data posterior which is given as $\ln f(\theta \mid x) = \ln f(x;\theta) + \ln f(\theta)$. To this maximum using expectation maximization we consider as before the complete data set. The complete data posterior is given by

$$\ln f_c(\theta \mid x, s) = \ln f_c(x, s;\theta) + \ln f(\theta) \ ,$$

where we omit an additive term not involving θ. In the E-step we calculate the conditional expectation of $\ln f_c(\theta \mid x, s)$ given the observed data x. Let $\theta^{(k)}$ be the current parameter estimate then

$$E_{\theta^{(k)}}[\ln f_c(\theta \mid x, s) \mid s\,] = \mathcal{Q}(\theta;\theta^{(k)}) + \ln f(\theta) \ .$$

In contrast to maximum likelihood estimation, we have an additional term given by the prior $\ln f(\theta)$. In the M-step we then maximize $E_{\theta^{(k)}}[\ln f_c(\theta \mid x, s) \mid s\,]$ with respect to $\theta \in \Theta$ and get $\theta^{(k+1)}$.

3.5.4 Model selection

The log-likelihood function $\ell(\theta) = \ln f(x;\theta)$ evaluated at the maximum likelihood estimate describes how good the model explains the data. However, this quantity can not be used to compare different models – a more complex model will directly lead to a higher log-likelihood value. To avoid overfitting, one needs to penalize for the model complexity.

There are two widely-used criteria for model selection that are based on the likelihood function. Both criteria differ in the penalty term for the model complexity. The Akaike information criterion (AIC) (Akaike, 1974) is given by

$$AIC = 2\,\ell(\hat{\theta}) - 2k \ ,$$

where $\hat{\theta}$ denotes the maximum likelihood estimate of the parameters and k is the number of model parameters. The second criterion is the Bayesian information criterion (BIC) (Schwarz, 1978) given by

$$BIC = 2\,\ell(\hat{\theta}) - \ln(n)k \ ,$$

where n denotes the number of observations. Thus, for $n \geq 8$ the BIC gives a stronger penalty compared to the AIC.

Bayesian model selection

For reasons of completeness and since the main contribution of this thesis – the algorithm `emGrade` – can be easily treated in a fully Bayesian manner we additionally discuss Bayesian model selection. So far, this method has not been applied in the contents of this thesis but it is useful for follow-up work.

To decide between two models $\mathcal{M}_1$ and $\mathcal{M}_2$ in a Bayesian context, we first define prior probabilities $\mathrm{P}(\mathcal{M}_1)$ and $\mathrm{P}(\mathcal{M}_2)$ of the models. The probabilities need to sum up to 1 and if no further information about model preference is available one can set both quantities equal to 0.5. The posterior probability of the model $\mathcal{M}_1$ (and for $\mathcal{M}_2$, accordingly) is then given by

$$\mathrm{P}(\mathcal{M}_1 \mid x) = \frac{f(x \mid \mathcal{M}_1)\,\mathrm{P}(\mathcal{M}_1)}{f(x \mid \mathcal{M}_1)\,\mathrm{P}(\mathcal{M}_1) + f(x \mid \mathcal{M}_2)\,\mathrm{P}(\mathcal{M}_2)} \ .$$

This follows directly from Bayes' theorem, where the integral in the denominator is here replaced by a (discrete) sum. The marginal likelihood in the nominator is of the form

$$f(x \mid \mathcal{M}_1) = \int f(x \mid \theta_1, \mathcal{M}_1) \cdot f(\theta_1 \mid \mathcal{M}_1)\,\mathrm{d}\theta_1 \ ,$$

where θ_1 might actually be a parameter vector.

The posterior chance $\mathrm{P}(\mathcal{M}_1 \,|\, x)/\mathrm{P}(\mathcal{M}_2 \,|\, x)$ is then given by

$$\underbrace{\frac{\mathrm{P}(\mathcal{M}_1 \mid x)}{\mathrm{P}(\mathcal{M}_2 \mid x)}}_{posterior\,chance} = \underbrace{\frac{f(x \mid \mathcal{M}_1)}{f(x \mid \mathcal{M}_2)}}_{Bayes\,factor} \times \underbrace{\frac{\mathrm{P}(\mathcal{M}_1)}{\mathrm{P}(\mathcal{M}_2)}}_{prior\,chance} \;,$$

where the first factor is called *Bayes factor*, the second is the prior chance. The Bayes factor $B_{12} = f(x \,|\, \mathcal{M}_1)/f(x \,|\, \mathcal{M}_2)$ can be interpreted as quotient of posterior chance of $\mathcal{M}_1$ and prior chance of $\mathcal{M}_1$. Thus, for $B_{12} > 1$ the data increases the propability of model $\mathcal{M}_1$ and one would prefer this model. To interpret the Bayes factor, Jeffreys (1961) originally invented a scheme to assign relevances to all possible values B_{12}. The scheme was slightly modified by Kass & Raftery (1995) and thus claims: 1-3 is not worth more than a bar mention, 3-20 is positive, 20-150 is strong, and above 150 is very strong evidence.

With this, we provided all necessary mathematical background – probabilistic models, stochastic processes, graphical models, parameter inference and model selection. In the next chapter we introduce blind source separation and give an overview of existing methods. Both chapters then build the basis for new BSS investigations in remaining part of this thesis.

4

Blind source separation

In this chapter we state the blind source separation problem and provide a review about existing blind source separation approaches. In particular we introduce independent component analysis, non-negative matrix factorization and (joint) diagonalization of autocovariance matrices. We discuss different joint diagonalization approaches for weakly stationary time series and review the algorithm `Grade` (Kowarsch *et al.*, 2010) which is an extension to network data. This chapter provides the basis for Chapters 5 and 6 where we determine the limiting distributions of the mixing estimates for time series data and invent the new probabilistic method `emGrade` for network data. The chapter concludes with a discussion about performance indices that we use to evaluate and compare algorithms in this thesis.

For better distinction, we use bold symbols to denote random variables in the remaining part of this thesis. Solid symbols denote realizations of random variables and parameters.

4.1 The idea of BSS

In the basic blind source separation (BSS) model we assume a p-dimensional observed random process $\{\boldsymbol{x}(t)\}_{t\in\mathbb{Z}}$ where the components of $\boldsymbol{x}(t)$ are generated by an instantaneous linear mixing

$$\boldsymbol{x}(t) = A\,\boldsymbol{s}(t) + \boldsymbol{\varepsilon}(t)\,, \quad t \in \mathbb{Z}. \tag{4.1}$$

Here, $A \in \mathbb{R}^{p \times q}$ is a deterministic full-rank mixing matrix and $\{\boldsymbol{s}(t)\}_{t \in \mathbb{Z}}$ is a q-dimensional unobserved process. The components $\{\boldsymbol{s}_k(t)\}_{t \in \mathbb{Z}}$ for $k = 1, \ldots, q$ are known as source signals. In general, one assumes that the unobserved variables $\boldsymbol{s}(t)$ are statistically stationary, i. e. $E[\boldsymbol{s}(t)]$ and $E[\boldsymbol{s}(t)\boldsymbol{s}(t)']$ are independent of the index $t \in \mathbb{Z}$. Finally, $\{\boldsymbol{\varepsilon}(t)\}_{t \in \mathbb{Z}}$ denotes an additive noise process which is independent of the unobserved process and with $\boldsymbol{\varepsilon}(t_1)$ independent of $\boldsymbol{\varepsilon}(t_2)$ for $t_1 \neq t_2$. Many BSS models assume white Gaussian noise with independently and identically distributed components, i. e. $\boldsymbol{\varepsilon}(t) \sim \mathcal{N}(0, \sigma^2 I_p)$ for $t \in \mathbb{Z}$; other models are noise-free and we omit the term $\boldsymbol{\varepsilon}(t)$ in (4.1). As stated in Section 3.3, any oberseved data set consists of countably many data points. We therefore use $\mathbb{Z}$ as index set for the random processes. Figure 4.1 illustrates the idea of BSS. Shown are three independently generated time series signals together with three linear mixtures of these signals. Given the mixtures the task of BSS is to recover both – the source signals and the mixing process.

Important for BSS models is the relation between the number of observations p and the number of source signals q. In the simplest case we have $p = q$ and the mixing matrix is quadratic. If $p > q$ the model is *overdetermined* and one is interested in a small number of representative and informative source signals. In this case, the concrete number of source signals is usually unknown. Tong *et al.* (1990), for example, determined the number of source signals based on the eigenvalue distribution of the sample covariance; Choudrey & Roberts (2001) used hyperparamters in their estimation procedure to intrinsically learn the number of signals; and in our new BSS method in Chapter 6 we apply model selection criteria to compare the estimation performances for different values q. Finally, the most challenging case is the *underdetermined* model where $p < q$. Applications of such models are, for example, coding theory where high-dimensional representations of the data are needed or signal encryption (Lin *et al.*, 2006; Yang *et al.*, 2008). Since the number of unknown model components can drastically outreach the number of observations, separability issues arise (Albera *et al.*, 2004; Cao & Liu, 1996; He *et al.*, 2008).

In the literature, various generalizations of the basic BSS model have been introduced. These include, amongst others, non-linear mixing models where the mixing matrix A is replaced by a non-linear function $\psi\colon \mathbb{R}^p \to \mathbb{R}^q$ or non-instantaneous (or *convolutive*) mixing models where

$$\boldsymbol{x}(t) = \sum_{k=0}^{K} A_k\, \boldsymbol{s}(t-k) + \boldsymbol{\varepsilon}(t)\,, \quad t \in \mathbb{Z}.$$

A discussion on convolutive mixing models is, for example, given in Parra & Spence (2000) and Wang *et al.* (2003).

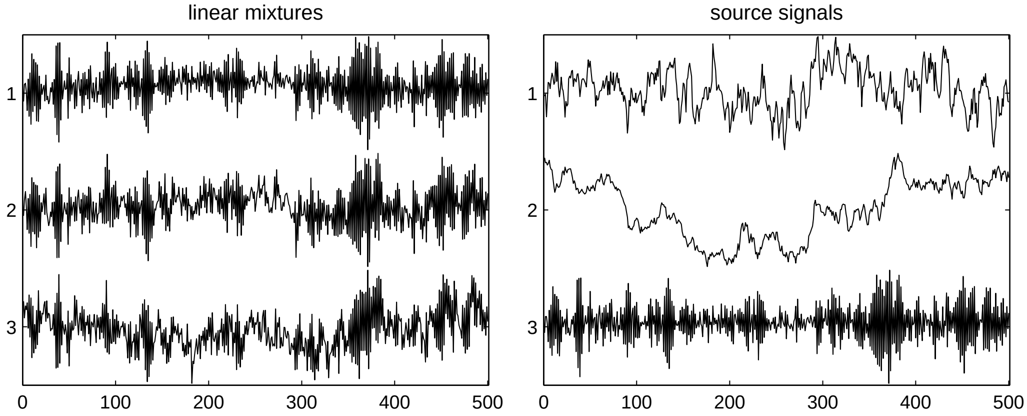

Figure 4.1: Separation of time series signals. Three idependently generated source signals (left) together with three linear mixtures (right). The source signals are AR(1)-processes with coefficients 0.9, 0.99999 and -0.9, respectively, and the mixing matrix is (row-wise) given by $[-0.03, -0.11, 0.43; -0.12, -0.02, 0.24; -0.63, 0.72, -0.82]$.

4.1.1 Properties and indeterminacies of BSS models

Without further assumptions on source signals or mixing matrix in (4.1) the separation of $\boldsymbol{x}(t)$ into A and $\boldsymbol{s}(t)$ is not uniquely defined. For any invertible matrix $C \in Gl(p)$ we get an equivalent decomposition

$$\boldsymbol{x}(t) = A\,\boldsymbol{s}(t) + \boldsymbol{\varepsilon}(t) = (AC)(C^{-1}\boldsymbol{s}(t)) + \boldsymbol{\varepsilon}(t) = \tilde{A}\,\tilde{\boldsymbol{s}}(t) + \boldsymbol{\varepsilon}(t)\;.$$

Therefore, the various existing BSS approaches introduce additional assumptions on the source signals (or the mixing matrix). Usually, the above indeterminacy can be limited to matrices with one non-zero entry per row and column. Such matrices can be written as a product LPD, where L is a sign-changing matrix, i. e. diagonal with entries ± 1, P is a permutation matrix, and D is diagonal with positive entries. Furthermore, one can assume $\mathrm{Cov}\big(\boldsymbol{s}(t), \boldsymbol{s}(t)\big) = I_p$ without loss of generality; the separation indeterminacy then reduces to sign-changing permutations LP. Discussions about model-specific indeterminacies are provided in the respective sections.

An important property of BSS models and algorithms is *affine equivariance*. The property relates the (un-)mixing estimates derived from dependent data sets and justifies, for example, data pre-whitening which we introduce in the next section. A BSS *model* with additional model assumptions $(\mathcal{A}_i)_{i\in I}$ is affine equivariant if for any solution A_0 and $s_0(t)$ of data $X = \big(x(t)\big)_{t=1}^{T}$ we have that CA_0 and $s_0(t)$ is a solution of data CX where $C \in Gl(p)$. Here, solution means, that the model assumptions are fulfilled. The definition further translates to BSS *algorithms*: If $\hat{A}$ is the mixing estimate derived with an affine equivariant BSS method from data X then the mixing estimate derived from data CX is given by $C\hat{A}$. Sometimes, one uses transformations of the coordinate system to assure affine equivariance of a BSS method (Ilmonen *et al.*, 2012). Furthermore, in Section 5.2.3 we introduce the algorithm `SOBIparis2` which depends on the initial value; here, pre-processing using a coherent initialization yields affine equivariance of the algorithm.

4.1.2 Centering and whitening

A widely-used pre-processing when applying BSS methods is centering and pre-whitening of the data. This means that the original data is transformed to new data with zero mean and unit variance. The pre-processing is commonly done in case of BSS models where we assume $E[\boldsymbol{s}(t)] = 0$ and $\mathrm{Cov}(\boldsymbol{s}(t), \boldsymbol{s}(t)) = I_q$ for $t \in \mathbb{Z}$. Here, one is usually interested in the decomposition of the centered data. Besides robustness, pre-whitening forces the mixing matrix to be orthogonal and many BSS methods are restricted to the estimation of such matrices. After performing BSS on the whitened data one transfroms the mixing estimate back to the original (centered) data.

Centering of the data $X = \big(x(t)\big)_{t=1}^{T}$ can be easily done by substracting the sample mean $E[x(t)] = \frac{1}{T}\sum_{t=1}^{T} x(t)$ from each observation. After centering we derive a whitening matrix $C \in Gl(p)$ using the singular value decomposition. Let therefore $R_X = E[x(t)x(t)']$ be the sample covariance of the centered data; it can be calculated as $R_X = \frac{1}{T-1}\sum_{t=1}^{T} x(t)x(t)'$ and to assure symmetry we use $(R_X + R_X')/2$ instead.

We then consider the singular value decomposition $R_X = U\Sigma V'$ where $U, V \in \mathcal{O}(p)$ and Σ contains the singular values $\psi_1 \geq, \ldots, \geq \psi_r > 0$ on its main diagonal; here, $r \leq p$ denotes the rank of R_X. With this, a whitening matrix is defined as $C = \Sigma_0 U_0'$ where $\Sigma_0 = \operatorname{diag}(1/\sqrt{\psi_1}, \ldots, 1/\sqrt{\psi_r})$ is a diagonal matrix and U_0 consists of the first r columns of U. The new data $CX = \big(Cx(t)\big)_{t=1}^{T}$ has then unit variance. To see this we need to consider the strong relation between singular value and eigenvalue decomposition discussed in Section 3.1. For the symmetric and positive semi-definite covariance matrix R_X an eigenvalue decomposition is given by $R_X = U\Sigma U'$ and we, thus, have

$$\begin{aligned} \operatorname{Cov}(Cx(t), Cx(t)) &= \Sigma_0 U_0' \operatorname{Cov}(x(t), x(t))(\Sigma_0 U_0')' \\ &= \Sigma_0 U_0' (U\Sigma U') U_0 \Sigma_0 \\ &= \Sigma_0 \Sigma \Sigma_0 = I_p \ . \end{aligned}$$

In practice, we use the singular value decomposition to determine a whitening matrix; this decomposition can be calculated more robustly in general.

The above is closely related to principal component analysis (PCA). Here, the data is projected to a new coordinate system by an orthogonal linear transformation. The first new coordinate describes the dimension with highest variance in the original data, the second new coordinate is orthogonal to the first and, again, describes the dimension with the highest variance and so on. The transformed data has uncorrelated components and can be determined as $UX = \big(Ux(t)\big)_{t=1}^{T}$ (Hyvarinen *et al.*, 2002). In contrast to data whitening, the covariance is here given by Σ, i. e. by the singular values of R_X.

4.2 Independent component analysis

In independent component analysis (ICA) one assumes identical and independently distributed random variables $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ such that

(I1) the components $\boldsymbol{s}_1(t), \ldots, \boldsymbol{s}_q(t)$ are stochastically independent

(I2) at least one component is Gaussian distributed

Comon (1994) originally stated that with (I1)-(I2) and a full-rank mixing matrix the source signals are uniquely determined up to scaling and permutation. Further identifiability properties have been investigated in Eriksson & Koivunen (2004) and Theis &

Gruber (2005). In the following we shortly outline two substantially different approaches to perform ICA.

4.2.1 Maximization of non-Gaussianity

A widely-used ICA approach is based on maximization of non-Gaussianity of the signals. The idea is that a linear mixture of stochastically independent random variables is more Gaussian than the independent random variables itselves. This is a consequence of the central limit theorem. Gaussianity here means the closeness of a distribution to the Gaussian distribution. A measure is the kurtosis (Section 3.2.3); if a random variable is Gaussian distributed then the kurtosis is zero. Algorithms based on this idea are, for example, `FOBI` (Cardoso, 1990) and `JADE` (Cardoso & Souloumiac, 1993). Most efficiently, `fastICA` (Hyvärinen & Oja, 1997) performs fixed-point kurtosis maximization. After pre-whitening, $w \in \mathbb{S}^{p-1}$ is chosen on the sphere such that $\operatorname{kurt}(w'X)$ is maximal for given data X. Here, the gradient of $\operatorname{kurt}(w'X)$ is given by

$$\frac{\partial \operatorname{kurt}(w'X)}{\partial w'} = 4\,(E[(w'X)^3 X] - 3\|w\|_2^2\, w)$$

and for an extremal point $w \in \mathbb{S}^{p-1}$ it holds $w \propto \operatorname{grad}(\operatorname{kurt}(w'X))$. A detailed derivation can be found in Theis (2003). With this, w can be determined iteratively (Algorithm 1) and the respective independent component (IC) is given by $w'X$. The above can be extended to determine a $p \times p$ unmixing matrix W and p independent components. The rows of W can be either estimated one after the other in a deflation-based approach or simultaneously. For both one needs to assure that W is orthogonal. The two different conceptual appraoches are discussed in more detail in Chapter 5 where we consider BSS for weakly stationary time series.

4.2.2 Maximum likelihood estimation

Furthermore, maximum likelihood approaches have been introduced for the ICA model. Let therefore X and S collect the observed and unobserved variables, respectively. If we assume white Gaussian noise with diagonal covariance matrix Σ in (4.1) then the likelihood of the BSS model is given by

$$f(X \mid A, S, \Sigma) = \det(2\pi\Sigma)^{-\frac{N}{2}} e^{-\frac{1}{2}\operatorname{Tr}((X-AS)'\Sigma^{-1}(X-AS))} \,.$$

Algorithm 1: fastICA (one independent component)

```
input  : Pre-whitened data $X$
output: One independent component IC, unmixing vector $w$
initialize $w \in \mathbb{S}^{p-1}$ randomly
repeat
    $w^{(0)} = w$
    $v = E[(w'X)^3 X] - 3w$
    $w = v/\|v\|_2$
until $\|w - w^{(0)}\|_2 < \varepsilon$;
IC $= w'X$
```

To estimate model parameters A, Σ and source signals S one can use expectation maximization methods. Belouchrani & Cardoso (1994) provided, for example, an expectation maximization scheme together with a stochastic version and Højen-Sørensen *et al.* (2002) derived advanced mean-field approaches. An application of likelihood-based ICA to neuroimaging data can be found in Hansen (2000).

4.3 BSS for weakly stationary time series

If the observations consist of time series signals we can use the temporal dependence to perform source separation. Instead of assuming independent source components we assume that the components are uncorrelated – even when shifted along the time axis. Thus, the source components contain different temporal information and are not identical up to a time shift. In contrast to ICA, this model allows multiple Gaussian components in the unobserved process.

4.3.1 Diagonalization of autocovariances

To perform source separation based on the second-order statistics of the data we assume that the stochastic processes are weakly stationary (Section 3.3). This means that the mean and the autocovariances at any lag $\tau \in \mathbb{Z}$ do not change with respect to time. The separation assumption is that the autocovariances of the unobserved process at different lags are diagonal. In more detail, let the observed process $\{\boldsymbol{x}(t)\}_{t\in\mathbb{Z}}$ be zero-centered after mean-removal and for the unobserved process $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ we assume

(A1) $E[\boldsymbol{s}(t)] = 0_p$,

(A2) $\operatorname{Cov}(\boldsymbol{s}(t), \boldsymbol{s}(t)) = I_p$,

(A3) $\operatorname{Cov}(\boldsymbol{s}(t), \boldsymbol{s}(t+\tau)) = D_\tau$ is diagonal for all lags $\tau \in \mathbb{Z}$, and

(A4) for all $i \neq j \in \{1, \dots, p\}$ there exists a lag $\tau \in \mathbb{Z}$ such that $d_{\tau i} \neq d_{\tau j}$ where $d_{\tau i}$ and $d_{\tau j}$ are the ith and jth diagonal entry of D_τ .

As stated in Sections 4.1.1 and 4.1.2 the assumptions (A1) and (A2) are without loss of generality. If we assume a noise-free linear mixing the autocovariances of the observed process are given by

$$\operatorname{Cov}(\boldsymbol{x}(t), \boldsymbol{x}(t+\tau)) = \begin{cases} A\, D_\tau A' & \tau \neq 0 \, , \\ AA' & \tau = 0 \, . \end{cases}$$

According to (A4), the mixing matrix A is uniquely determined up to sign and permutation of the columns. To see this we consider the whitened process $\{\boldsymbol{x}^*(t)\}_{t\in\mathbb{Z}}$ with mixing matrix $A^* \in \mathcal{O}(p)$ introduced in Section 4.1.2. The decomposition $A^* D_\tau (A^*)'$ is then an eigenvalue decomposition of $\operatorname{Cov}(\boldsymbol{x}^*(t), \boldsymbol{x}^*(t+\tau))$. At a single lag τ some of the eigenvalues $d_{\tau 1}, \dots, d_{\tau p}$ might be identical and the columns of A^* are not uniquely determined. Together with (A4) we find that each column of A^* generates a 1-dimensional intersection of eigenspaces. Thus, A^* (and with this also A) is uniquely determined up to sign and permutation.

In the presence of white Gaussian noise, i. e. $\boldsymbol{\varepsilon}(t) \sim \mathcal{N}(0, \sigma^2 I_p)$ for $t \in \mathbb{Z}$, the autocovariance at lag zero is given by $\operatorname{Cov}(\boldsymbol{x}(t), \boldsymbol{x}(t)) = AA' + \sigma^2 I_p$. If we assume that the noise variance is known or can be estimated from the data we get the same identifiability properties as before (Tong *et al.*, 1990).

If now observations $X = \big(x(t)\big)_{t=1}^T$ are given, one can jointly diagonalize sample autocovariances to determine an unmixing estimate. The sample autocovariance at lag $\tau \in \mathbb{Z}$ is given by

$$M_\tau = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} x(t) x(t+\tau)' \, .$$

For joint diagonalization we use sample autocovariances at distinct lags $\tau_1, \dots, \tau_K \in \mathbb{Z}$ and assume that (A4) holds for $\{\tau_1, \dots, \tau_K\}$ instead of $\mathbb{Z}$. For better readability, we denote the sample autocovariances as $M_1, \ldots, M_K$ even if the lags are different from $1, \ldots, K$. An unmixing estimate is then a $p \times p$ matrix $W = (w_1, \ldots, w_p)'$ that minimizes the off-diagonal elements of WM_kW' for all $k = 1, \ldots, K$ in the sense that

$$f(W) = \sum_{k=1}^{K} \|\mathrm{off}(WM_kW')\|_{\mathrm{F}}^2 \, . \tag{4.2}$$

is minimized under the constraint $WM_0W' = I_p$. Here, $\mathrm{off}(M) = M - \mathrm{diag}(M)$ with $\mathrm{diag}(M)$ a diagonal matrix consisting of the diagonal entries of M. From the spectral theorem it follows that an optimal solution W is indeed an estimate of the unmixing matrix.

4.3.2 Joint diagonlization algorithms

Many algorithms are based on the idea of (jointly) diagonalizing autocovariances, e. g. `AMUSE` (Tong *et al.*, 1990), `SOBI` (Belouchrani *et al.*, 1997), `TDSEP` (Ziehe & Müller, 1998), `ACDC` (Yeredor, 2002), and `LSDIAG` (Ziehe *et al.*, 2003). Most of the algorithms require pre-whitening of the data since the (un-)mixing estimate is orthogonal by construction. For image processing the assumption of uncorrelated source components can be extended to the spatial dimension of the data (Schießl *et al.*, 2000; Theis *et al.*, 2008). Moreover, multi-dimensional autocovariances have been introduced yielding the algorithm `mdSOBI` (Theis *et al.*, 2004b). A review on joint diagonalization algorithms is for example given in (Theis & Inouye, 2006). In the following we explain three important algorithms in more detail.

`SOBI` (Belouchrani *et al.*, 1997) is the original second-order blind identification algorithm and jointly diagonalizes sample autocovariances $M_1, \ldots, M_K$ of pre-whitened data. The algorithm is a generalization of the Jacobi technique to determine eigenvalue decompositions (Section 3.1). Starting with an orthogonal initial guess for the unmixing matrix, the algorithm determines for each pair of rows in turn an optimal Jacobi rotation to maximize (4.2). The current unmixing estimate is then rotated in the plane spanned by the two rows. If only one sample autocovariance is considered (i. e. $K = 1$) the algorithm exactly yields the Jacobi technique. Similar joint diagonalizing procedures have been discussed in Bunse-Gerstner *et al.* (1993) and Cardoso & Souloumiac (1996).

ACDC (Yeredor, 2002) is a non-orthogonal algorithm to jointly diagonalize $M_1, \ldots, M_K$. The *weighted* cost-function is here defined by

$$h(A, D_1, \ldots, D_K) = \sum_{k=1}^{K} \lambda_k \|M_k - AD_kA'\|_{\mathrm{F}}^2 ,$$

where $\lambda_1, \ldots, \lambda_K \in \mathbb{R}_+$ are known positive weights and $D_1, \ldots, D_K$ are diagonal matrices. The algorithm performs two optimization steps in turn. In the AC-step the above cost-function is minimized w.r.t. one single column of the mixing matrix A and in the DC-step it is minimized w.r.t. all diagonal matrices D_k. Since the final mixing estimate is not necessarily orthogonal, ACDC does not require pre-whitened data. To approximately fulfill (A2) from Section 4.3.1 the set of autocovariances should include M_0 at lag zero.

The "Algorithm for Multiple Unknown Signals Extraction" (AMUSE) was introduced by Tong *et al.* (1990). In contrast to true joint diagonalization approaches, only one autocovariance (at a single lag $\tau \in \mathbb{Z}$) is considered. The algorithm forms the basis for the extension to network data in the next section and we therefore present data-pre-processing and the diagonalization scheme in more detail. The original algorithm assumes white Gaussian noise and determines its variance from the data. In Algorithm 2 we only present the version for a noise-free linear mixing.

Algorithm 2: Amuse

```
input  : Observations $X$
output: Source signals $S$, unmixing matrix $W$

% center and pre-whiten data
$x(t) \leftarrow x(t) - E[x(t)]$ remove sample mean
$R_X = E[x(t)x(t)']$ sample covariance
$R_X \leftarrow U\Sigma V'$ singular value decomposition
$C = \Sigma_0 U_0'$ (Section 4.1.2)
$Y = CX$ whitened data

% diagonalization
$R_Y(\tau) = E[y(t)y(t-\tau)']$ sample autocovariance at lag $\tau$
$R_Y(\tau) \leftarrow U\Sigma V'$ singular value decomposition
$W = U'C$
$S = U'CX$
```

4.3.3 The Grade algorithm

Kowarsch *et al.* (2010) generalized the concept of weakly stationary time series to signaling networks and provided a BSS approach similarly to `AMUSE`. The idea is to identify source components that are uncorrelated when shifted along a given graph. Since the separation is based on diagonalizing the *graph-delayed* covariance matrix we introduce the concept at this point. The following refers to some properties of graphs discussed in Section 3.4.1.

Let $G = (V, E, \mathcal{K})$ be a weighted directed graph with $V = \{1, \ldots, N\}$ the set of nodes, $E \subseteq V \times V$ the set of edges and $\kappa_{ij} \in \mathbb{R}$ are weights assigned to the edges $(i, j) \in E$. We further assume, that $1, \ldots, n_0 - 1$ are the root nodes of the graph. Let now $\big(\boldsymbol{s}(i)\big)_{i=1}^{N}$ be random variables indexed according to the nodes V. Kowarsch *et al.* introduced the graph-shift of $\boldsymbol{s}(i)$ as

$$\boldsymbol{s}^G(i) = \sum_{j \in pa(i)} \kappa_{ji} \boldsymbol{s}(j) \ , \tag{4.3}$$

where $pa(i)$ are all parent nodes of i. With this, they defined the graph-delayed covariance as $\mathrm{Cov}(\boldsymbol{s}(i), \boldsymbol{s}^G(i))$ and assumed that it is independent of the index i. This is the stationarity assumption for graphs. Similarly to (A1)-(A4) for weakly stationary time series data (Section 4.3.1), we get the following separation assumptions for signaling data

(A1) $E[\boldsymbol{s}(i)] = 0_p$
(A2) $\mathrm{Cov}(\boldsymbol{s}(i), \boldsymbol{s}(i)) = I_p$
(A3) $\mathrm{Cov}(\boldsymbol{s}(i), \boldsymbol{s}^G(i)) = D^{\mathrm{Pa}}$ is diagonal
(A4) the (diagonal) entries of D^{Pa} are pairwise different

Let now $x(1), \ldots, x(N)$ be pre-whitened data. To determine an unmixing estimate W one diagonalizes the sample graph-delayed covariance which is given by

$$D_X^{\mathrm{Pa}} = \frac{1}{N - n_0 - 1} \sum_{i=n_0}^{N} \sum_{j \in pa(i)} \kappa_{ji} x(j) x(i)' \ . \tag{4.4}$$

The resulting algorithm is called `Grade` – graph-decorrelation algorithm – and an implementation is provided in Algorithm 3. In case of a line-graph the above BSS approach corresponds to `AMUSE`, and in case of a graph with no egdes we get the same as principal component analysis. In Chapter 6 we invent a new probabilistic formulation of the algorithm where we use a Bayesian network to model the variables.

Algorithm 3: Grade

```
input  : Observations $X$, weighted directed acyclic graph $G$
output: Source signals $S$, unmixing matrix $W$, graph-decorrelation $D^{\mathrm{Pa}}$

% center and pre-whiten data
$x(i) \leftarrow x(i) - E[x(i)]$ remove sample mean
$R_X = E[x(i)x(i)']$ sample covariance
$R_X \leftarrow U\Sigma V'$ singular value decomposition
$C = \Sigma_0 U_0'$ (Section 4.1.2)
$Y = CX$ whitened data

% diagonalization
$D_Y^{\mathrm{Pa}} = E[y(i)y^G(i)']$ sample graph-delayed covariance
$D_Y^{\mathrm{Pa}} \leftarrow U\Sigma V'$ singular value decomposition
$W = U'C$
$S = U'CX$
```

4.4 Non-negative matrix factorization

To give a broader view of existing BSS approaches we shortly outline non-negative matrix factorization (NMF). In NMF we assume non-negative observations and sources together with a non-substractive mixing. In contrast to previously discussed methods, we have no further assumptions about statistical dependencies of the latent variables. The non-negativity constraint opens a large field of applications and it has successfully been applied in the context of image recognition (Lee & Seung, 1999), text data mining (Pauca *et al.*, 2004) and gene expression analysis (Gao & Church, 2005; Kim & Park, 2007).

Let now $X \in \mathbb{R}^{p\times N}$ be an observed non-negative matrix. We aim to decompose X into non-negative matrices $A \in \mathbb{R}^{p\times q}$ and $S \in \mathbb{R}^{q\times N}$ such that $X \approx AS$. NMF can be formulated as a constrained optimization problem where we minimize

$$f(A,S) = \frac{1}{2}\,\|X - AS\|_{\mathrm{F}}^2 \;\; s.t.\; A,S \geq 0\;.$$

Here, $A, S \geq 0$ means that all entries of A and S are non-negative. Lee & Seung (2001) introduced multiplicative update rules to determine A and S – both updates assure a non-increasing cost-function f. A detailed review about existing NMF algorithms is given in Berry *et al.* (2007).

Furthermore, penalized optimization can be used to learn a sparse non-negative representation of X. The problem is known as sparse NMF and can be formulated as

$$f_{\text{pen}}(A, S) = \frac{1}{2} \|X - AS\|_{\text{F}}^2 + \alpha \|A\|_{\text{F}}^2 + \beta \|S\|_{\text{F}}^2 \ \ s.t.\ A, S \geq 0 \ .$$

Here, α and β are the penalization parameters that press single entries of A and S to zero. According to Tibshirani (1996) the L_1-norm is more appropriate to enforce sparsity than the L_2-norm. Sparse NMF has for example been applied to gene expression data to classify cancer (Gao & Church, 2005; Kim & Park, 2007). In Appendix A we discuss a similar penalization for the problem of jointly diagonalizing autocovariances. However, since the non-negativity constraint is replaced by a (contradictory) orthogonality constraint naive numerical optimization fails.

4.5 Performance indices

In this part we discuss performance measures for BSS approaches. Since NMF is very different in terms of identifiability we focus on ICA and weakly stationary time series approaches. For these BSS methods a variety of performance indices have been proposed. An overview and comparison is, for example, given by Nordhausen *et al.* (2011). There are two different conceptual approaches – the first is based on the (un-)mixing estimate the second on the estimated source sequence.

For an evaluation based on the (un-)mixing estimate one typically considers the gain matrix $G = \hat{W}A$ where $\hat{W}$ denotes the unmixing estimate and A is the true mixing matrix. A well-known index that is easy to compute is the Amari error (Amari *et al.*, 1996) defined by

$$\text{AE}(G) = \frac{1}{2p(p-1)} \Big[\sum_{i=1}^{p} \Big(\sum_{j=1}^{p} \frac{|g_{ij}|}{\max_k |g_{ik}|} - 1 \Big) + \sum_{j=1}^{p} \Big(\sum_{i=1}^{p} \frac{|g_{ij}|}{\max_k |g_{kj}|} - 1 \Big) \Big] \ ,$$

where $G = (g_{ij})_{ij}$. The index yields values in $[0, 1]$, and the smaller the value the better is the estimate $\hat{W}$. Due to a common indeterminacy of BSS models (Section 4.1.1) we have for any true unmixing matrix that $WA = LPD$, where L is a sign-changing matrix, P is a permutation matrix and D is diagonal with positive entries. The Amari error only corrects for sign-changing and permutations. To fairly compare unmixing estimates in the presence of scaling indeterminacies, Nordhausen *et al.* (2008) introduced a unique sorting and scaling of $\hat{W}$.

An affine invariant performance index is the minimum distance index (Ilmonen *et al.*, 2010; Theis *et al.*, 2004a) defined by

$$\mathrm{MDI}(G) = \frac{1}{\sqrt{p-1}} \inf_{C \in \mathcal{C}} \|C\hat{W}A - I_p\|_{\mathrm{F}} \ .$$

Here, $\mathcal{C}$ is the set of $p \times p$ matrices with one non-zero entry per row and column. In other words, $\mathcal{C}$ consists of matrices $C = LPD$ with L, P and D as before. Thus, the MDI is independent of sign, permutation and scaling of the rows of $\hat{W}$. The index yields values in $[0, 1]$, and we say that the unmixing estimate $\hat{W}$ is close to the true unmixing matrix if this value is close to zero. For a fast computation we consider $\tilde{G}$ with entries $\tilde{g}_{ij} = g_{ij}^2 / \sum_{k=1}^{p} g_{ik}^2$ for $i, j = 1, \dots, p$. The MDI can then be reformulated as

$$\mathrm{MDI}(G) = \frac{1}{\sqrt{p-1}} \Big(p - \max_{P} (\mathrm{Tr}(P\tilde{G})) \Big)^{1/2} \ ,$$

with permutation matrix P. The maximization can efficiently be calculated using linear programming.

To evaluate the performance based on the estimated source signals $\hat{S}$ one uses the mean squared error

$$\mathrm{MSE}(\hat{S}, S) = \frac{1}{dN} \min_{L,P} \|LP\hat{S}^* - S^*\|_{\mathrm{F}}^2 \ .$$

As before, L is a sign-changing matrix and P a permutation matrix. $\hat{S}^*$ and S^* denote the estimated and true source signals scaled to unit sample variance. In the signal processing community, e. g. in Karvanen *et al.* (2000), one often considers the signal-to-inference ratio. This performance measure transforms the MSE to logarithmic scale and is defined as $\mathrm{SIR} = -10 \log_{10}(\mathrm{MSE}(\hat{S}, S))$.

In Chapter 6 we introduce a new BSS approach where the mixing matrix is not necessarily quadratic and the mixing estimate is unique up to sign and permutation of the columns. Mixing matrix and source signals are jointly estimated using a log-likelihood approach and the source signals follow a normal distribution with unit variance. To evaluate $\hat{A} \in \mathbb{R}^{p \times q}$ and $\hat{S} \in \mathbb{R}^{q \times N}$ with respect to the model assumptions and model indeterminacies we use a normalized version of the Frobenius matrix norm and correct for the model indeterminancies. We define the distance measures for mixing matrix and source signals as

$$\mathrm{minDist}(\hat{A}, A) = \frac{1}{\sqrt{pq}} \min_{L,P} \| \hat{A}LP - A \|_{\mathrm{F}} \ , \tag{4.5}$$

$$\mathrm{minDist}^{\dagger}(\hat{S}, S) = \frac{1}{\sqrt{qN}} \min_{L,P} \| LP\hat{S} - S \|_{\mathrm{F}} \ . \tag{4.6}$$

Both distances are not scaled to $[0, 1]$ but like above a low value indicates a better fit. The distances minDist and MDI usually differ since only the latter optimizes with respect to the scaling of the mixing rows. minDist†, in contrast, is directly related to the MSE – in our model we assume that the sources are scaled to unit variance and thus we omit the scaling to unit sample variance for the distance measure.

In this chapter we gave a review about existing BSS methods for time series data, images and network data. In particular, we introduced the concept of joint diagonalization of autocovariances – this concept is continued in Chapter 5 – and we introduced the algorihtm `Grade` which leads the way to its probabilistic extension `emGrade` in Chapter 6.

5

Separation of time series data

In this chapter we start our investigation of probabilistic blind source separation approaches with time series data. Thus, the variables (time points) are linked by a line-graph and we move on to more complex structures in the later chapters.

We focus on the joint diagonalization methods `SOBIdef` and `SOBIsym` for separating weakly stationary time series (Miettinen *et al.*, 2014, 2015). The latter has only recently been submitted in joint work. For both algorithms the (un-)mixing estimates are asymptotically normally distributed under mild conditions and one can determine limiting variances of the estimates when the time series length goes to infinity. We shortly review the algorithms, discuss further variants and compare all algorithms to widely-used BSS methods. In the second part we use the limiting distributions to perform probabilistic pattern identification: We want to decide whether close-to-zero entries of the mixing estimate are actually zero and, thus, determine the zero-pattern of the true mixing matrix. To achieve this we formulate hypothesis tests and model selection approaches using the limiting distributions of the estimates and we validate both in simulations.

The chapter is based on and in parts identical with the following publications

- **K. Illner**, J. Miettinen, C. Fuchs, S. Taskinen, K. Nordhausen, H. Oja, F.J. Theis (2015). Model selection using limiting distributions of second-order blind source separation algorithms. Signal Processing, 113, 95–103.

- J. Miettinen, **K. Illner**, K. Nordhausen, H. Oja, S. Taskinen, F.J. Theis. Separation of uncorrelated stationary time series using autocovariance matrices, *accepted for Journal of Time Series Analysis.*

5.1 Probabilistic pattern identification

In applications of BSS models, sometimes questions about the shape (or *pattern*) of the mixing matrix arise. In the following we outline two scenarios where sound decisions about the mixing pattern are needed.

In some applications we are interested in *active* source signals. In the several speakers problem, for example, we want to decide whether a single speaker's sound is recorded by a specific microphone or not. This is related to the question whether single entries in the mixing matrix are zero. If the jth source signal is not present in the ith observation then the entry (i, j) of the mixing matrix is zero. We are therefore interested in the zero-pattern of the mixing matrix, i. e. the positions of zero-entries. Nevertheless, BSS algorithms typically estimate a dense matrix where no entry is exactly equal to zero. Simple thresholding implies the cruical choice of an appropriate cutoff and does not appear convincing.

Experimental data are often collected under various experimental conditions like treatment vs. control or rest period vs. stimulus. In Hong & Calhoun (2004) and Beckmann & Smith (2004), for example, ICA methods are applied to human fMRI data (images of the brain activity) where the patients go through periods of rest and visual or audio-visual stimuli. Here, it is interesting to decide which source signals are *task-related* (McKeown *et al.*, 1998). Since the columns of the mixing matrix reflect the impact of the respective source signals among the different stimuli we want to test whether mixing columns are (statistically) significantly different from some pre-defined index vector.

To provide profound answers to these questions we use *limiting distributions* of the mixing estimates. Limiting distributions describe the asymptotic distribution of the mixing estimate when the time series length goes to infinity. In Ollila & Kim (2011), for example, such distributions are derived for the `fastICA` estimate of the independent component model and in Miettinen *et al.* (2012) for the `AMUSE` estimate of the BSS model for weakly stationary time series. In this chapter, we focus on the joint diagonalization (JD) problem from Section 4.3.1 with the algorithms `SOBIdef` and `SOBIsym` (Miettinen *et al.*, 2014, 2015). If we fix a mixing model (i. e. the mixing matrix and the time series model of the source signals) and consider several samples from the time series $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ we find that the mixing estimates approximately follow normal distributions (Figure 5.1). If $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ is an MA(∞)-process the limiting distributions can be calculated or approximated from the estimated source signals. With this additional information about the mixing estimates we provide a family of linear hypothesis tests to decide whether mixing columns are of some pre-defined shape and we provide model selection approaches to identify the most appropriate zero-pattern of the mixing matrix. If we are only interested in the zero-entries of the mixing matrix, another idea might be to add a penalty term to the JD problem (4.2). However, numerical optimization of such a penalized version fails in practice and we discuss the reasons in Appendix A.

In the following we first introduce the algorithms `SOBIdef` and `SOBIsym` and compare them to other BSS methods. We then move on to the limiting distributions of the (un-)mixing estimates and introduce and validate probabilistic pattern identification approaches.

5.2 Deflation-based and symmetric approaches

We consider the BSS model for weakly stationary time series introduced in Section 4.3.1. Thus, let $\{\boldsymbol{x}(t)\}_{t\in\mathbb{Z}}$ be a p-variate observed time series that is weakly stationary and zero-centered. We assume a noise-free linear mixing

$$\boldsymbol{x}(t) = A\,\boldsymbol{s}(t)\,, \quad t\in\mathbb{Z}, \tag{5.1}$$

where A is a deterministic full-rank $p\times p$ mixing matrix and $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ is a p-variate unobserved time series that fulfills assumptions (A1)-(A4) from page 48. Given data $X = \big(x(t)\big)_{t=1}^{T}$ let $M_1,\dots,M_k$ denote the sample autocovariances at distinct lags $\tau_1,\dots,\tau_K\in\mathbb{Z}$. M_0 is the sample autocovariance at lag zero. An unmixing estimate W can be determined by minimizing

$$f(W) = \sum_{k=1}^{K} \|\mathrm{diag}(WM_kW')\|_{\mathrm{F}}^2 = \sum_{j=1}^{p}\sum_{k=1}^{K}(w_j' M_k w_j)^2 \tag{5.2}$$

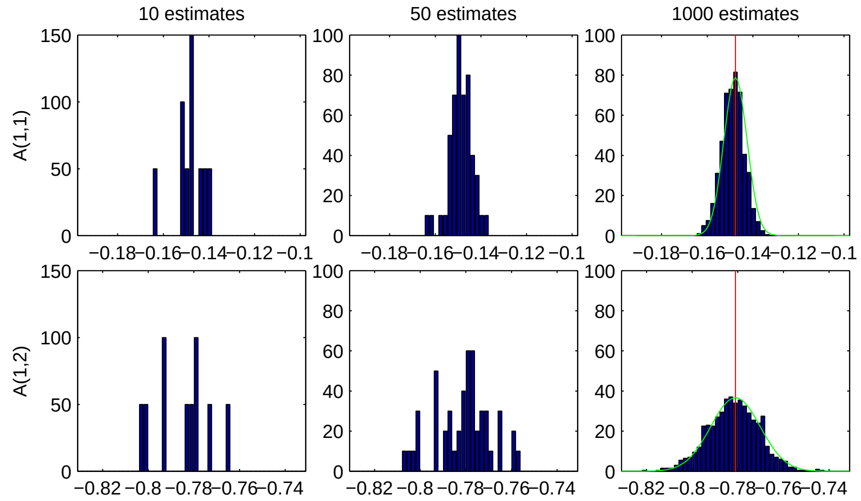

Figure 5.1: Asymptotic variance. Histrograms of the `SOBIsym` mixing entries $A(1,1)$ and $A(1,2)$ derived from 10, 50 and 1 000 simulations of an observed time series. The observations are generated from three AR(1)-processes with coefficients 0.9, −0.6, and 0.3 and fixed mixing matrix. The right plots also indicate the true mean (red) together with the (theoretically) asymptotic distribution (green).

under the constraint $WM_0W' = I_p$. This constraint maximization problem is equivalent to the constraint minimization (4.2) discussed in Section 4.3.1.

There are several ways to determine an optimal solution of (5.2). In the following we recall two conceptual approaches introduced by Miettinen *et al.* (2014, 2015). In the deflation-based approach the single rows of the unmixing matrix are estimated one after the other, in the symmetric approach, in contrast, all rows are estimated simultaneously. We derive the estimating equations and provide algorithms for both approaches.

5.2.1 Deflation-based SOBI

We consider the single rows of the unmixing matrix one after the other and successively maximize the inner sums of (5.2). Thus, at each step $j = 1, \ldots, p-1$ we maximize

$$f_j(w_j) = \sum_{k=1}^{K} (w_j' M_k w_j)^2 \tag{5.3}$$

under the constraint $w_j' M_0 w_j = \delta_{rj}$ for $r \leq j$. The constraint assures that for the final unmixing estimate it holds $W'M_0W = I_p$. To address this constraint optimization problem we use the Lagrange multiplier technique (Sun & Yuan, 2006). The Lagrangian function is given by

$$L(w_j, \lambda_j) = \sum_{k=1}^{K} (w_j' M_k w_j)^2 - \lambda_{jj}(w_j' M_0 w_j - 1) - \sum_{r=1}^{j-1} \lambda_{jr}(w_r' M_0 w_r - 0)$$

where $\lambda_j = (\lambda_{j1}, \ldots, \lambda_{jj})'$ are the Lagrangian multipliers for w_j. If $\hat{w}_j$ is an optimal solution of the constrained maximization problem (5.3) then $\hat{\lambda}_j \in \mathbb{R}^j$ exists such that the first-order partial derivatives of the Lagrangian function vanish at $(\hat{w}_j, \hat{\lambda}_j)$. The partial derivative of $L(w_j, \lambda_j)$ with respect to w_j is given by

$$\frac{\partial L}{\partial w_j} = 2 \sum_{k=1}^{K} (w_j' M_k w_j) M_k w_j - 2\lambda_{jj} M_0 w_j - 2 \sum_{r=1}^{j-1} \lambda_{jr} M_0 w_r \ .$$

If we set $\partial L / \partial w_j$ equal to zero, multiply both sides from the left with w_i' and summate the resulting equations for all $i < j$ we obtain the estimating equations for the deflation-based approach. For better readability we use the abbreviation

$$H(w_j) = \sum_{k=1}^{K} (w_j' M_k w_j) M_k w_j \ .$$

Proposition 5.1. *Let $W = (w_1, \ldots, w_p)'$ be an optimal BSS unmixing estimate derived by a deflation-based approach using autocovariance matrices M_0, $M_1, \ldots, M_K$. Then w_j for $j = 1, \ldots, p-1$ solves the estimating equation*

$$H(w_j) = M_0 (\sum_{r=1}^{j} w_r' w_r) H(w_j) \ .$$

The algorithm `SOBIdef` (Algorithm 4) determines an unmixing estimate according to the above estimating equations. In the algorithm we assume that the data is pre-whitened (i. e. $M_0 = I_p$) and the unmixing estimate is orthogonal. It can be easily seen that after convergence the estimating equations in Proposition 5.1 are fulfilled (Miettinen *et al.*, 2014, Appendix A).

The performance of `SOBIdef` depends on the extraction order of the rows, or equivalently, on permutations of the initial vectors $w_1, \ldots, w_p$. If we consider all $p!$ permutations we get up to $p!$ different unmixing estimates with different values for the maximization function (5.2). This behavior has been studied in detail in Miettinen

Algorithm 4: SOBIdef

```
input  : Autocovariances $M_1, \dots, M_K$ of pre-whitened data
output: Orthogonal unmixing estimate $W$

for $j = 1, \dots, p-1$ do
    inititalize $w_j$ optimal (cf. p.62)
    repeat
        $w_j^{(0)} = w_j$
        $w_j \leftarrow H(w_j)$
        $w_j \leftarrow (I_p - \sum_{r=1}^{j-1} w_r w_r')w_j$
        $w_j \leftarrow w_j / \|w_j\|_2$
    until $\|w_j - w_j^{(0)}\|_2 < \varepsilon$ or $\|w_j + w_j^{(0)}\|_2 < \varepsilon$;
```

et al. (2014). As *optimal initialization* the authors introduced the following: Among a set of random vectors w_j orthogonal to the first estimates $\hat{w}_1, \dots, \hat{w}_{j-1}$ one chooses w_j with highest value $f_j(w_j) = \sum_{k=1}^{K}(w_j' M_k w_j)^2$; this vector w_j is then used as initialization at step j. If the considered set of random vectors is sufficiently large, `SOBIdef` becomes independent of the initial guess. Furthermore, the unmixing estimate derived from `SOBIdef` is affine equivariant.

5.2.2 Symmetric SOBI

In the symmetric approach all rows of the unmixing matrix are estimated at once, such that the complete sum in (5.2) is maximized. The constraint $WM_0W' = I_p$ can be split into single constraints $w_i' M_0 w_j' = \delta_{ij}$ for $i, j = 1, \dots, p$. Since M_0 is symmetric it holds $w_i' M_0 w_j = w_j' M_0 w_i$ and we, thus, have $(p+1)p/2$ different constraints. The Lagrangian function is given by

$$L(W, \Lambda) = \sum_{j=1}^{p} \sum_{k=1}^{K} (w_j' M_k w_j)^2 - \sum_{j=1}^{p} \lambda_{jj}(w_j' M_0 w_j - 1) - \sum_{j=1}^{p} \sum_{r=1}^{j-1} \lambda_{jr}(w_r' M_0 w_j - 0) \ ,$$

where $\Lambda = (\lambda_{ij})$ is a symmetric matrix of Lagrange multipliers. The partial derivation with respect to w_j can be derived similarly to the deflation-based approach and for a solution W it holds

$$2H(w_j) = M_0 \Big(2\lambda_{jj} w_j + \sum_{r=1}^{j-1} \lambda_{rj} w_r + \sum_{r=j+1}^{p} \lambda_{jr} w_r \Big) \ .$$

As outlined in Miettinen *et al.* (2015), multiplying both sides from the left with w_i' yields $2\, w_i' H(w_j) = \lambda_{ij}$ for $i < j$ and $2\, w_i' H(w_j) = \lambda_{ji}$ for $i > j$. Since Λ is symmetric we obtain the following estimating equations for the symmetric approach.

Proposition 5.2. *Let $W = (w_1, \ldots, w_p)'$ be an optimal BSS unmixing estimate derived by a symmetric approach using autocovariance matrices M_0, $M_1, \ldots, M_K$. Then for $i, j = 1, \ldots, p$ it holds*

$$w_i' H(w_j) = w_j' H(w_i) \quad \textit{and} \quad w_i' M_0 w_j = \delta_{ij} \ .$$

To provide a BSS algorithm, we assume as before that the data is pre-whitened. `SOBIsym` (Algorithm 5) determines the unmixing estimate according to the above estimation equations, i. e. after convergence the estimating equations are fulfilled. In the algorithm, svd(W) denotes orthogonalization of W using the singular value decomposition (Section3.1).

Algorithm 5: SOBIsym

```
input  : Autocovariances $M_1, \ldots, M_K$ of pre-whitened data
output: Orthogonal unmixing estimate $W$
initialize $W \in \mathcal{O}(p)$ randomly
repeat
    $W^{(0)} = W$
    $W \leftarrow (H(w_1), \ldots, H(w_p))$
    $W \leftarrow \mathrm{svd}(W)$
until $\|W - W^{(0)}\|_{\mathrm{F}} < \varepsilon$;
```

5.2.3 Pairwise variants

The widely-used JD algorithm `SOBI` (Section 4.3.2) uses Jacobi rotations to update *pairs* of mixing columns. Accordingly, we now discuss two pairwise methods based on updates $H(\,.\,)$. For both methods we fix a sequence of pairs of indices (i, j) with $i < j$ and repeat updating the complete sequence until convergence. The algorithms assume pre-whitened data and only differ in the orthogonalization step. For `SOBIpairs1` (Algorithm 6) we use the singular values decomposition. Here, $\mathrm{svd}^{\dagger}(w_i, w_j)$ denotes orthogonalization of w_i and w_j using the singular values decomposition in the 2-dimensional subspace spanned by these vectors; after orthogonalization we perform back-transformation into the original space. For `SOBIpairs2` (Algorithm 7), in contrast, we use Gram-Schmidt orthogonalization.

Algorithm 6: SOBIpairs1

```
input  : Autocovariances $M_1, \dots, M_K$ of pre-whitened data
output: Orthogonal unmixing estimate $W$
initialize $W \in \mathcal{O}(p)$ randomly
repeat
    $W^{(0)} = W$
    for $i = 1, \dots, p$ do
        for $j = i + 1, \dots, p$ do
            $(w_i, w_j) \leftarrow (H(w_i), H(w_j))$
            $(w_i, w_j) \leftarrow \mathrm{svd}^{\dagger}(w_i, w_j)$
until $\|W - W^{(0)}\|_{\mathrm{F}} < \varepsilon$;
```

Algorithm 7: SOBIpairs2

```
input  : Autocovariances $M_1, \dots, M_K$ of pre-whitened data
output: Orthogonal unmixing estimate $W$
initialize $W \in \mathcal{O}(p)$ randomly
repeat
    $W^{(0)} = W$
    for $i = 1, \dots, p$ do
        for $j = i + 1, \dots, p$ do
            $(w_i, w_j) \leftarrow (H(w_i), H(w_j))$
            $w_i \leftarrow (I_p - \sum_{r \neq i, r \neq j} w_r w_r')w_i$
            $w_i \leftarrow w_i / \|w_i\|_2$
            $w_j \leftarrow (I_p - \sum_{r \neq j} w_r w_r')w_j$
            $w_j \leftarrow w_j / \|w_j\|_2$
until $\|W - W^{(0)}\|_{\mathrm{F}} < \varepsilon$;
```

The column updates of these pairwise methods are different from the Jacobi rotations in `SOBI`. However, we found in our simulations that after convergence `SOBIpairs1` and `SOBI` lead to exactly the same estimates. The estimates derived from `SOBIpairs2` are not affine equivariant. To assure affine equivariance for all proposed second-order blind identification variants one can use the `AMUSE` unmixing estimate as initialization. This yields a coherent initialization since the `AMUSE` estimate is affine equivariant.

For both pairwise methods we investigated a variety of modifications. Since the orthogonalization is only symmetric in the first version, we determined in the second version for each pair (i, j) whether the update of (w_i, w_j) or (w_j, w_i) leads to the better improvement on the estimate. Furthermore, we considered for both algorithms all pairs of indices (i, j) with $i \neq j$ (instead of $i < j$). We introduced inner convergence where we repeat the update and orthogonalization step for each pair until convergence. In contrast, we determined a best sequence of pairs (i. e. at each step we determined the pair whose update leads to the highest increase of (5.2)) and we update that sequence once. All these modifications did not lead to a truly better performance and a comparison is given in the next section.

5.3 Algorithm performance

We consider the following four time series models:

(*i*) *AR(4)-model*: three AR(4)-processes with coefficient vectors $(0.2, -0.5, 0.5, -0.4)$, $(0.3, 0.1, -0.7, 0.2)$, $(-0.2, 0.3, 0.1, 0.1)$ and normal innovations

(*ii*) *ARMA-model*: three ARMA-processes with AR-coefficient vectors $(-0.4, 0.2, -0.3)$, $(0.2, 0.5, -0.1)$, $(0.5, -0.1, 0.1)$ and MA-coefficient vectors $(0.1, -0.3, 0.2, 0.2, -0.1)$, $(0.7, 0.4, -0.3, 0.1, -0.2)$, $(-0.5, -0.4, -0.2, 0.5, 0.1)$ and normal innovations

(*iii*) *mixed model*: one AR(3)-, one AR(1)-, one MA(10)-process with coefficient vectors $(0.5, 0.1, 0.3)$, (0.7), $(0.4, 0.2, -0.1, -0.4, 0.3, 0.2, 0.6, 0.1, -0.3, -0.1)$ and normal innovations

(*iv*) *close-coefficient model*: three MA(3)-processes with coefficient vectors $(-0.25, 0.1, 0.5)$, $(-0.3, 0.1, 0.35)$, $(-0.2, 0.07, 0.4)$ and normal innovations

From all models we generate times-series of length T and scale each component to unit variance. We mix observations from these source signals using a random mixing matrix A with entries from $\mathcal{U}[-1, 1]$. For joint diagonalization we consider sample autocovariances at lags $\tau = 1, \ldots, K$. Note, that all algorithms are applied to the whitened data, and we need to transform the estimate to the original coordinate system afterwards. Further, all algorithms contain loops to update the single vectors or matrices iteratively. In the simulations we repeat these loops until the change in terms of the Frobenius norm (of the updated vector or matrix) is less than 10^{-6}, or a maximum number of 1 000 iterations is achieved. If there is no convergence after this maximum number of iterations we consider the run as non-convergent. All non-convergent runs are excluded from the performance results.

In Figure 5.2 A.-C. we generated data from models (*i*)-(*iii*) with a sample size of $T = 10\,000$ and used sample autocovariances at lags $\tau = 1, \ldots, 10$ for joint diagonalization. All algorithms are initialized with the identity matrix – except `SOBIdef` which has some intern randomization for correct row selection. An initialization with the `AMUSE` estimate only changes the performance of `SOBIpairs2` and this improved result is also added to the comparison. In these first abundant data situations all algorithms achieve comparable performances, where `SOBIdef` is slightly slower in terms of runtime. Surprisingly, `SOBI`, `SOBIsym` and `SOBIpairs1` lead to exactly the same estimates after convergence, and this was true for all considered data situations.

We then move on to more challenging data situations. In Figure 5.3 D. we consider model (*i*) but reduce the length of the time series to $T = 50$, in E. we add noise from $\mathcal{N}(0, 0.3)$ to each observation, and in F. we consider the close-coefficient model (*iv*). For the pairwise methods `SOBIpairs1+2` and the non-orthogonal `ACDC` the runtime increases and the estimates of `SOBIdef` and `SOBIpairs1` show a decrease in performance. A summary of the performances over all considered examples A.-F. is given in Table 5.1.

In the previous section we discussed several modifications of `SOBIpairs1+2` where we updated all pairs (w_i, w_j) for $i \neq j$, introduced inner convergence and determined a best sequence of pairs that we update once. For `SOBIpairs2` we also determined the better pair (w_i, w_j) or (w_j, w_i) to be updated. Figure 5.5 shows the estimation performance of all these modifications. In the plots we consider data from the AR(4)-model (*i*) and the close-coefficient model (*iv*). As before the time series length is fixed at $T = 10\,000$ and

Table 5.1: Summary of JD performances. Median MDI and median runtime together with the interquantile range IQR (i. e. the difference between the 75% and the 25% quantile) over all six time series examples from Figure 5.2 and 5.3. The last column indicates the number of non-convergent runs among a total of 60 000 runs; these non-convergent runs are omitted from the results.

	MDI	(IQR)	time (in ms)	(IQR)	#non-convergent
`SOBIdef`	0.058	(0.276)	0.130	(0.030)	1
`SOBIsym`	0.049	(0.207)	0.010	(0.020)	1
`SOBI`	0.049	(0.207)	0.000	(0.000)	16
`SOBIpairs1`	0.049	(0.207)	0.060	(0.080)	1
`SOBIpairs2`	0.064	(0.280)	0.050	(0.050)	166
`SOBIpairs2*`	0.058	(0.280)	0.030	(0.030)	295
`ACDC`	0.049	(0.210)	0.050	(0.110)	0

for joint diagonalization we use autocovariances at lags $\tau = 1, \dots, 10$. For `SOBIpairs1` all modifications did not lead to a better performance. For `SOBIpairs2` updating all pairs slightly increased to estimation performance on data from model (*iv*) but with the loss of a worse runtime. For both pairwise methods it was important to repeat updating the complete sequence of pairs several times – if we consider a best sequence that we update once the estimation performance strongly decreases.

For `SOBIdef`, `SOBIsym` and the original pairwise methods `SOBIpairs1+2` we further illustrate the convergence behavior. Figure 5.5 shows the MDI value among the single iterations for data from time series models (*i*) and (*iv*). In case of `SOBIdef` the trace contains jumps since we estimate the unmixing rows consecutively. Moreover, all algorithms – except `SOBIdef` – determine in-between estimates that outperform the final one.

To determine the impact of the number of sample autocovariances, we generated data from model (*i*) and increased $K = 3, \dots, 100$. While the performance only slightly changes we find a remarkable increase in runtime for `SOBIdef` and `ACDC` (Figure 5.6, upper plots). Similarly, the dimensionality of the problem mainly affects the runtime; the performance only slightly decreases. In dimensions $p > 10$ the estimation with

SOBIpairs1+2 was intractable and we focus on the remaining algorithms for increasing dimensions $p = 3, 5, \ldots, 50$ (Figure 5.6, lower plots).

In summary, we find that SOBI, SOBIsym and SOBIpairs1 yield the best overall performances and determine exactly the same mixing estimates after convergence. Although SOBI shows the lowest runtime among all algorithms, SOBIsym provides a reasonable alternative and has the advantage that the limiting distributions of the mixing estimates can be determined (Section 5.4). In comparison, ACDC is midrange and the algorithms SOBIdef, SOBIpairs2 and SOBIpairs2* show an inferior performance in terms of correctness of the estimate and runtime. The AMUSE initialization in SOBIpairs2* mainly affects the runtime and only slightly improves the estimation performance. Furthermore, all pairwise methods were intractrable in high dimensions ($p > 10$) and also the runtime of ACDC does not scale well with the dimensionality.

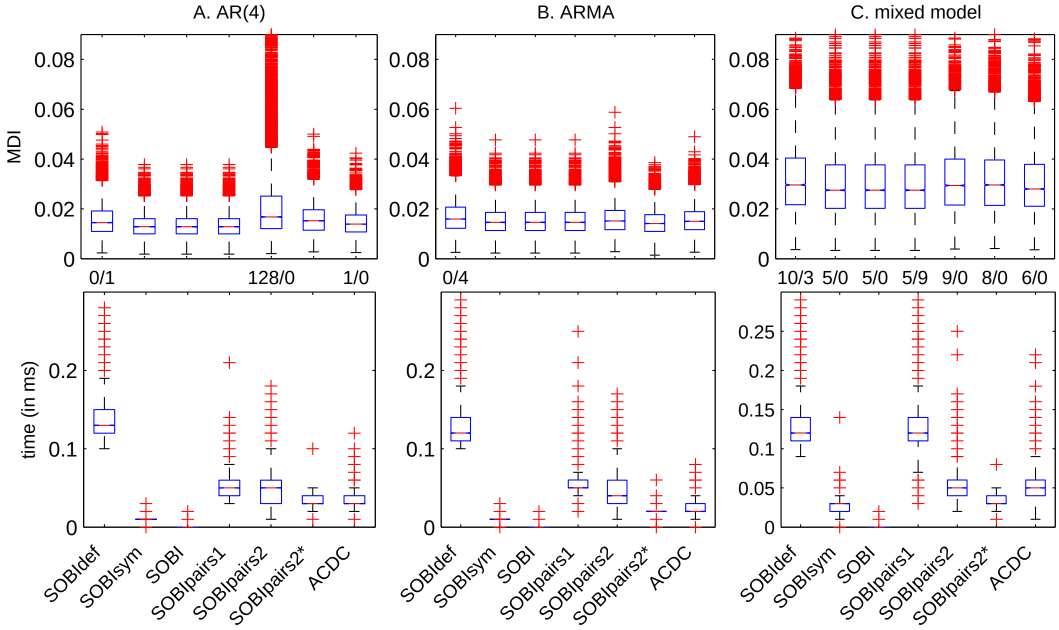

Figure 5.2: JD performance for different time series models. In A.-C. we show the median MDI (upper plots) and the median runtime (lower plots) of the mixing estimates over 10 000 repetitions. In A. the data was generated from the AR(4)-model (i), in B. from the ARMA-model (ii) , and in C. from the mixed model (iii). The sample size is fixed at $T = 10\,000$ and for joint diagonalization we consider autocovariances at lags $\tau = 1, \ldots, 10$. All algorithms are initialized with the identity matrix; for `SOBIpairs2` we also show the improved results with `AMUSE` initialization (*). Values between the plots indicate the counts of larger MDI/time values above the axis scaling.

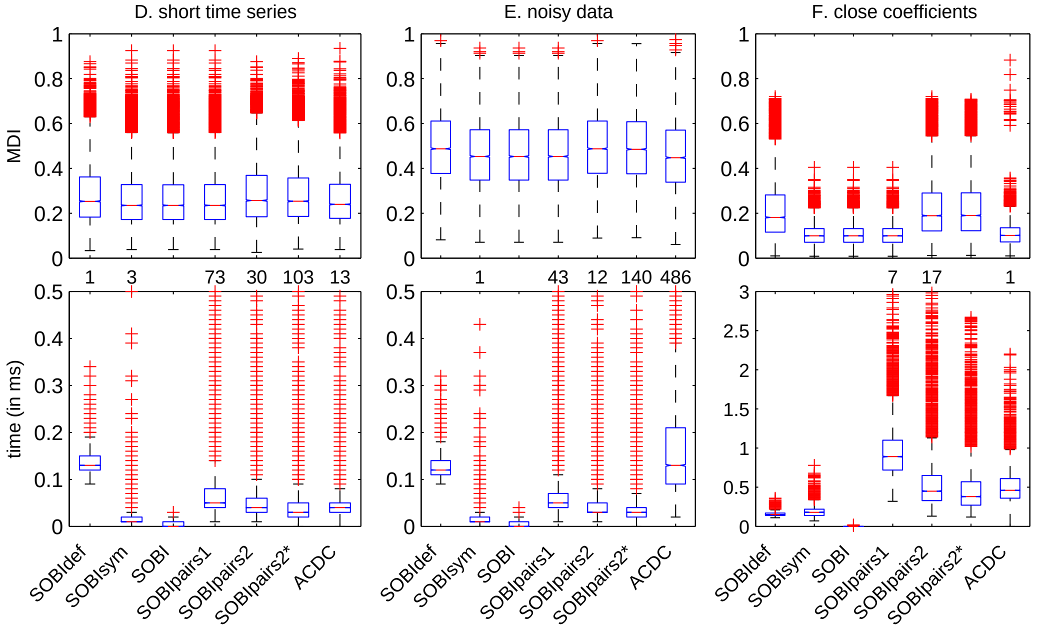

Figure 5.3: JD performance for different time series models. In D.-F. we consider challenging data situations and show the median MDI (upper plots) and the median runtime (lower plots) of the mixing estimates over 10 000 repetitions. In D. the data was generated from model (i) with a small sample size of $T = 50$, in E. the data was generated from model (i) including additive noise, and in F. we used data from the close-coefficient model (iv). If not stated differently, the sample size is fixed at $T = 10\,000$ and for joint diagonalization we consider autocovariances at lags $\tau = 1, \ldots, 10$. All algorithms are initialized with the identity matrix; for `SOBIpairs2` we also show the improved result with `AMUSE` initialization (*). Values between the plots indicate the counts of larger time values above the axis scaling.

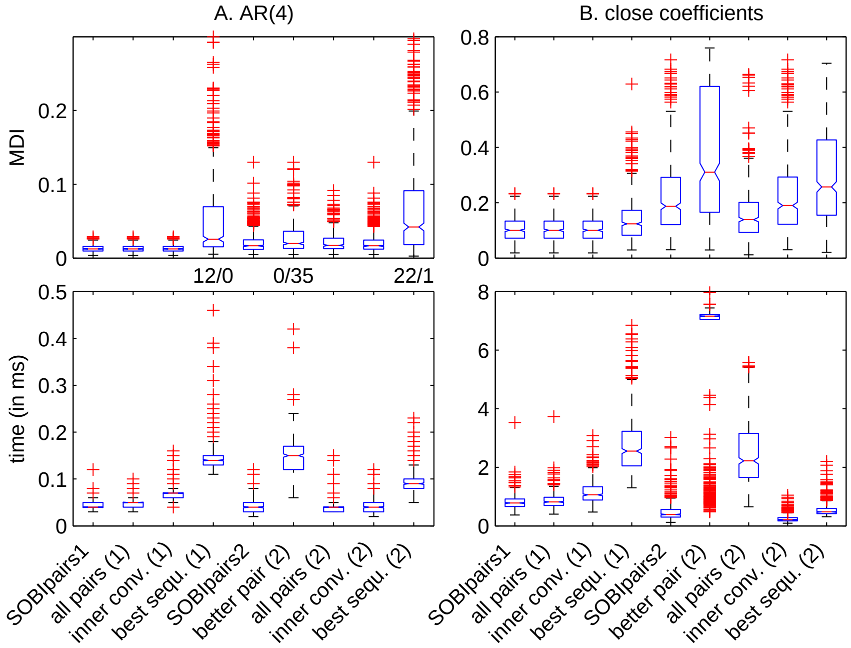

Figure 5.4: JD performance for pairwise variants. The plots show the median MDI (upper plots) and the median runtime (lower plots) of the mixing estimates derived from all pairwise variants over 500 repetitions. In A. the data was generated from the AR(4)-model (i) and in B. from the close-coefficient model (iv). The sample size is fixed at $T = 10\,000$ and for joint diagonalization we consider autocovariances at lags $\tau = 1, \ldots, 10$. All algorithms are initialized with the identity matrix. Values between the plots indicate the counts of larger MDI/time values above the axis scaling.

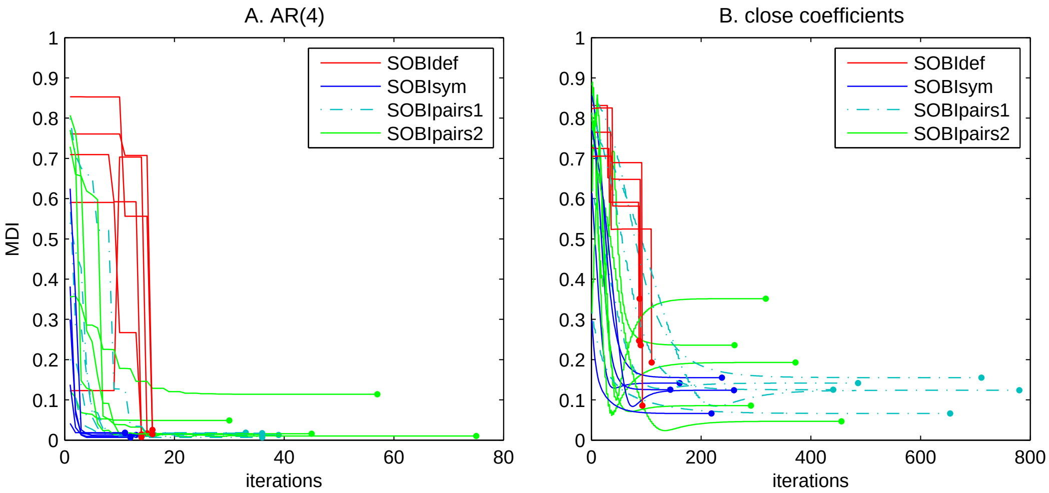

Figure 5.5: JD convergence. The plots show the MDI values among the single iterations of five estimation procedures of `SOBIdef`, `SOBIsym`, `SOBIpairs1` and `SOBIpairs2`. Since `SOBIdef` estimates the columns of the unmixing matrix one after the other we observe jumps in the MDI trace. In A. the data was generated from the AR(4)-model (i) and in B. from the close-coefficient model (iv) where the sample size is fixed at $T = 10\,000$ and for joint diagonalization we consider autocovariances at lags $\tau = 1, \ldots, 10$.

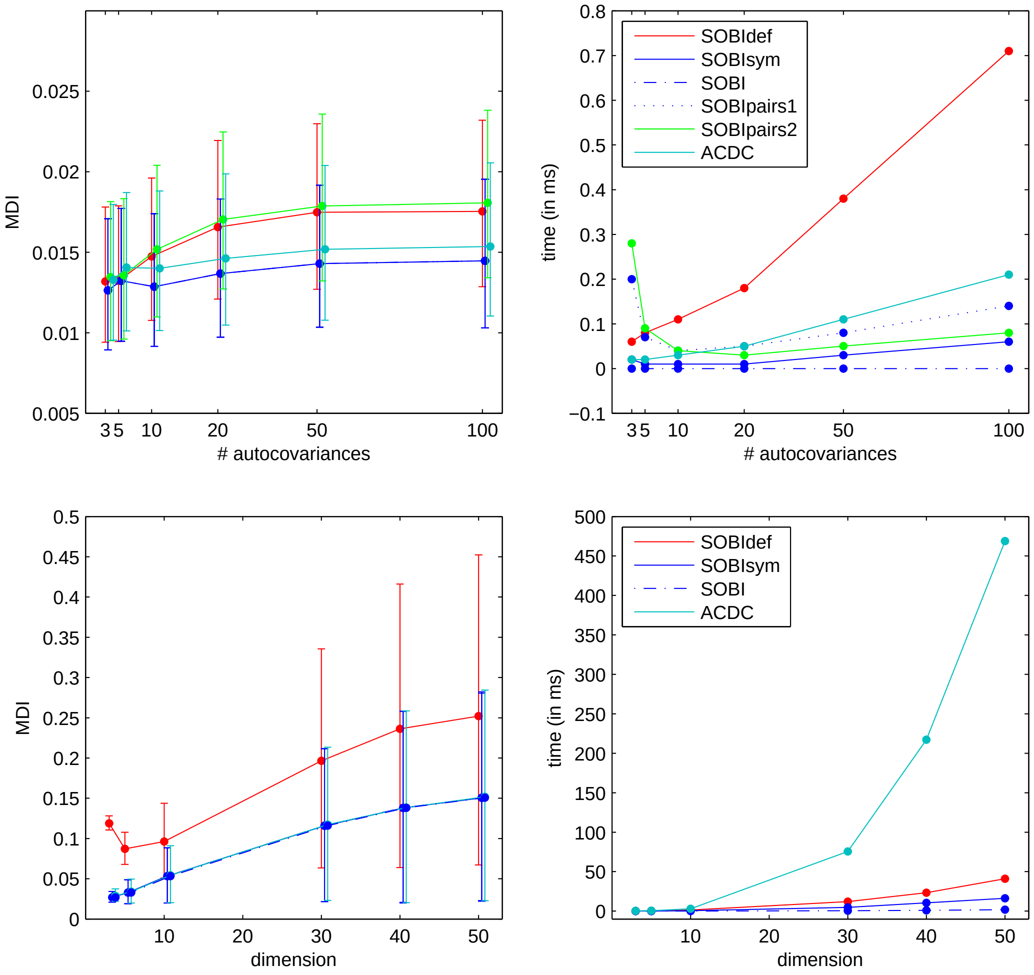

Figure 5.6: JD performance for increasing number of autocovariances and increasing dimension. Shown is the average MDI together with the upper and lower 25%-quantiles (left) and the average runtime (right) for the mixing estimates over 10 000 repetitions. *Upper plots:* The data was generated from the AR(4)-model (i) and has a length of $T = 10\,000$. For joint diagonalization we consider autocovariances at lags $\tau = 1, \ldots, K$ with increasing $K = 3, \ldots, 100$. *Lower plots:* The data was generated from AR(3)-models with random coefficients in each dimension and has a length of $T = 10\,000$. For joint diagonalization we consider autocovariances at lags $\tau = 1, \ldots, 10$ and the dimension of the data increases from $p = 3, \ldots, 50$. Note that `SOBIsym`, `SOBI` and `SOBIpairs1` yield the same mixing estimates.

5.4 Limiting distributions

The crucial point of strength about `SOBIdef` and `SOBIsym` is that we know the asymptotic distribution of the estimates. Under general multivariate time series assumptions, the (un-)mixing matrix estimates converge in probability to a true (un-)mixing matrix with limiting multivariate normal distribution. In the following we explain how the limiting variances can be calculated when the underlying time series model is given and how they can be estimated in case of real data. Further details and less restricting time series assumptions can be found in Miettinen *et al.* (2014, 2015).

In the BSS model (5.1) with assumptions (A1)-(A4) from page 48 the separation into mixing matrix and unobservable process is unique only up to sign-changing permutations. For the remaining part we need to clear this unidentifiability since we want to compare mixing estimate and true mixing matrix on the level of single entries. We therefore replace (A4) by the stronger assumption

(A4)* $\sum_k d_{k1}^2 > \ldots > \sum_k d_{kp}^2$, where $d_{k1}, \ldots, d_{kp}$ are the diagonal entries of the autocovariance of $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ at lag $k \in \mathbb{Z}$, and $a_j' 1_p \geq 0$ for all columns a_j of A, where 1_p denotes a p-dimensional vector of ones.

With this the separation problem becomes unique: From (A4) it follows that the mixing matrix A is unique up to sign-changing permutations. The first constraint in (A4)* determines the sorting of the components of $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ and thus the sorting of the columns of A. The second constraint fixed the indeterminacy of scaling the columns of A with ±1. To assure (A4)* for the `SOBIdef` and `SOBIsym` estimates we add a post-processing step and sort the rows of the original unmixing estimate such that $\sum_k (w_j M_k w_j')^2$ is decreasing for $j = 1, \ldots, p$. In addition we may need to multiply single columns by -1 to obtain positive column sums. For `SOBIdef` the discussed optimal initialization assures that the above sums are decreasing; we only need to consider the positive column sums.

We further specify the assumptions about the unobserved process. We assume a p-variate MA(∞)-process as introduced in Section 3.3 where

$$\boldsymbol{s}(t) = \sum_{j=0}^{\infty} \Psi_j \, \boldsymbol{\varepsilon}(t-j) \ , \quad \text{for } t \in \mathbb{Z} \ ,$$

with matrices $\Psi_j \in \mathbb{R}^{p\times p}$ and $\boldsymbol{\varepsilon}(t) \sim \mathcal{N}(0, I_p)$ for $t \in \mathbb{Z}$. The more general assumption that $\boldsymbol{\varepsilon}(t)$ are independent and identically distributed p-vectors is discussed in Miettinen *et al.* (2014, 2015). According to (A2), the process $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ has unit variance; thus, the matrices Ψ_j are diagonal and satisfy $\sum_{j=0}^{\infty} \Psi_j^2 = I_p$ for $j \in \mathbb{Z}$.

The variance in the mixing estimates mainly depends on the autocovariances of the unobservable process. Let therefore $\psi_j = (\psi_{j1}, \dots, \psi_{jp})'$ be the diagonal elements of Ψ_j and we define

$$F_k = \sum_{j=0}^{\infty} \psi_j \psi'_{j+k} \; , \quad \text{for } k \in \mathbb{Z} \; .$$

The diagonal elements of F_k are then the autocovariances of $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ at lag k. For better readability we define the $p \times p$ matrix $Q^{(lm)}$ for lags $l, m \in \mathbb{Z}$:

$$\begin{aligned} Q_{ii}^{(lm)} &= \sum_{s=-\infty}^{\infty} \big((F_{s+l})_{ii}(F_{s+m})_{ii} + (F_{s+l})_{ii}(F_{s-m})_{ii}\big) \; , \\ Q_{ij}^{(lm)} &= \sum_{s=-\infty}^{\infty} \frac{1}{2}\big((F_{s+l-m})_{ii}(F_s)_{jj} + (F_s)_{ii}(F_{s+l-m})_{jj}\big) \; . \end{aligned}$$

From now on we consider the mixing and unmixing estimates as p^2-variate random variables rather than concrete estimates. For visual distinction we use bold symbols $\hat{\boldsymbol{A}} = (\hat{\boldsymbol{a}}_{ij})$ and $\hat{\boldsymbol{W}} = (\hat{\boldsymbol{w}}_{ij})'$. We further use the notation vec(.) to denote the columnwise vectorization of a matrix. The following collects important results from Miettinen *et al.* (2014, 2015) in a form useful for their application in practice.

Theorem 5.1. *Let $\boldsymbol{x}(t) = I_p\, \boldsymbol{s}(t)$, $t \in \mathbb{Z}$, be an identity BSS model with mixing matrix I_p, and let $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ be a p-variate MA(∞)-process as defined above. We assume (A1)-(A3) and (A4)*, and for joint diagonalization we consider sample autocovariances at lags $k \in \{\tau_1, \dots, \tau_K\}$.*

If $\hat{\boldsymbol{W}}$ is the `SOBIdef` or the `SOBIsym` estimate for the unmixing matrix based on observations $x(1), \dots, x(T)$, then $\hat{\boldsymbol{W}} \to_{pr} I_p$ in probability and the limiting distribution of $\sqrt{T}\mathrm{vec}(\hat{\boldsymbol{W}} - I_p)$ for $T \to \infty$ is a p^2-variate normal distribution with a mean vector zero and a covariance matrix

$$\Sigma = \sum_{i,j} v_{ij}\, (\mathrm{e}_i\mathrm{e}_i' \otimes \mathrm{e}_j\mathrm{e}_j') + \sum_{i,j} c_{ij}\, (\mathrm{e}_i\mathrm{e}_j' \otimes \mathrm{e}_j\mathrm{e}_i') \; ,$$

where e_i *and* e_j *denote the canonical unit vectors with 1 at the ith or jth component, and the non-zero coefficients* v_{ij} *and* c_{ij} *for* $i, j = 1, \ldots, p$ *are as follows:*

a) If $\hat{\boldsymbol{W}}$ *is the* `SOBIdef` *estimate, then*

$$v_{ij} = \begin{cases} \dfrac{\sum_{l,m} d_{li} d_{mi}\, Q_{ij}^{(lm)} - 2\mu_{ii} \sum_k d_{ki}\, Q_{ij}^{(k0)} + \mu_{ii}^2\, Q_{ij}^{(00)}}{(\mu_{ii} - \mu_{ij})^2} & i < j\,, \\ \dfrac{\sum_{l,m} d_{lj} d_{mj}\, Q_{ij}^{(lm)} - 2\mu_{ji} \sum_k d_{kj}\, Q_{ij}^{(k0)} + \mu_{ji}^2\, Q_{ij}^{(00)}}{(\mu_{ji} - \mu_{jj})^2} & i > j\,, \\ 4^{-1}\, Q_{ii}^{(00)} & i = j\,, \end{cases}$$

$$c_{ij} = -v_{ij} + \frac{\mu_{ii}\, Q_{ij}^{(00)} - \sum_k d_{ki}\, Q_{ij}^{(k0)}}{\mu_{ii} - \mu_{ij}}\;, \quad \textit{for } i < j\,.$$

b) If $\hat{\boldsymbol{W}}$ *is the* `SOBIsym` *estimate, then*

$$v_{ij} = \begin{cases} \dfrac{\sum_{l,m} (d_{li} - d_{lj})(d_{mi} - \lambda_{mj})\, Q_{ij}^{(lm)}}{(\sum_k (d_{ki} - d_{kj})^2)^2} + \cdots & \\ \dfrac{-2\nu_{ij} \sum_k (d_{ki} - d_{kj})\, Q_{ij}^{(k0)} + \nu_{ij}^2\, Q_{ij}^{(00)}}{(\sum_k (d_{ki} - d_{kj})^2)^2} & i \neq j\,, \\ 4^{-1}\, Q_{ii}^{(00)} & i = j\,, \end{cases}$$

$$c_{ij} = v_{ij} + \frac{\nu_{ij}\, Q_{ij}^{(00)} - \sum_k (d_{ki} - d_{kj})\, Q_{ij}^{(k0)}}{\sum_k (d_{ki} - d_{kj})^2}\;, \quad \textit{for } i \neq j\,.$$

Here, $\mu_{ij} = \sum_k d_{ki} d_{kj}$ *and* $\nu_{ij} = \sum_k (d_{ki}^2 - d_{ki} d_{kj})$ *with* d_{ki}, d_{kj} *the ith and jth diagonal element of the autocovariance of* $\{\boldsymbol{s}(t)\}_{t \in \mathbb{Z}}$ *at lag k, and* $Q^{(lm)}$ *are as defined above.*

Under the assumptions of Theorem 5.1 the relation between the limiting distribution of mixing and unmixing estimate is given by

$$\sqrt{T}\mathrm{vec}(\hat{\boldsymbol{A}} - I_p) = -\sqrt{T}\mathrm{vec}(\hat{\boldsymbol{W}} - I_p) + o_{pr}(1)\;,$$

where $o_{pr}(.)$ is the order in probability. Thus, $\sqrt{T}\mathrm{vec}(\hat{\boldsymbol{A}} - I_p)$ and $-\sqrt{T}\mathrm{vec}(\hat{\boldsymbol{W}} - I_p)$ have the same limiting distribution in case of the identity mixing model. The coefficients v_{ij} and c_{ij} are the asymptotic (co-)variances of $\sqrt{T}\hat{\boldsymbol{w}}_{ij}$ for $T \to \infty$. More precisely, $v_{ij} = As\mathrm{Var}(\sqrt{T}\hat{\boldsymbol{w}}_{ij})$ and $c_{ij} = As\mathrm{Cov}(\sqrt{T}\hat{\boldsymbol{w}}_{ij}, \sqrt{T}\hat{\boldsymbol{w}}_{ji})$.

Since the `SOBIdef` and `SOBIsym` estimates are affine equivariant, one can further derive the limiting distributions of the mixing and unmixing estimate in case of the general model ($A \neq I_p$).

Corollary 5.1. *Let $\boldsymbol{x}(t) = A\boldsymbol{s}(t)$, $t \in \mathbb{Z}$, be a BSS model with an arbitrary mixing matrix $A = W^{-1}$ and $\{\boldsymbol{s}(t)\}_{t\in\mathbb{Z}}$ as before. Then $\sqrt{T}\text{vec}(\hat{\boldsymbol{A}} - A)$ and $\sqrt{T}\text{vec}(\hat{\boldsymbol{W}} - W)$ are asymptotically normally distributed with a mean vector zero and covariance matrices of the form*

$$\begin{aligned} ASV(\hat{\boldsymbol{A}}) &= \ (I_p \otimes A)\,\Sigma\,(I_p \otimes A')\ , \\ ASV^{\dagger}(\hat{\boldsymbol{W}}) &= (W' \otimes I_p)\,\Sigma\,(W \otimes I_p)\ , \end{aligned}$$

where $A \otimes B = (a_{ij}B)_{ij}$ denotes the Kronecker product of two matrices and Σ is given in Theorem 5.1.

Given a finite sample $x(1), \ldots, x(T)$, the deflation-based or symmetric mixing estimate $\hat{\boldsymbol{A}}$ is then approximately normally distributed as

$$\mathcal{N}(\text{vec}(A)\,, \frac{1}{T}ASV(\hat{\boldsymbol{A}}))\ ,$$

where the distribution depends on the true mixing matrix as well as the underlying source model. Since in general the true mixing matrix and the source model are unknown, we can approximate the distribution using the unmixing estimate $\hat{W}$ and the estimated source signals $\hat{W}x(1), \ldots, \hat{W}x(T)$. We determine the asymptotic variance using sample autocovariances of the source estimates and consider lags from a finite subset of $\mathbb{Z}$. In addition, we need to consider finite sums when calculating F_k, $Q^{(lm)}$ and Σ. We denote the resulting finite-sample variance by $\widehat{ASV}(\hat{\boldsymbol{A}})$. Functions to compute the asymptotic and the finite-sample variances can be found in the R-package 'BSSasymp' (Miettinen *et al.*, 2013).

5.5 Identification of the mixing pattern

Based on the limiting distributions from the previous section we now introduce probabilistic concepts to decide whether close-to-zero-entries of the mixing estimate are actually zero. With this, we can determine which source signals $\big(s_j(t)\big)_{t=1}^{T}$ actually contribute to an observation $\big(x_i(t)\big)_{t=1}^{T}$. We use the expression *zero-pattern* to refer to the positions of zero-entries in the mixing matrix.

Like in the previous section, we need to suppress the permutation indeterminacies of the BSS model. Thus, we assume the model variation (A4)* from page 74 together with the discussed post-processing of `SOBIdef` and `SOBIsym`.

5.5.1 Pattern identification via hypothesis tests

First, we investigate hypothesis tests on linear combinations of the mixing entries. Let therefore $x(1), \ldots, x(T)$ be observations of a BSS model with mixing matrix A. We consider a family of linear null hypotheses $H_0 : \Gamma \operatorname{vec}(A) = b$ and alternatives $H_1 : \Gamma \operatorname{vec}(A) \neq b$, where Γ is a $k \times p^2$ matrix and b is a k-vector. If the rows of Γ contain only one non-zero entry and this entry equals 1 we can test whether single entries of $\hat{A}$ are different from zero. Under the above null hypothesis and with $\hat{A}$ the `SOBIdef` or `SOBIsym` estimate we have

$$\sqrt{T}(\Gamma \operatorname{vec}(\hat{A}) - b) \ \to_d \ \mathcal{N}_k(0, \Gamma(ASV(\hat{\boldsymbol{A}}))\Gamma')$$

in distribution. This can be used in a test construction (still under H_0) as

$$M := (\Gamma \operatorname{vec}(\hat{A}) - b)' \left(\Gamma(\frac{1}{T}\widehat{ASV}(\hat{\boldsymbol{A}}))\Gamma'\right)^{-1}(\Gamma \operatorname{vec}(\hat{A}) - b) \ \to_d \ \chi^2_k \, .$$

Here, χ^2_k denotes the chi-squared distribution with k degrees of freedom (Section 3.2.4) and k is the number of linear equations in $\Gamma \operatorname{vec}(A) = b$. If M is larger than the upper αth quantile of χ^2_k, we reject H_0 with (asymptotic) probability of false alarm equal to α. Similar test statistics have been introduced by Ollila & Kim (2011) for the independent component model using the fastICA estimate.

To determine the zero-pattern of the mixing matrix we independently test $H_0 : a_{ij} = 0$ vs. $H_1 : a_{ij} \neq 0$ for all mixing entries. If H_0 is not rejected we assume that the corresponding entry is zero. This first approach for pattern identification is rather simplistic since the dependence structure of the mixing entries is not taken into account. In the simulations in Section 5.6 we refer to this approach as *h-test*.

5.5.2 Pattern identification via information criteria

We now move on to information criteria to select between different zero-patterns of the mixing matrix. Let A^h denote a $p \times p$ matrix with zero-entries at positions given in

$h \subseteq \{1, \ldots, p\} \times \{1, \ldots, p\}$. Thus, h is the zero-pattern of A^h. In the following we define the probabilistic model for such *reduced* mixing matrices.

According to Section 5.4, the mixing estimate $\hat{\boldsymbol{A}}$ is asymptotically normally distributed with the mean being the true mixing matrix and variance given by the limiting variance. Based on observations $x(1), \ldots, x(T)$ one can estimate the variance using the finite sample variance. A model for the (unknown) reduced mixing matrix A^h with zero-pattern h is given by

$$\mathcal{N}(\mathrm{vec}(A^h)\,, \frac{1}{T}\widehat{ASV}(\hat{\boldsymbol{A}}))\ , \tag{5.4}$$

where $\widehat{ASV}(\hat{\boldsymbol{A}})$ is the finite sample variance calculated from $x(1), \ldots, x(T)$. The number of model parameters equals the number of non-zero entries in A^h and for $h = \emptyset$ we get the full model with p^2 parameters. The observations are given by the mixing estimate $\hat{A}$ and with this the likelihood function is defined as $\ell(A^h) = \ln f(\hat{A}; A^h)$ where f is the density function with respect to (5.4). Finally, let $\hat{A}^h = \mathrm{argmax}_A\ \ell(A^h)$ denote the maximum likelihood estimate of the reduced mixing matrix.

To determine the most appropriate zero-pattern of the mixing matrix, we study a complete range of information criteria

$$IC(h) = -2\,\ell(\hat{A}^h) + kc\ , \tag{5.5}$$

where h is any zero-pattern, $k = p^2 - |h|$ denotes the number of model parameters, and c is some constant. For $c = 2$ the above equation yields the Akaike information criterion (AIC) and for $c = \ln(T)$ the equation yields the Bayesian information criterion (BIC) where T is the length of the observed time series (Section 3.5.4). With this, we identify the lowest value $IC(h)$ among all zero-patterns and the resulting h is the estimated zero-pattern for the mixing matrix. In the result part we refer to this approach as *AIC*, *BIC*, or *IC*.

In the above approach one needs to maximize the likelihood function for all zero-patterns h to determine the reduced estimate $\hat{A}^h$. To save computational time we invent a more heuristic variant: Since $\hat{A}$ itself is a mixing estimate, the non-zero entries of $\hat{A}^h$ will typically be close at the corresponding entries of $\hat{A}$. Thus, we might directly set entries of $\hat{A}$ to zero and leave all other entries unchanged. For a zero-pattern h we set $\hat{A}^h(i,j) = \hat{A}(i,j)$ for $(i,j) \notin h$ and zero otherwise. Note that this approach yields different estimates than before whenever $\widehat{ASV}(\hat{\boldsymbol{A}})$ contains non-zero off-diagonal entries. Using this modified estimate $\hat{A}^h$ we again determine the zero-pattern with the lowest value $IC(h)$. We refer to this approach as *AICmod*, *BICmod*, or *ICmod*.

5.6 Simulations

In the following we first validate the test statistics from Section 5.5.1 and investigate the impact of noise. We then provide a comparison of the three pattern identification methods IC, ICmod and h-test from Section 5.5.2. We consider mixing matrices with different numbers of zero-entries and different time series models.

We consider the AR(4)-model (i) from Section 5.3 and generate 3-dimensional data with mixing matrix $A = I_3$. For $j = 1, 2, 3$ we then test the hypothesis $H_0^{(j)} : a_j = \mathrm{e}_j$ vs. $H_1^{(j)} : a_j \neq \mathrm{e}_j$, where e_j is the canonical unit vector with 1 at the jth component. In addition we consider the complete mixing matrix and test $H_0^{(all)} : \mathrm{vec}(A) = \mathrm{vec}(I_3)$ vs. $H_1^{(all)} : \mathrm{vec}(A) \neq \mathrm{vec}(I_3)$. For all these tests we can easily define a matrix Γ with entries in $\{0, 1\}$ such that $\Gamma\,\mathrm{vec}(A) = \mathrm{e}_j$ for $j = 1, 2, 3$ or $\Gamma\,\mathrm{vec}(A) = \mathrm{vec}(I_3)$, respectively. In the first case the degrees of freedom of χ^2_k equal p in the latter p^2. Table 5.2 shows the percentage of (falsely) rejected null hypotheses at significance level 0.05 over 5 000 runs for a sample length of $T = 500, 1\,000, 10\,000$. We find a better identification for the first column of the estimate, but for $T = 10\,000$ all tests come close to the expected value of 5%.

We further address the question of how large entries of the mixing matrix must be such that they can be identified as non-zero. We therefore replace the previous mixing matrix, and we assume now that the first column of A is of the form $a_1 = (1, \varepsilon, 0)'$. All other entries are chosen randomly from the uniform distribution $\pm\mathcal{U}[0.1, 1.0]$ (U1) or $\pm\mathcal{U}[0.5, 1.0]$ (U2). Here, $\pm\mathcal{U}[v, w]$ for $0 < v < w$ denotes a uniform distribution with support $[-w, -v] \cup [v, w]$. We test $H_0^{(1)} : a_1 = \mathrm{e}_1$ vs. $H_1^{(1)} : a_1 \neq \mathrm{e}_1$ for increasing $\varepsilon = 0, 0.01, \ldots, 0.05$ and with 1 000 runs in each case. The percentage of correctly rejected null hypotheses increases with the value of ε and already at a value of $\varepsilon = 0.02$ we observe a rejection rate of 80% (Figure 5.7).

Table 5.2: Hypothesis tests on mixing entries. We generate data from the AR(4)-model (i) with a time series length of $T = 10\,000$. The true mixing matrix is chosen as identity matrix and for each column a_j $(j = 1, 2, 3)$ we test $H_0^{(j)} : a_j = e_j$ vs. $H_1^{(j)} : a_j \neq e_j$. In addition, we consider the complete mixing matrix and test $H_0^{(all)} : \text{vec}(A) = \text{vec}(I_3)$ vs. $H_1^{(all)} : \text{vec}(A) \neq \text{vec}(I_3)$. The table shows the percentage of (falsely) rejected null hypotheses at significance level 0.05 over 5 000 samples.

	`SOBIdef`				`SOBIsym`			
T	$H_0^{(1)}$	$H_0^{(2)}$	$H_0^{(3)}$	$H_0^{(all)}$	$H_0^{(1)}$	$H_0^{(2)}$	$H_0^{(3)}$	$H_0^{(all)}$
500	5.50	6.16	7.98	8.28	5.84	6.36	7.42	8.38
1 000	5.62	5.46	6.70	6.50	5.96	5.16	6.92	6.58
10 000	4.32	5.26	5.26	4.62	4.46	5.50	5.04	4.66

To perform model selection we generate data using the AR(4)-model (i) and fix the sample size at $T = 10\,000$. For the mixing matrix we consider the following four zero-patterns:

$$A_1 = \begin{pmatrix} * & 0 & * \\ * & * & * \\ * & * & * \end{pmatrix}, \quad A_2 = \begin{pmatrix} * & 0 & * \\ * & * & 0 \\ * & * & * \end{pmatrix}, \quad A_3 = \begin{pmatrix} * & 0 & 0 \\ * & * & * \\ * & * & * \end{pmatrix}, \quad A_4 = \begin{pmatrix} * & 0 & 0 \\ 0 & * & * \\ 0 & * & * \end{pmatrix},$$

where (*) denotes the non-zero entries. In case 1, for example, the second source signal has no impact on the first observation, and in case 3 the first observation depends only on the first source signal and so on. Let h_i denote the set of zero-entries in each case, i.e. $h_1 = \{(1,2)\}$, $h_2 = \{(1,2),(2,3)\}$, $h_3 = \{(1,2),(1,3)\}$ and $h_4 = \{(1,2),(1,3),(2,1),(3,1)\}$. The non-zero entries of the mixing matrix are chosen randomly from the uniform distribution $\pm\mathcal{U}[0.1, 1.0]$.

The `SOBIdef` and `SOBIsym` mixing estimates and their distributions are based on sample autocovariances at lags $\tau = 1, \dots, 10$. From these estimates we determine the most appropriate zero-patterns following the three approaches in Section 5.5. For evaluation we compare the zero-entries of the true mixing matrix to those of the estimated pattern. Figures 5.8 and 5.9 show the percentage of correctly determined patterns (filled areas) as well as the percentage of partly determined patterns (shaded areas), where not all or more zero-entries were detected. We considered 500 samples from time series models $(i) - (iv)$ with random mixing matrices. We found a crucial increase in performance if we used AIC/BIC with parameter maximization. In this case, BIC determined nearly

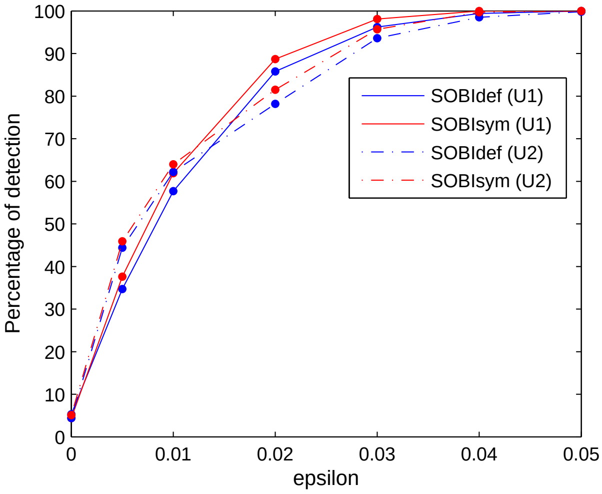

Figure 5.7: Hypothesis tests on noisy mixing entries. The first column of the mixing matrix is chosen as $a_1 = (1, \varepsilon, 0)$ for increasing ε, all other entries are randomly sampled from $U1 = \pm\mathcal{U}[0.1, 1.0]$ and $U2 = \pm\mathcal{U}[0.5, 1.0]$. The figure shows the percentage of rejected null hypothesis $H_0^{(1)} : a_1 = e_1$ at significance level 0.05. In case of $\varepsilon = 0$, this is a wrong decision, otherwise a correct one. We used 1 000 repetitions for each ε.

all zero-patterns for models $(i) - (iii)$ correctly. For the close-coefficient model (iv) the performance is significantly degrading.

We further investigated the impact of the information criterion constant c in (5.5) and the sample size T. Figures 5.10 and 5.11 show the percentage of correctly determined zero-patterns for `SOBIsym` estimate. The data was generated from time series models $(i) - (iv)$ with a sample length of $T = 500, 1\,000, 10\,000$. We increased $c = 1, \ldots, 100$ where the BIC is given for $c = 6.2$, 6.9, and 9.2 depending on T. We find that in nearly all settings IC clearly outperforms the modified IC. In comparison to the h-test the information criterion is only slightly better. The highest rates of correct zero detection are achieved for $c = \ln(T)$ (BIC). Furthermore, the performance depends on the underlying time series model; for the AR(4)-model (i) and the ARMA-model (ii) we find higher recovery rates compared to the mixed-model (iii) and the close coefficient model (iv).

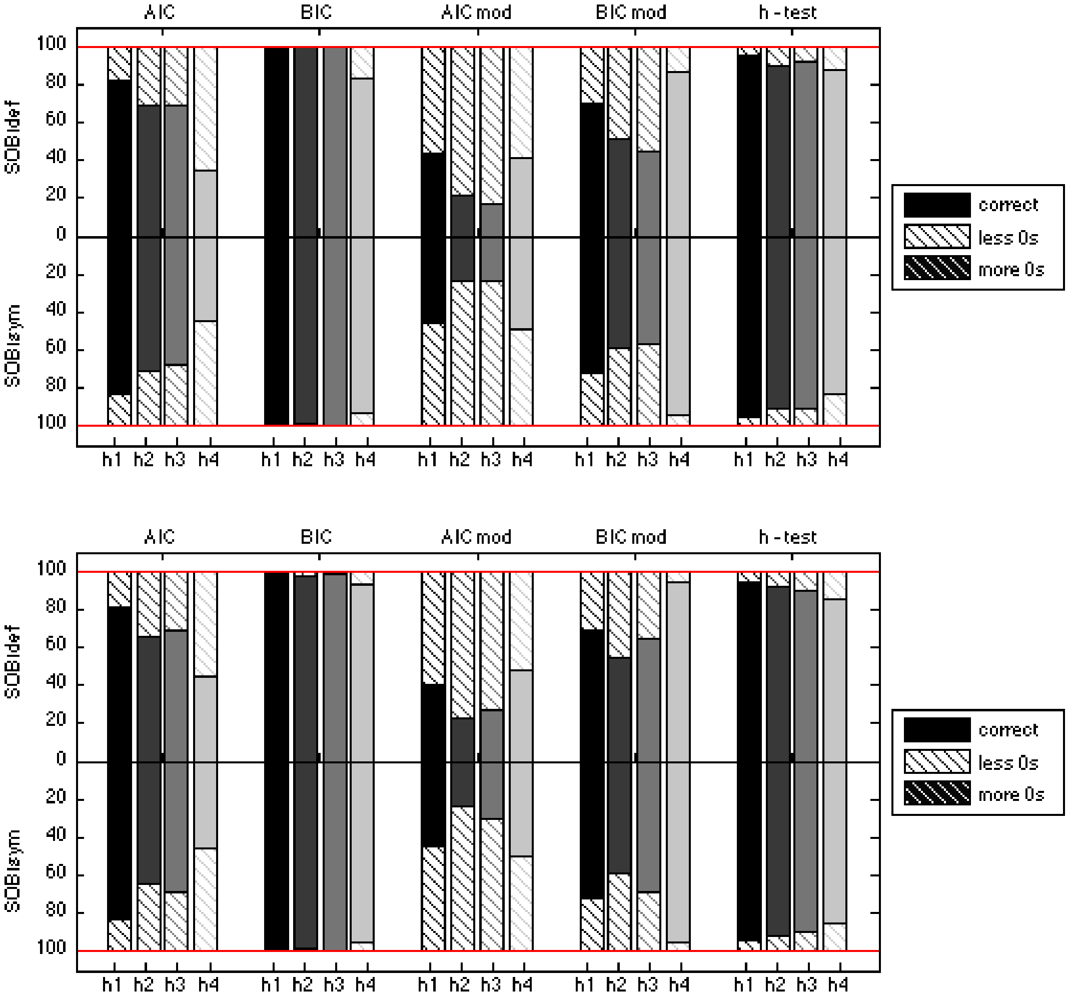

Figure 5.8: Pattern identification to determine zero-entries of the mixing matrix, models (i)-(ii). The data is generated using the AR(4)-model (*i*) (upper plot) and the ARMA-model (*ii*) (lower plot) with a time series length of $T = 10\,000$. The true mixing matrix contains zeros at positions $h_1 = \{(1,2)\}$, $h_2 = \{(1,2),(2,3)\}$, $h_3 = \{(1,2),(1,3)\}$ and $h_4 = \{(1,2),(1,3),(2,1),(3,1)\}$. We reconstruct these zero-patterns from the `SOBIdef` and `SOBIsym` mixing estimates using the selection methods AIC, BIC, AICmod, BICmod and h-test from Section 5.5. The percentages of correctly determined, under- and overdetermined patterns over 500 repetitions are shown as filled, shaded and dark shaded areas, respectively.

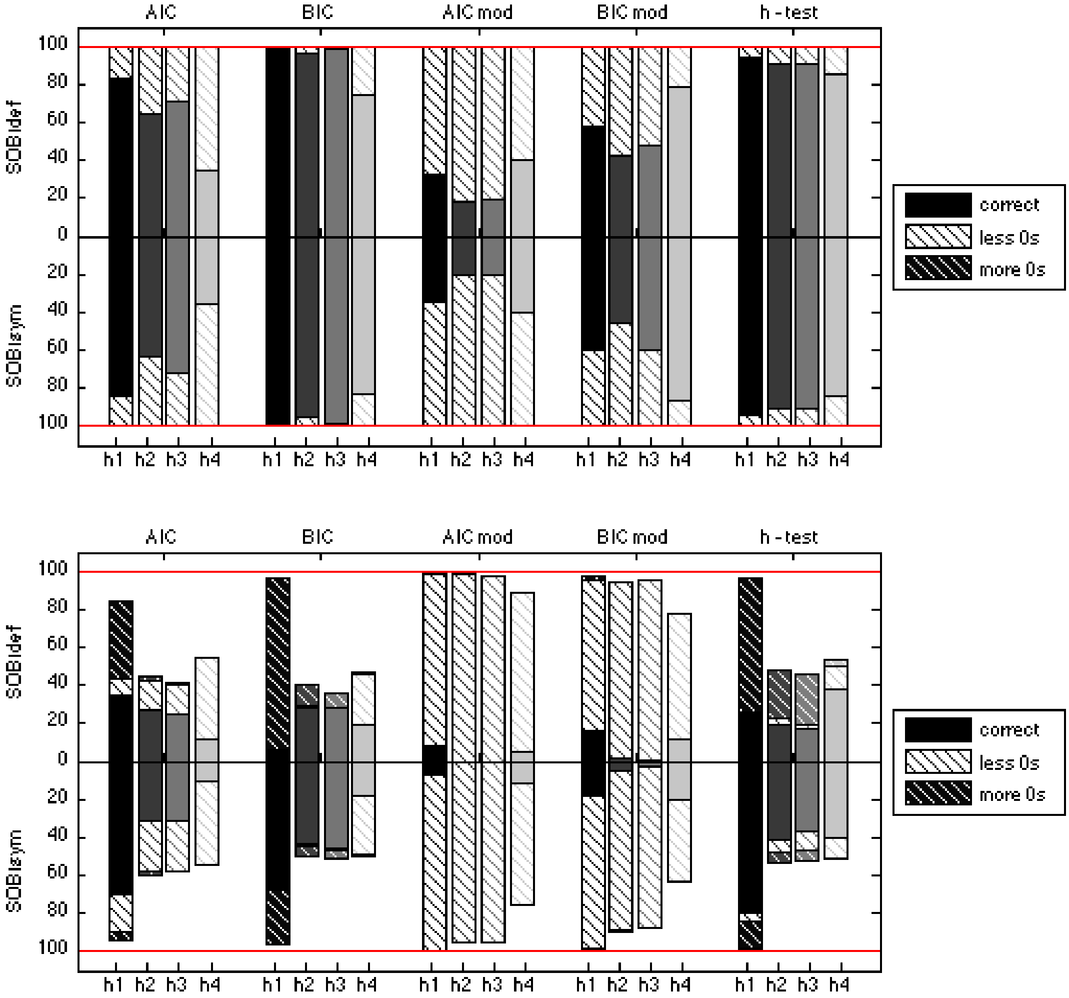

Figure 5.9: Pattern identification to determine zero-entries of the mixing matrix, models (iii)-(iv). The data is generated using the mixed model (*iii*) (upper plot) and the close-coefficient model (*iv*) (lower plot) with a time series length of $T = 10\,000$. The true mixing matrix contains zeros at positions $h_1 = \{(1,2)\}$, $h_2 = \{(1,2),(2,3)\}$, $h_3 = \{(1,2),(1,3)\}$ and $h_4 = \{(1,2),(1,3),(2,1),(3,1)\}$. We reconstruct these zero-patterns from the `SOBIdef` and `SOBIsym` mixing estimates using the selection methods AIC, BIC, AICmod, BICmod and h-test from Section 5.5. The percentages of correctly determined, under- and overdetermined patterns over 500 repetitions are shown as filled, shaded and dark shaded areas, respectively.

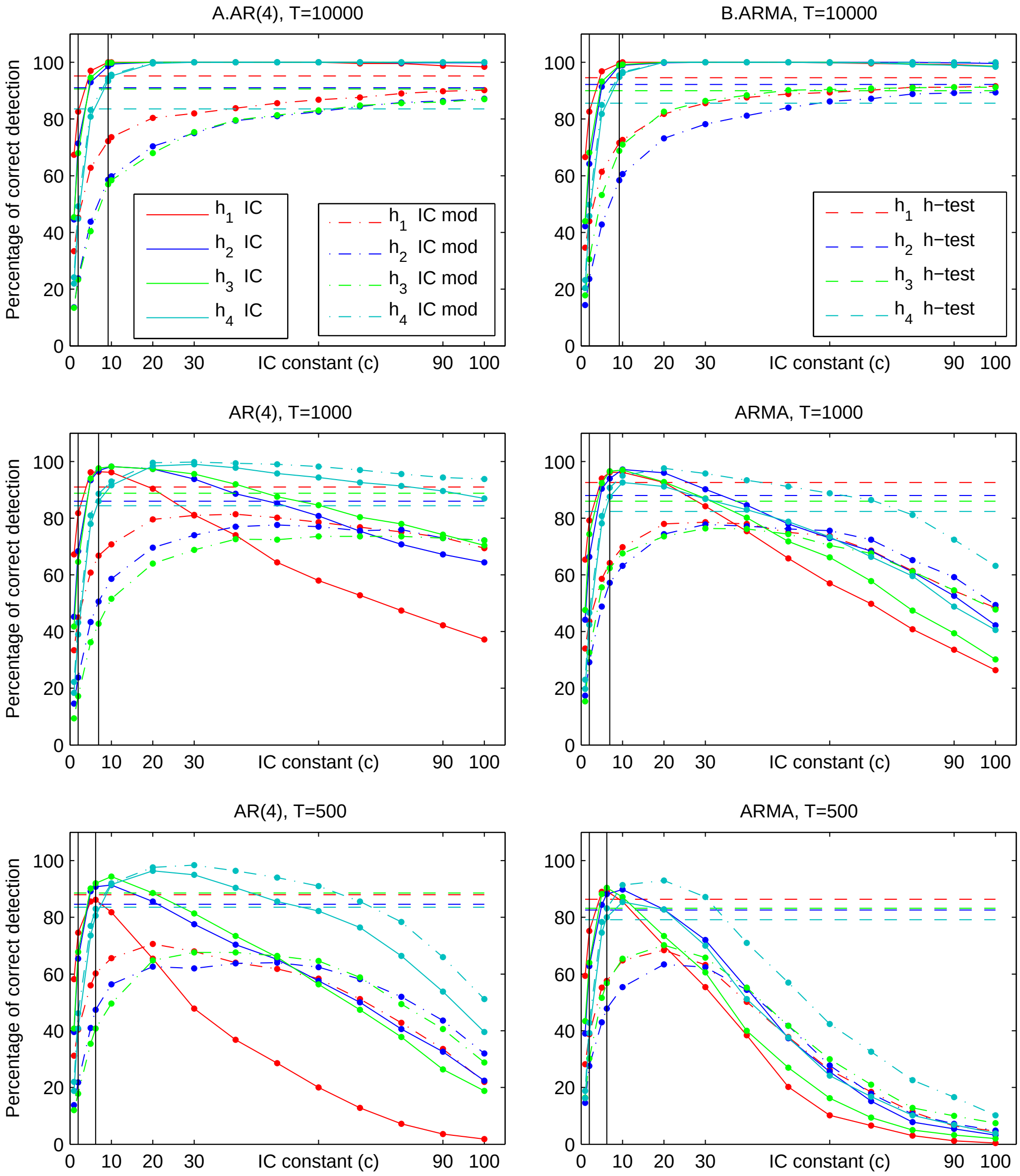

Figure 5.10: Performance of the information criterion for increasing constant c, models (i)-(ii). The data is generated using A. the AR(4)-model (i) and B. the ARMA-model (ii) with a time series length of $T = 10\,000$, $T = 1\,000$ and $T = 500$ in the single rows. The true mixing matrix contains zeros at positions $h_1 = \{(1,2)\}$, $h_2 = \{(1,2),(2,3)\}$, $h_3 = \{(1,2),(1,3)\}$ and $h_4 = \{(1,2),(1,3),(2,1),(3,1)\}$. We reconstruct these zero-patterns from the `SOBIsym` mixing estimates using the information criterion for increasing constant c (with and without maximization) and the h-test. The figure shows the percentage of correctly determined patterns over 500 repetitions. The black vertical lines indicate the constant values $c = 2$ (AIC) and $c = \ln(T)$ (BIC).

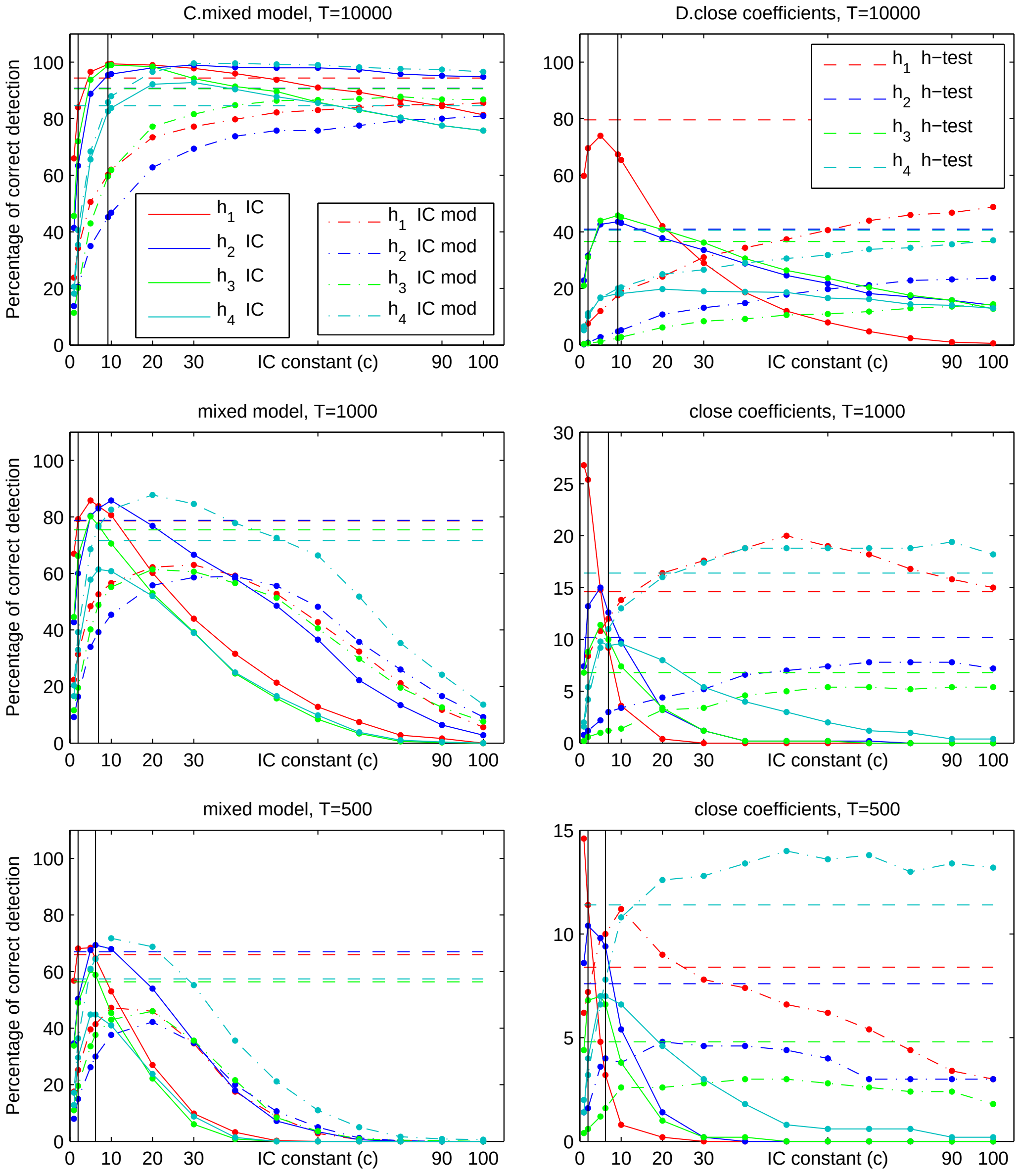

Figure 5.11: Performance of the information criterion for increasing constant c, models (iii)-(iv). The data is generated using C. the mixed model (iii) and D. the close-coefficient model (iv) with a time series length of $T = 10\,000$, $T = 1\,000$ and $T = 500$ in the single rows. The true mixing matrix contains zeros at positions $h_1 = \{(1,2)\}$, $h_2 = \{(1,2),(2,3)\}$, $h_3 = \{(1,2),(1,3)\}$ and $h_4 = \{(1,2),(1,3),(2,1),(3,1)\}$. We reconstruct these zero-patterns from the `SOBIsym` mixing estimates using the information criterion for increasing constant c (with and without maximization) and the h-test. The figure shows the percentage of correctly determined patterns over 500 repetitions. The black vertical lines indicate the constant values $c = 2$ (AIC) and $c = \ln(T)$ (BIC).

5.7 Conclusions

In this chapter we considered the second-order source separation methods `SOBIdef` and `SOBIsym`. Both algorithms provide limiting distributions of the (un-)mixing estimates and we developed solutions for scenarios where we can benefit from these distributions.

Firstly, we compared the relatively new methods to the well-established algorithms `SOBI` and `ACDC` and to additional pairwise variants. In simulations we considered different time series models, different numbers of autocovariances and high dimensions. The algorithms `SOBIsym` and `SOBI` provided exactly the same mixing estimates after convergence and showed the best overall performance. `SOBIdef` yielded an inferior performance with higher runtimes; `ACDC` performed midrange with a sharp increase in runtime in high dimensions. The pairwise methods could not improve the estimation performances and were intractable in higher dimensions.

In the second part we introduced the limiting distributions; based on these distributions we formulated a family of linear hypothesis tests to compare linear combinations of the mixing columns to pre-defined values. In particular, we tested the hypothesis that the mixing columns equal the canonical unit vectors, i. e. the mixing matrix is the identity matrix. Here, we considered 3×3 mixing matrices where the first column is of the form $(1, \varepsilon, 0)'$ with increasing disturbance $\varepsilon \geq 0$. We found correct rejection rates over 80% for $\varepsilon \geq 0.02$.

In addition, we proposed a model selection setup to determine the zero-pattern of mixing matrices. We used a maximum-likelihood approach and a variant with lower computational costs and derived *reduced* mixing estimates with zero-entries at specific positions. We then determined the most appropriate zero-pattern using model selection criteria with different penalties for complexity. In simulations we found very high recovery rates of the true zero-pattern using the BIC and hypothesis testing on single entries. The other proposed methods tended to underestimate the number of zero-entries.

In the following chapters we expand the linear structure of time series to more complex network structures, namely directed acyclic graphs. We introduce the probabilistic BSS method `emGrade` based on Baysian networks; in the end, model selection criteria provide again useful insights into the mixing process.

6

Separation of network data

In this chapter we introduce the probabilistic BSS method `emGrade`. With this, we make all gains from probabilistic BSS available for the application to network data. The term “network data” here means that the information of the signals propagates along the edges of an a priori known network. In our applications we focus on signaling data from systems biology where the variables are for example given by genes connected by a signaling pathway.

An analytical method to separate network data is `Grade` (Kowarsch *et al.*, 2010) introduced in Section 4.3.3. This method assumes stationary signals that are graph-decorrelated. We continue this idea and derive a statistically interpretable model. We first define a stationary Bayesian network to flexibly model the structure of the source signals and then expand the network to incorporate the linear mixing. The method not only determines estimates of the mixing and the underlying source signals, but also provides distributions of the estimates and makes model selection and Bayesian extensions applicable. In the last part we evaluate the performance of `emGrade` and demonstrate gains from the probabilistic modelling in simulations.

Parts of this chapter are already published in a journal and as a conference proceeding; the latter also provides an application of `emGrade` to the gene expression data that were originally used to evaluate `Grade`.

- **K. Illner**, C. Fuchs, F.J. Theis (2014). Bayesian blind source separation for data with network structure. Journal of Computational Biology, 21, 855–865.

- **K. Illner**, C. Fuchs, F.J. Theis (2012). Blind source separation using latent Gaussian graphical models. In Proc. 9th International Workshop on Computational Systems Biology (WCSB 2012), 34–37.

6.1 The idea of emGrade

Biological signaling pathways consist of variables that have an activating or inhibiting effect on the other variables. To mathematically model such a structure we use *Bayesian networks* (Section 3.4.3). Thus, we need to assume that the network is a directed acyclic graph, in particular, there are no self-loops. Since Bayesian networks fulfill the Markov property, we further assume that a variable given all its parents is independent of the previous variables. Although these are restricting assumptions, Bayesian networks have been successfully applied in the context of computational biology. In the literature, one finds several approaches to learn a Bayesian network from biological signaling data, i. e. one wants to determine the unknown network structure. The approaches mainly differ in the definition of the conditional distributions of a variable given its parents. In the following we shortly outline the various approaches.

Friedman *et al.* (2000) originally introduced Baysian networks for biological data. They considered a multinomial model (with discrete variables) and a linear Gaussian model to define the conditional distributions. Since the linear Gaussian model can only detect dependencies that are close to linear an extension to also non-linear dependencies is given by Imoto *et al.* (2002). Here, the dependence between a variable and its parents is captured by a non-parametric regression with B-splines as basis. Hartemink *et al.* (2001) and Pe'er *et al.* (2001) introduced edge-annotations for binary data: “+” indicates a positive influence of a factor and “$-$” a negative influence. In the Bayesian network these annotations are included as constraints on the parameters of the conditional distributions.

We follow the approach of Friedman *et al.* (2000) and assume conditional Gaussian distributions with linear dependencies. Instead of learning the network structure, we assume that the network is a priori known. We use it to perform a more appropriate source separation.

A further key assumption in our model is *stationarity* of the signals. We follow the approach of Kowarsch *et al.* (2010) who generalized the concept of stationarity from time series signals to network data (Section 4.3.3). Basically, they assumed that all variables share the same expectation and variance, and that the covariance between a variable and the (weighted) sum of its parents is constant over the network. To define the conditional distributions of the Bayesian network, we sharpen these assumptions and assume that the covariance between two adjacent variables is constant up to a known scaling factor. With this, the conditional distributions can be defined based on a single parameter. The scaling factors are given as edge weights in the network. In a gene regulatory network the weights refer, for example, to activation ($+1$) and inhibition (-1).

Finally, we make a similar *separation* assumption as provided by Kowarsch *et al.* (2010). We assume that the variance of each (multi-dimensional) variable as well as the covariance between adjacent variables is diagonal. Later, we extend the model and assume different network structures and thus different distributions for each source signal. In this case, we model 1-dimensional sources that are idenpendent.

In the context of BSS we assume that we observe a linear mixture of the source signals. If we consider the observations as a matrix, then one dimension represents the variables (e. g. genes) that are connected by a known network structure. The second dimension represents different time points or different experimental conditions. In addition, we might have repeated measurements where one variable is measured multiple times under the same condition. Or, on the other hand, some measurement components might be unreliable or missing. If the data consists, for example, of microarray measurements some probe sets on the chip can have extremely outlying values which prevent meaningful source identification. In the last part we therefore extend the basic model and account for repeated observations and missing components.

6.2 The source model

BSS approaches mainly differ in the *source model*, i. e. the assumptions on the source signals. In the following we define a source model in terms of a stationary Bayesian network. We illustrate the distribution assumptions on graphs and explain the connection to the `Grade` assumptions.

6.2.1 A stationary Bayesian network

Let $G_0 = (V, E)$ be a directed acyclic graph with $V = \{1, \ldots, N\}$ the set of nodes and $E \subset V \times V$ the set of edges. Let the nodes be ordered such that for each node i all parent nodes $\text{pa}(i) = \{j_1, \ldots, j_{n_i}\}$ have numbers lower than i. We further assume that the first $n_0 - 1$ nodes are the root nodes of the graph, i. e. $pa(i) = \emptyset$ for $i < n_0$.

The associated Bayesian network is given by a set of random variables $\boldsymbol{S} = \big(\boldsymbol{s}(i)\big)_{i=1}^{N}$ such that the joint distribution decomposes as

$$\text{p}(\boldsymbol{S}) = \prod_{i=n_0}^{N} \text{p}(\boldsymbol{s}(i) \mid \mathbf{Pa}(i)) \prod_{i=1}^{n_0-1} \text{p}(\boldsymbol{s}(i)) \ . \tag{6.1}$$

Here, $\mathbf{Pa}(i) = (\boldsymbol{s}(j_1)', \ldots, \boldsymbol{s}(j_{n_i})')'$ with $j_1 < \ldots < j_{n_i}$ denotes the vector of all random variables associated with the parent nodes of v_i. We further assume that $\big(\boldsymbol{s}(i)\big)_{i=1}^{N}$ are q-dimensional Gaussian random variables with state space $\mathbb{R}^q$.

Let now $\lambda_{ij} \in \mathbb{R}$ be weights assigned to the edges $(i, j) \in E$. We denote the resulting weighted graph by $G = (V, E, \Lambda)$. Let further $\boldsymbol{s}(i)$ and $\boldsymbol{s}(j)$ be random variables associated with adjacent nodes of the graph. We make the following stationarity (and scaling) assumptions:

(A1) $E[\boldsymbol{s}(i)] = 0_q$,
(A2) $\text{Cov}(\boldsymbol{s}(i), \boldsymbol{s}(i)) = I_q$,
(A3) $\text{Cov}(\boldsymbol{s}(i), \boldsymbol{s}(j)) = \lambda_{ij}\, D$.

The parameter D is constant over the network and we call it the *graph-delayed covariance* of the stationary Gaussian model. According to our actual purpose of source separation, we assume that

(A4) $D = \text{diag}(d_1, \ldots, d_q)$ is a diagonal matrix.

The larger an entry d_i the larger is the (absolute) value of the covariance between two adjacent random variables in the ith component.

With (A1)-(A3) and the Markov property of Bayesian networks all conditional distributions in (6.1) are uniquely defined. This can be seen by inductively constructing $\mathrm{p}(\boldsymbol{S})$: Let $\mathrm{p}(\boldsymbol{s}(1),\ldots,\boldsymbol{s}(j-1))$ be given. For the induction step we need to determine $\mathrm{Cov}(\boldsymbol{s}(i),\boldsymbol{s}(j))$ for all $i \leq j$. For $i=j$ and for $i<j$ with $(i,j)\in E$ this covariance is given in (A2)-(A3). For $i<j$ with $(i,j)\notin E$ we use the variance theorem and condition on all parent nodes of $\boldsymbol{s}(j)$. We thus have

$$\begin{aligned}\mathrm{Cov}\left(\boldsymbol{s}(i),\boldsymbol{s}(j)\right) &= \mathrm{Cov}\left(E[\boldsymbol{s}(i)\,|\,\mathbf{Pa}(j)]\,,\,E[\boldsymbol{s}(j)\,|\,\mathbf{Pa}(j)]\right)\\ &\quad + E[\mathrm{Cov}\left(\boldsymbol{s}(i),\boldsymbol{s}(j)\,|\,\mathbf{Pa}(j)\right)]\,,\end{aligned}$$

where all conditional distributions are known and the second term equals zero due to the Markov property. With this, we finally get $\mathrm{p}(\boldsymbol{S})$ and we denote the conditional distributions in (6.1) by

$$\boldsymbol{s}(i)\,|\,\mathbf{Pa}(i) \sim \begin{cases} \mathcal{N}(0_q, I_q) & \mathit{if}\ \mathbf{Pa}(i)=\emptyset\,,\\ \mathcal{N}(\nu_D(i)\,\mathbf{Pa}(i),\Sigma_D(i)) & \mathit{otherwise}\,,\end{cases} \tag{6.2}$$

where $\nu_D(i)\in\mathbb{R}^{q\times q\,n_i}$ and $\Sigma_D(i)\in\mathbb{R}^{q\times q}$ only depend on D. Since D is diagonal it holds that $\Sigma_D(i)$ is also diagonal and $\nu_D(i)$ consists of blocks of diagonal matrices. Dependent on the weighted graph G one can determine an interval $I^G\subseteq\mathbb{R}$ such that all covariance matrices $\Sigma_D(i)$ are positive definite for $d_1,\ldots,d_q\in I^G$.

In case of a line-graph ($1\to 2\to 3\to \ldots$) with all edge weights identically and non-zero, the definition of a stationary network directly corresponds to (weakly) stationarity of time series introduced in Section 3.3. If the edge weights equal 1 then the autocovariances of the *process* $\left(\boldsymbol{s}(i)\right)_{i=1}^{N}$ are given by $\mathrm{Cov}(\boldsymbol{s}(i),\boldsymbol{s}(i-k)) = D^{|k|}$ where $k\in\mathbb{Z}$ denotes the lag. The conditional distributions of the corresponding Bayesian network are of the form $\mathrm{p}(\boldsymbol{s}(i)\mid\boldsymbol{s}(i-1))\sim\mathcal{N}(D\boldsymbol{s}(i-1),1-D^2)$.

In summary, the Gaussian distributions defined in (6.2) provide a source model $\mathcal{M}(G,q)$ to separate signals from networks. Here, G is a weighted directed acyclic graph and q is the dimension of the random variables. We assume that the graph structure – including the edge weights – is a priori known. Thus, and because of the stationarity assumptions (A1)-(A3), the model is parameterized by the graph-delayed covariance D. This restriction to a single parameter makes parameter inference feasible and the maximum likelihood approach naturally yields an estimate $\hat{D}^{\mathrm{ML}}$. In case of a line-graph with all weights equal to 1 the separation assumptions directly correspond to a weakly stationary time series with uncorrelated components at any lag.

6.2.2 Illustration on graphs

To illustrate the covariance structure of the stationary Gaussian model we introduce three graph models. The first two graphs are related to gene regulatory networks. The last graph ist related to time series signal for comparison.

(CC) Cell-cycle: The estimated network for the cell-cycle pathway based on gene expression data (Imoto *et al.*, 2002). The network consists of 81 nodes and 84 edges.

(TF) Transcription factors: Three hub nodes and each directly signals on a subset of nodes.

(LL) Line signals: Similar to time series we define a network that consists of two line signals sharing the middle part, and one separated line signal.

Figure 6.1 illustrates these networks together with the associated covariance structure where we randomly assigned weights ±1 to the edges. These weights might indicate activating (+1) or inhibiting (-1) effect of a variable on its targets. Note, that our model theoretically allows any edge weights $\lambda_{ij} \in \mathbb{R}$.

Since the proposed source model is based on *directed* graphs we further investigate the importance of the edge directions on the covariance structure. We therefore randomly reverse 50% of the edges in the graph models (CC), (TF) and (LL). Note, that the resulting graph needs to be acyclic. Figure 6.2 shows the covariance structure for the new graphs where the edge weights are as before. If edges are reversed the absolute value of the covariance between random variables can change or even become zero. Nevertheless, the sign of the covariance remains unchanged.

6.2.3 Review: Graph-delayed covariance

In this part we compare our new definition of a graph-delayed covariance to the original definition from Kowarsch *et al.* (2010). Let as before D denote the graph-delayed covariance introduced in Section 6.2.1 based on a weighted directed graph $G = (V, E, \Lambda)$.

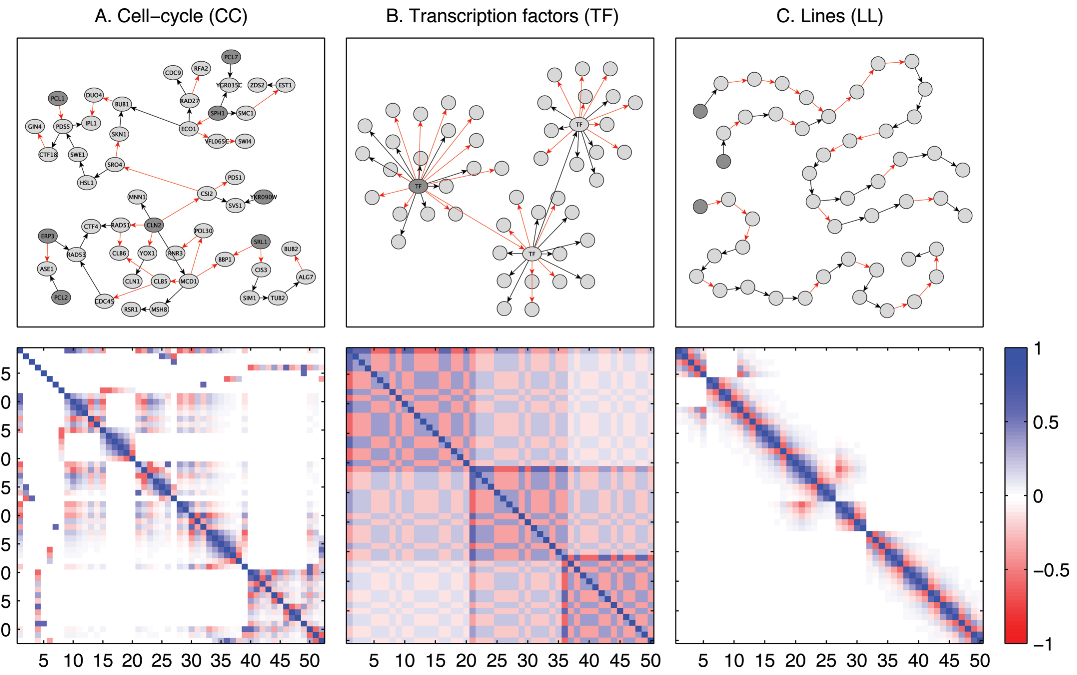

Figure 6.1: Graph models and covariance structure. The upper graphics illustrate a connected subnetwork of cell-cycle (CC), the transcription factors (TF), and lines (LL). Darker nodes indicate root nodes and we randomly assigned edge weights with values +1 (black) and −1 (red). The lower graphics show the covariance structure associated with each graph model for 1-dimensional random variables. The graph-delayed covariance was set to $d = 0.6$.

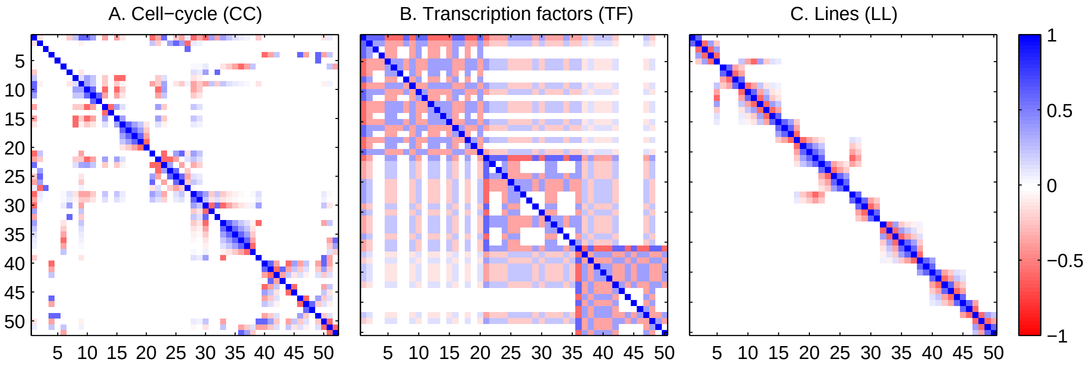

Figure 6.2: Covariance structure for graph models with reversed edges. We randomly reversed 50% of the edges in graph models (TF), (LL) and (CC) and show the covariance structure. All edge weights are retained and the graph-delayed covariance is set to $d = 0.6$ like before.

Let now $\kappa_{ij} \in \mathbb{R}$ be new weights for the same graph structure and let $\boldsymbol{S} = \big(\boldsymbol{s}(i)\big)_{i=1}^{N}$ be random variables. Kowarsch *et al.* originally introduced the graph-delayed covariance as

$$D^{\mathrm{Pa}} = \mathrm{Cov}\big(\sum_{i \in pa(j)} \kappa_{ij} \boldsymbol{s}(i), \boldsymbol{s}(j)\big) , \tag{6.3}$$

and they assumed that it is independent of the index j. This directly relates to our definition of D. According to assumption (A3), both definitions coincide for $\lambda_{ij} = 1/(\,|pa(j)|\,\kappa_{ij})$. Note that our model specifications (Markov property and stationarity) uniquely define the distribution of the variables $\boldsymbol{S}$. This does not hold for the assumptions in (Kowarsch *et al.*, 2010). Thus, our approach is on the one hand more specific but also more restricting in terms of application.

To estimate D^{Pa} from samples $s(1), \ldots, s(N)$ Kowarsch *et al.* further introduced

$$\hat{D}^{\mathrm{Pa}} = \frac{1}{N - n_0 - 1} \sum_{j=n_0}^{N} \sum_{i \in pa(j)} \kappa_{ij} s(i)\, s(j)' , \tag{6.4}$$

where we assume that the first $n_0 - 1$ nodes are the root nodes of the graph. In our model, we have more detailed information about the covariance between adjacent variables. We therefore additionally consider the following refined edge-based estimate

$$\hat{D}^{\mathrm{E}} = \frac{1}{|E| - 1} \sum_{(i,j) \in E} \frac{1}{\lambda_{ij}} s(i)\, s(j)' . \tag{6.5}$$

Diagonalization of the estimates $\hat{D}^{\mathrm{Pa}}$ and $\hat{D}^{\mathrm{E}}$ yields non-probabilistic algorithms to separate network data (Section 4.3.3). For distinction we use the notation `Grade(Pa)` and `Grade(E)`, where the former is the original method from Kowarsch *et al.* (2010). In the next section we introduce the probabilistic algorithm `emGrade` that is based on the maximum likelihood estimate $\hat{D}^{\mathrm{ML}}$. A comparison of all algorithms is provided in Section 6.4.2.

6.3 The BSS method emGrade

In BSS we assume that we observe a linear mixture of the actual signals of interest. The aim is to estimate the mixing together with the underlying signals. In the following we derive a new BSS method for network data and decribe the unobserved (latent) signals in terms of the source model from the previous section. We provide an expectation-maximization scheme to estimate the parameters. In the last part we extend the basic model and account for repeated observations, allow missing components and use different network structures to model each source signal separately.

6.3.1 The linear mixing

We consider the following mixing model: $\boldsymbol{X} = \big(\boldsymbol{x}(i)\big)_{i=1}^{N}$ are observed Gaussian variables with state space $\mathbb{R}^p$, and we assume latent Gaussian variables $\boldsymbol{S} = \big(\boldsymbol{s}(i)\big)_{i=1}^{N}$ with state space $\mathbb{R}^q$ ($p \geq q$), such that each variable $\boldsymbol{x}(i)$ is a linear mixture of the components of the latent variable $\boldsymbol{s}(i)$:

$$\boldsymbol{x}(i) = A\,\boldsymbol{s}(i) + \mu + \boldsymbol{\varepsilon}(i)\,, \quad i = 1, \ldots, N\,. \tag{6.6}$$

Here, $\boldsymbol{\varepsilon}(i) \in \mathbb{R}^p$ is additive i.i.d. noise $\boldsymbol{\varepsilon}(i) \sim \mathcal{N}(0_p, \sigma^2 I_p)$ and independent of the latent variables. $A \in \mathbb{R}^{p \times q}$ denotes the mixing matrix and $\mu \in \mathbb{R}^p$ is a constant mean vector for all $\boldsymbol{x}(i)$. We refer to the components of the latent variables as sources, i.e. for $k = 1, \ldots, q$ we have a source $\boldsymbol{s}_k = \big(\boldsymbol{s}_k(i)\big)_{i=1}^{N}$.

We now expand the Bayesian network from Section 3.4.3. Let $\boldsymbol{S}$ be latent variables and the dependence is given by a weighted graph $G = (V, E, \Lambda)$ as before. We additionally introduce observed variables $\boldsymbol{X}$, where $\boldsymbol{x}(i) = A\boldsymbol{s}(i) + \mu + \boldsymbol{\varepsilon}(i)$ for all i. The joint distribution of $\boldsymbol{X}$ and $\boldsymbol{S}$ then decomposes as

$$\mathrm{p}(\boldsymbol{X}, \boldsymbol{S}) = \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{x}(i)\,|\,\boldsymbol{s}(i)) \prod_{i=n_0}^{N} \mathrm{p}(\boldsymbol{s}(i)\,|\,\mathbf{Pa}(i)) \prod_{i=1}^{n_0-1} \mathrm{p}(\boldsymbol{s}(i))\,, \tag{6.7}$$

where $\boldsymbol{x}(i) \mid \boldsymbol{s}(i) \sim \mathcal{N}(A\boldsymbol{s}(i) + \mu, \sigma^2 I_p)$ directly follows from the linear mixing and $\boldsymbol{s}(i) \mid \mathbf{Pa}(i)$ is given in (6.2). A graphical representation of the latent variable model is given in Figure 6.3a. The model parameters are given by $\theta = (A, \mu, \sigma^2, D)$ and we have $k = dq + d + q + 1$ single parameter entries.

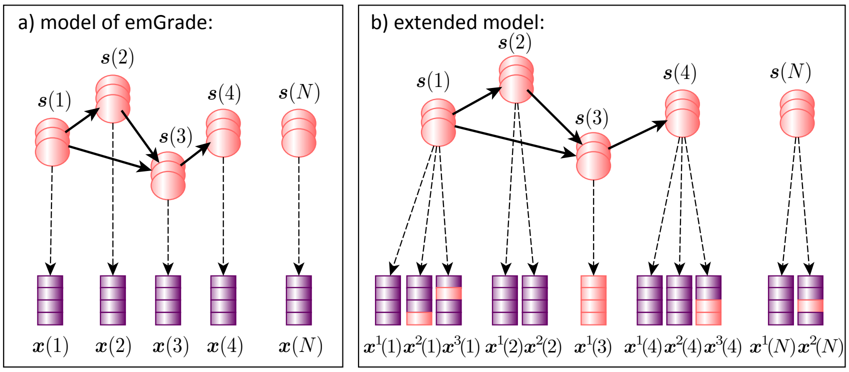

Figure 6.3: Graphical representation of emGrade. Figure a) shows the basic model with one observed variable $\boldsymbol{x}(i) = A\boldsymbol{s}(i) + \mu + \boldsymbol{\varepsilon}(i)$ for $i = 1, \ldots, N$. In b) we take into account multiple observations and missing components. In both figures the observed part is shown in purple and the latent part in red.

6.3.2 Parameter inference using expectation maximization

The unknown components of our model are the parameters θ and the latent variables $\boldsymbol{S}$, and we are interested in both. A widely-used approach for latent variable models is expectation maximization where parameters and latent variables are updated alternately (Section 3.5.3). Each update improves the data log-likelihood $\ell(\theta; \boldsymbol{X}) = \ln \mathrm{p}(\boldsymbol{X} \mid \theta)$ and in the following we explain these updates for the `emGrade` model.

For expectation maximization we consider the complete data log-likelihood which is given by $\ell_c(\theta; \boldsymbol{X}, \boldsymbol{S}) = \ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S} \mid \theta)$. Let $E_{S|X,\theta}[\,.\,]$ denote the conditional expectation of the latent variables $\boldsymbol{S}$ given the observable variables $\boldsymbol{X}$ and parameters θ (Section 3.2.3). The expectation of the complete data log-likelihood is then given by

$$E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S} \mid \theta)\big] = E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X} \mid \boldsymbol{S}, A, \mu, \sigma^2)\big] + E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{S} \mid D)\big] \; .$$

For better readability we define a combined parameter $A_\mu = (A, \mu) \in \mathbb{R}^{p \times (q+1)}$ and enlarge the latent variables by a constant component, i. e. $\boldsymbol{s}_*(i) = (\boldsymbol{s}(i)', 1)'$ for $i =$

$1, \ldots, N$. We then have:

$$\begin{aligned}
E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X} \mid \boldsymbol{S}, A, \mu, \sigma^2)\big] &= -\frac{Np}{2}\ln(2\pi) - \frac{Np}{2}\ln(\sigma^2) \\
&\quad - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\Big[\mathrm{Tr}\left(E_{S|X,\theta}[\boldsymbol{x}(i)\boldsymbol{x}(i)']\right) \\
&\quad - 2\,\mathrm{Tr}\left(E_{S|X,\theta}[\boldsymbol{s}_*(i)\boldsymbol{x}(i)']\,A_\mu\right) \\
&\quad + \mathrm{Tr}\left(E_{S|X,\theta}[\boldsymbol{s}_*(i)\boldsymbol{s}_*(i)']\,A_\mu' A_\mu\right)\Big] \qquad (6.8)
\end{aligned}$$

$$\begin{aligned}
E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{S} \mid D)\big] &= -\frac{Nq}{2}\ln(2\pi) - \sum_{i=n_0}^{N}\ln\!\big(\det(\Sigma_D(i))\big) \\
&\quad - \frac{1}{2}\sum_{i=n_0}^{N}\Big[\mathrm{Tr}\left(E_{S|X,\theta}[\boldsymbol{s}(i)\boldsymbol{s}(i)']\Sigma_D(i)^{-1}\right) \\
&\quad - 2\,\mathrm{Tr}\left(E_{S|X,\theta}[\boldsymbol{s}(i)\mathbf{Pa}(i)']\,\Sigma_D(i)^{-1}\nu_D(i)\right) \\
&\quad + \mathrm{Tr}\left(E_{S|X,\theta}[\mathbf{Pa}(i)\mathbf{Pa}(i)']\,\nu_D(i)'\Sigma_D(i)^{-1}\nu_D(i)\right)\Big] \\
&\quad - \frac{1}{2}\sum_{i=1}^{n_0-1} E_{S|X,\theta}\big[\boldsymbol{s}(i)\boldsymbol{s}(i)'\big] \qquad (6.9)
\end{aligned}$$

The EM-algorithm consists of two steps that are repeated alternately until convergence. In the *E-step* we determine the posterior distribution of the latent variables which yields the expectations in (6.8) and (6.9). Here, we use the property $E[\boldsymbol{z}_1\boldsymbol{z}_2'] = \mathrm{Cov}(\boldsymbol{z}_1, \boldsymbol{z}_2) + E[\boldsymbol{z}_1]\,E[\boldsymbol{z}_2]'$ for random variables $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ from Section 3.2.3. If $\boldsymbol{z}_2$ or both variables are observed the left-hand side equals $E[\boldsymbol{z}_1]\,\boldsymbol{z}_2'$ and $\boldsymbol{z}_1\boldsymbol{z}_2'$, respectively. To get these posterior estimates we use the junction tree algorithm implemented in the Bayes net toolbox for MATLAB (Murphy *et al.*, 2001). This algorithm performs marginalization in general graphs; details can be found in Neapolitan *et al.* (2004). In the *M-step* we then maximize $E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S})\big]$ with respect to the parameters. Due to our specific stationarity assumptions the Bayes net toolbox is not applicable for parameter maximization. Let $\mathrm{Esx} = \sum_{i=1}^{N} E_{S|X,\theta}\big[\boldsymbol{s}_*(i)\boldsymbol{x}(i)'\big]$ and we define Ess and Exx accordingly. The parameter updates for A, μ (in form of A_μ) and σ^2 can be derived directly from (6.8) and we get

$$A_\mu = \big(\mathrm{Esx}\big)'\big(\mathrm{Ess}\big)^{-1} \qquad (6.10)$$

$$\sigma^2 = \frac{1}{Np}\big[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}\,A_\mu) + \mathrm{Tr}(\mathrm{Ess}\,A_\mu' A_\mu)\big] \qquad (6.11)$$

A detailed derivation including the partial derivaties of $E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X},\boldsymbol{S})\big]$ is provided in Appendix B. The parameter D occurs as different rational terms ν_D and Σ_D in (6.9). For all source models $\mathcal{M}(G,.)$ with G a weighted directed acyclic graph one can theoretically derive formulas to update D. In our simulations we consider different graph models regarding structure, number of nodes, and egde weights, and we use numerical optimizers to obtain D. In our MATLAB implementation, for example, we use the function `fmincon` for constraint optimization problems. The search space for all diagonal entries of D is given by the interval I^G and since D – as well as $\nu_D(i)$ and $\Sigma_D(i)$ – is (block-)diagonal we can maximize $E_{S|X,\theta}[\ln \mathrm{p}(\boldsymbol{S})]$ with respect to each component of D separately.

The proposed expectation maximization scheme for the linear mixing model from Section 6.3.1 provides a method to separate network data. Similarly to the separation assumptions of `Grade` (graph-decorrelation algorithm) we assume a diagonal matrix D, and we therefore call the new algorithm `emGrade` (expectation maximization graph-decorrelation algorithm). A pseudo-code implementation is given in Algorithm 8.

Since the EM scheme maximizes the log-likelihood function it is natural to define the stop criterion in terms of a threshold for the increase of $\ell(\theta;X)$. Moreover, if we want to perform model selection we are only interested in the log-likelihood and not in the concrete parameter estimates. Thus, we consider $|\ell(\theta;X)-\ell(\theta^{(0)};X)|<10^{-6}$ as stop criterion when calculating AIC or BIC values. Abbi *et al.* (2008), in contrast, defined the stop criterion as a sufficient small change in the single parameter estimates. Convergence of single parameters usually requires more EM-iterations (when the same threshold is considered) and the parameters can still vary even when the change of the log-likelihood is negligible. To assure to have approximately reached the final parameter estimates we use the strong requirement

$$\forall i \quad |\,\theta_i-\theta_i^{(0)}|<10^{-8}\ ,$$

whenever we are interested in the model parameters. Here, θ_i denotes the single entries of the parameters A, μ, σ^2 and D. Furthermore, we use 10 000 iterations at maximum.

After convergence, the estimates of parameters and source signals are independent of the parameter initialization (Figure 6.4). However, for a fast convergence of the expectation maximization procedure a good choice of parameter initialization is cruical. A beneficial

Algorithm 8: emGrade

```
input  : Observations $X$, weighted directed acyclic graph $G$
output: Source signals $S$, parameter estimates $A$, $\mu$, $\sigma^2$, $D$
% initialization
Determine interval $I^G = [d_{\min}, d_{\max}]$
Build BN w.r.t. $G$, set observed variables equal to $X$
Initialize parameters $\theta = (A, \mu, \sigma^2, D)$ randomly where $D = \mathrm{diag}(d_1, \ldots, d_q)$
repeat
    % E-step
    $\theta^{(0)} = \theta$
    Determine $\nu_D(i)$ and $\Sigma_D(i)$ in $\boldsymbol{s}(i) \,|\, \mathbf{Pa}(i) \sim \mathcal{N}(\nu_D(i)\, \mathbf{Pa}(i), \Sigma_D(i))$
    Update BN with parameters $A, \mu, \sigma^2, \nu_D, \Sigma_D$
    Infer from posterior distribution (junction-tree algorithm): $E[\boldsymbol{s}(i)]$, $\mathrm{Cov}(\boldsymbol{s}(i), \boldsymbol{s}(i))$, $\mathrm{Cov}(\boldsymbol{s}(i), \mathbf{Pa}(i))$, and $\mathrm{Cov}(\mathbf{Pa}(i), \mathbf{Pa}(i))$
    % M-step
    $\mathrm{Ess} = \sum_{i=1}^N \mathrm{Cov}(\boldsymbol{s}_*(i), \boldsymbol{s}_*(i)) + E[\boldsymbol{s}_*(i)]E[\boldsymbol{s}_*(i)]'$
    $\mathrm{Esx} = \sum_{i=1}^N \mathrm{Cov}(\boldsymbol{s}_*(i), \boldsymbol{s}_*(i)) + E[\boldsymbol{s}_*(i)]x(i)'$
    $\mathrm{Exx} = \sum_{i=1}^N \mathrm{Cov}(\boldsymbol{s}_*(i), \boldsymbol{s}_*(i)) + x(i)x(i)'$
    $A_\mu = \big(\mathrm{Esx}\big)'\big(\mathrm{Ess}\big)^{-1}$
    $\sigma^2 = \frac{1}{Np}\big[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}\, A_\mu)$
    for $i = 1, \ldots, q$ do
        $d_i =$ fmincon(@$d_i$, $E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{S} \mid D)\big]$) with $E_{S|X,\theta}[\,.\,]$ a function of $d_i \in I^G$
    Update $\theta = (A, \mu, \sigma^2, D)$
until $\big|\ell(\theta; X) - \ell(\theta^{(0)}, X)\big| < \varepsilon$;
```

initialization is, for example, to use the mean of the observations for μ and the `Grade` estimates for A and D. For the noise variance we use a low initialization $\sigma^2 = 0.1$.

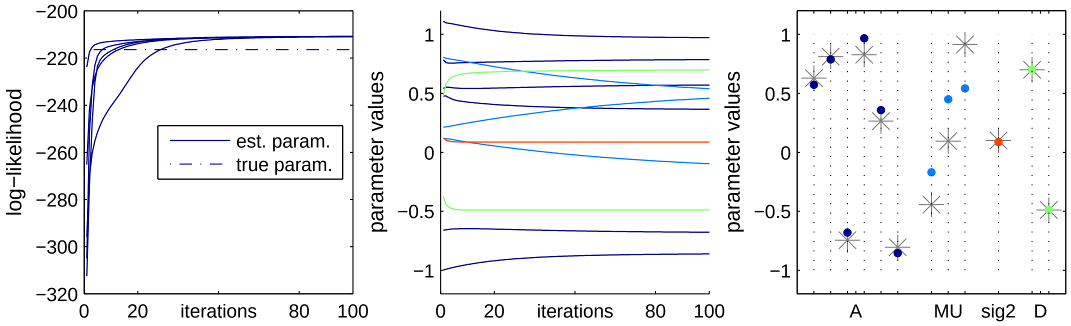

Figure 6.4: Log-likelihood and parameter traces. We generate data from model (CC) with dimensions $p = 3$ (observed variables) and $q = 2$ (latent variables). We consider five different parameter initalizations for `emGrade`. The left plot shows the trace of the log-likelihood evaluations among 100 EM-iterations. The middle plot shows the traces of the parameters A, μ, σ^2 and D for one parameter initialization and the right plot illustrates the final parameter estimates of *all* initializations after convergence together with the true parameter values (gray stars).

6.3.3 Repeated and missing observations

In this part we extend the basic model of `emGrade` and allow missing components in the observed variables and repeated measurements. Since in Bayesian networks a random variable is either latent or observed we need to split the multivariate observed variables into multiple one-dimensional variables. We further introduce replicates of the variables to account for repeated measurements. A graphical representation of the extended model is shown in Figure 6.3b.

As before, let $i = 1, \ldots, N$ index the nodes in the underlying network. For each node we assume a latent random variable $\boldsymbol{s}(i) = (\boldsymbol{s}_1(i), \ldots, \boldsymbol{s}_q(i))'$. In contrast to the previous sections, we further introduce multiple variables $\boldsymbol{x}^r(i) = (x_1^r(i), \ldots, x_p^r(i))'$ where $r = 1, \ldots, r_i$ indicates the measurement replicate. The number of replicates can be dependent on the index i. The mixing model for $\boldsymbol{x}_k^r(i)$ is then given by

$$\boldsymbol{x}_m^r(i) = \sum_{k=1}^{q} a_{km}\boldsymbol{s}_k(i) + \mu_m + \boldsymbol{\varepsilon}_m^r(i) \ ,$$

where $(a_{1m}, \ldots, a_{qm})$ is the mth row of the mixing matrix and μ_m is the mth component of the mean parameter. We further assume univariate noise variables $\boldsymbol{\varepsilon}_m^r(i) \sim \mathcal{N}(0, \sigma^2)$.

The variables $\boldsymbol{x}_m^r(i)$ can be either observed or *latent*; the latter is the case, if for the network component i the measurement under the mth experimental condition in replicate r is not available or not reliable. Let now $\boldsymbol{X}$ collect all variables $\boldsymbol{x}_m^r(i)$ that are observed, and let $\boldsymbol{X}^0$ collect all variables where the measurement is missing. In the E-step we infer $\boldsymbol{S}$ and $\boldsymbol{X}^0$ jointly from the posterior distribution $\boldsymbol{S}, \boldsymbol{X}^0 \mid \boldsymbol{X}, \theta$. For the extended model the density function in (6.8) is given by

$$\mathrm{p}(\boldsymbol{X}, \boldsymbol{X}^0 \mid \boldsymbol{S}) = \prod_{m=1}^{p} \prod_{i=1}^{N} \prod_{r=1}^{r_i} \mathrm{p}(\boldsymbol{x}_m^r(i) \mid \boldsymbol{s}(i)) \ ,$$

and the number of variables equals $\sum_{i=1}^{N} r_i \, p$. If for a network component i in all replicates the measurement under the experimental condition m is missing, we only introduce one latent variable $\boldsymbol{x}_m^1(i)$. Thus, the number of variables can be different for every m; in this case, we perform maximization of mixing matrix and mean parameter for each row separately.

In the following we demonstrate the impact of repeated and/or missing observations on the predictive power of the latent variable model. We assume that the variable $\boldsymbol{x}(i)$ is either completely unobserved or that we have multiple observed variables $\boldsymbol{x}^1(i), \ldots, \boldsymbol{x}^r(i)$ with a fixed number of replicates r. We assign the true parameters $\theta = (A, \mu, \sigma^2, D)$ to the model and perform the E-step, i. e. we infer source signals from the posterior expectation. We then compare the estimated and true source signals dependent on the number of replicates and/or missing observations and dependent on the variance of the observation noise. For evaluation we use the distance measure minDist† introduced in Section 4.5; since we fix the mixing matrix we actually do not need to correct for sign and permutation of the estimate. In Figure 6.5 the data is generated from model (CC) with weights +1 and random parameters $\theta = (A, \mu, \sigma^2, D)$. As expected, we find a better source recovery if we have many repeated and no missing variables as well as a low noise level. The performance also increases if the dimension of the latent variables is smaller than the dimension of the observed variables (A.-B.), and disregarding the graph structure of the observations yields a worse performance (C.). For the other graph models (TF) and (LL) the results are similar.

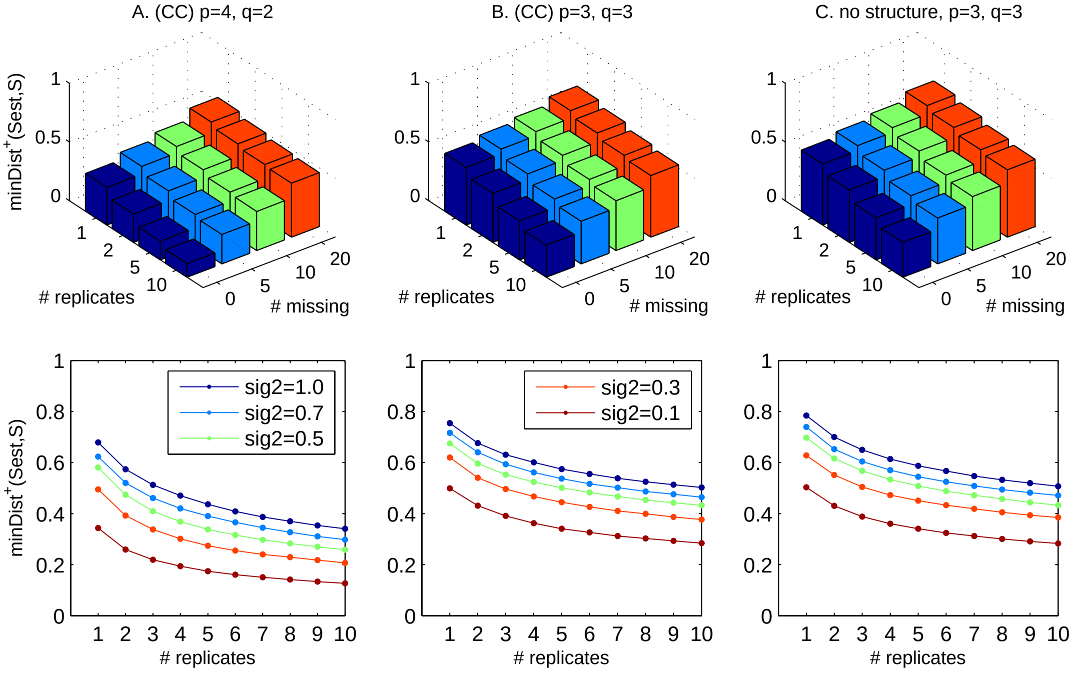

Figure 6.5: Source recovery from repeated observations with missing values and observation noise. We generate repeated data from model (CC). In the upper plots we ignore up to 20 of the observed variables, and in the bottom plots we add noise from $\mathcal{N}(0, \sigma^2)$ for $\sigma^2 = 0.1, 0.3, 0.5, \ldots, 1.0$ to all entries. We infer the source signals from the posterior distribution, where the data and the true parameters are given. The plots show the mean difference over 100 runs between the original source signals and the estimates. In A. we fix the dimensions at $p = 4$ (observed variables) and $q = 2$ (latent variables), in B. we have $p = q = 3$. In C., for comparison, we disregard the structure of the data and consider the trivial network without edges for source estimation.

6.3.4 Separation of sub-networks

Until now we assumed one q-dimensional source model $\mathcal{M}(G, q)$ to jointly model all source signals $\boldsymbol{s}_k = \big(\boldsymbol{s}_k(i)\big)_{i=1}^N$ for $k = 1, \ldots, q$. If we have more detailed information about single components of a larger network (e.g. pathways in a gene-regulatory network) and we want to separate the data according to these sub-networks we can model the distribution of each source signal separately. Let therefore $P_1, \dots, P_q$ be weighted graphs on the same set of nodes. For each source we consider the 1-dimensional source model $\mathcal{M}(P_i, 1)$ with graph-delayed covariance $d_i \in \mathbb{R}$. We then assume that the joint distribution of S decomposes as

$$\mathrm{p}(\boldsymbol{S}) = \prod_{k=1}^{q} \mathrm{p}(\boldsymbol{s}_k) = \prod_{k=1}^{q} \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{s}_k(i) \mid \mathbf{Pa}_k(i)) \ . \tag{6.12}$$

We denote this source model based on q different pathways as $\mathcal{M}(P_1, \dots, P_q)$. If all pathways are identical (i. e. $P_i = G$ for all i), the above definition yields the original source model $\mathcal{M}(G, q)$ with diagonal graph-delayed covariance $D \in \mathbb{R}^{q \times q}$. The new definition only effects the expectation step; in the graphical representation we split the node for $\boldsymbol{s}(i)$ into q nodes representing the 1-dimensional random variables $\boldsymbol{s}_1(i), \dots, \boldsymbol{s}_q(i)$ for $i = 1, \dots, N$. Again, the junction tree algorithm provides posterior estimates of the latent variables.

6.4 Performance and features

We now evaluate the performance of `emGrade`. We first investigate the empirical convergence of the iterative expectation-maximization scheme in terms of number of iterations and runtime. We compare the method to `Grade` and to BSS methods for time series data. In the last part, we consider a family of information criteria introduced in Section 3.5.4. With this we perform model selection and determine number and structure of the unknown source signals. For all simulations we consider the graph models (CC), (TF), and (LL) from Section 6.2.2. If not stated differently, the egde weights are fixed at +1.

6.4.1 Empirical convergence

We first give an impression of the empirical convergence of `emGrade` in terms of number of iterations and runtime. As stated in Section 6.3.2 we stop the iterative estimation scheme when $|\theta_i - \theta_i^{(0)}| < 10^{-8}$ for all i. The respective parameter values are then the estimation result, and we consider 10 000 EM-iterations at maximum. For illustration we generated data from model (CC) and considered different combinations of $p, q = 1, \ldots, 5$ (dimension of latent/observed variables). Figure 6.6 shows that for $p > q$ the mean number of EM-steps is small, and for $p < q$ or large dimensions $p = q$ the number of EM-steps explodes and many runs do not converge at all. In BSS applications we usually assume $p \geq q$, and we limit ourselves to such cases in the following.

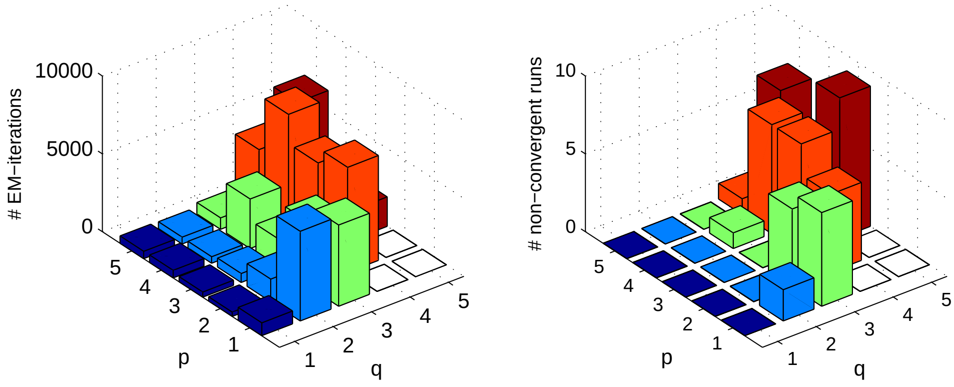

Figure 6.6: Number of EM-iterations. We generate data from model (CC) and consider different combinations of $p, q = 1, \ldots, 5$ (dimension of observed/latent variables). The plots show the mean number of EM-iterations among all convergent runs (left) and the number of non-convergent runs (right). For all p/q-combinations with $q \geq p + 2$ we performed 10 runs of `emGrade`.

6.4.2 Algorithm comparison

We now compare `emGrade` to other BSS algorithms. The most similar algorithm regarding model assumptions is `Grade`, which diagonalizes the sample graph-delayed covariance using singular value decomposition. We distinguish between two versions – `Grade(Pa)` and `Grade(E)`, where the graph-delayed covariance is estimated as $\hat{D}^{\mathrm{Pa}}$ and $\hat{D}^{\mathrm{E}}$ (Section 6.2.3). For comparison, we consider the BSS algorithms `AMUSE` and `SOBI` from Section 4.3.2. Both algorithms assume weakly stationary time series data and we simply resume the order of the random variables. For `SOBI`, we use sample autocovariances at lag 1 and at lags 1-2, respectively. Finally, we compare our results to `PCA`

(Section 4.1.2) and `fastICA` (Section 4.2.1); both act independently of the structure of the random variables.

All algorithms provide estimates of the mixing matrix and the source signals – but unlike `emGrade` they do not estimate the full parameter vector $\theta = (A, \mu, \sigma^2, D)$. Instead of a likelihood-based evaluation of the performance we consider the distance measure minDist from Section 4.5 and compare the mixing estimate to the true mixing. If p-dimensional data is given all algorithms – except `emGrade` – estimate a quadratic $p \times p$ mixing matrix. In case of $p > q$ we only use the first q columns of the mixing estimates. In Figure 6.7 we compare the estimation performance of all algorithms, and we fix the dimensions at $p = 3$ and $q = 2$. For all proposed graph models `emGrade` outperforms the other algorithms in terms of correctness of the estimates, the drawback is a much higher runtime. In case of $p = q$ the improvement of the estimates is less apparent.

6.4.3 Determining the number of source signals

We now take full advantage of the probabilistic modeling and use model selection criteria to determine the correct number of source signals and to identify active pathways in the network. Let $\mathcal{M}(G, q)$ denote the source model introduced in Section 6.2.1 where G is a weighted graph that determines the joint distribution of the latent variables and q is the number of source signals. The information criterion (Section 3.5.4) is given by

$$IC(\mathcal{M}(G, q)) = -2\,\ell(\theta; X, \mathcal{M}(G, q)) + k\,c\ ,$$

where k denotes the number of model parameters and c is some constant. To determine the true number of source signals we fix the graph G and search for the lowest IC value among different source models. In Figure 6.8 we generated data from model (CC) with $q = 3$ the true number of source signals (dimension of the latent variables) and $p = 3, 4, 5$ observations (dimension of the observed variables). We then compare the IC values of $M(\hat{q}) = \mathcal{M}((CC), \hat{q})$ for $\hat{q} = 1, \ldots, 5$, where we consider different constant values c. In case of $p > q$ we find a nearly perfect estimation of the true number of source signals for $c = 2$ (AIC) and $c = \ln(N)$ (BIC).

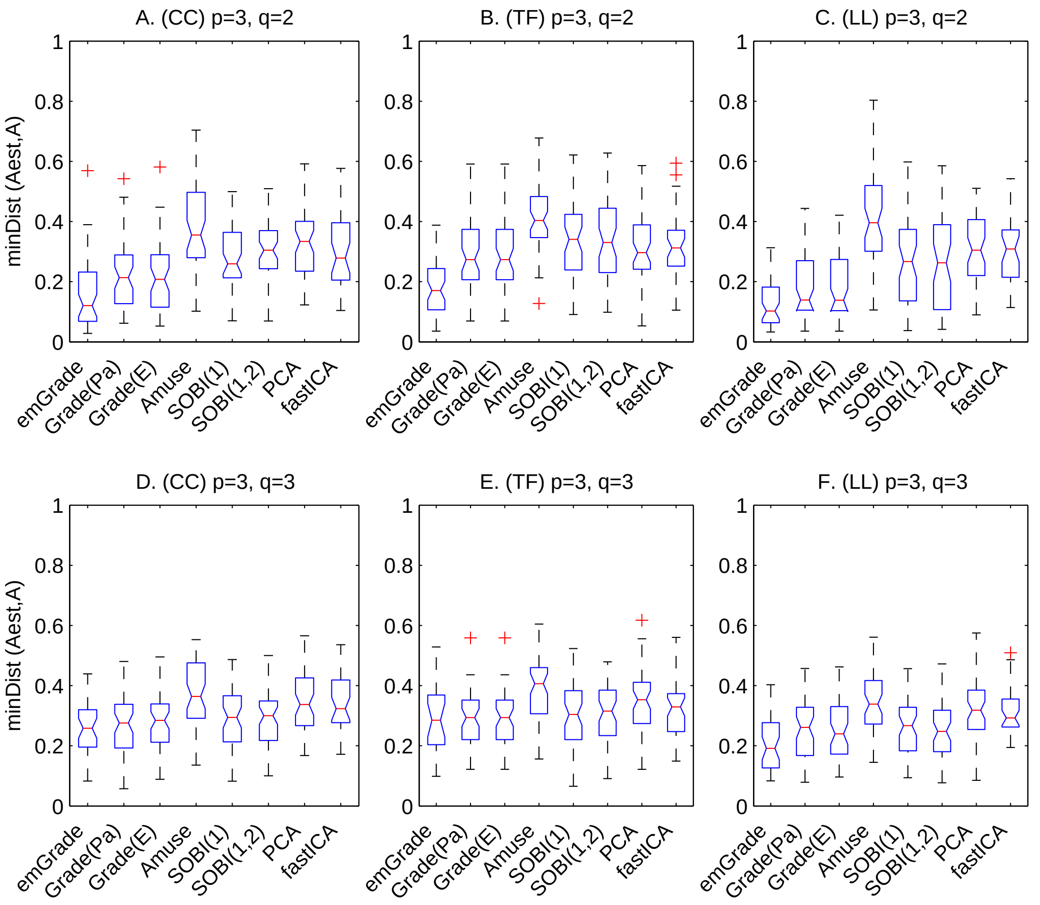

Figure 6.7: Algorithm comparison. The plots show the mean permuted distance over 50 runs of the algorithms `emGrade`, `Grade(G)`, `Grade(E)`, `AMUSE`, `SOBI` (at lag 1, and at lags 1-2), `PCA`, and `fastICA`. In A.-C. we generated data from models (CC), (TF), and (LL) with $p = 3$ and $q = 2$. In this case ($p > q$) `emGrade` yields the best estimation performance for all graph models. In D.-F. where $p = q = 3$ the improvement compared to the other algorithms is smaller.

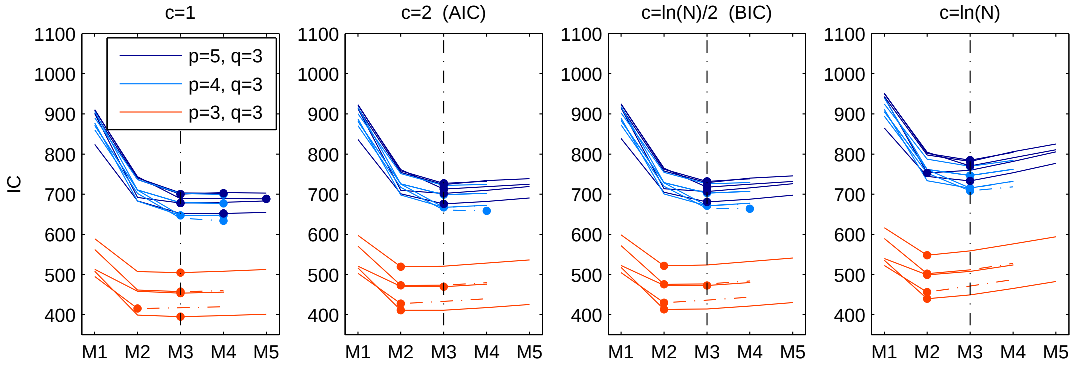

Figure 6.8: Estimation of the number of source signals. We generate data from model (CC) with $q = 3$ source signals (dimension of the latent variables) and p-dimensional observations for $p = 5, 4, 3$ in different colors. For the graph-delayed covariance we consider $D = [0.7, -0.49, 0.3]$. The plots show the IC values of all source models $M(\hat{q}) = \mathcal{M}((\text{CC}), \hat{q})$ for increasing $\hat{q} = 1, \ldots, 5$. In each plot we consider a different constant c. The black vertical lines show the true number of source signals and the dots indicate the selected source model in each comparison, i. e. the model with the lowest IC value. Dashed lines lead to IC values of non-convergent runs (using the log-likelihood value after 10 000 iterations), and some IC values are $+\infty$.

6.4.4 Pathway identification

For pathway identification we divide each of the networks (CC), (TF), and (LL) into three pathways P_1, P_2 and P_3. For (CC) we define the pathways as the three connected components of the complete cell-cycle network, for (TF) we consider the single hub-nodes together with their target nodes as pathways, and for (LL) we consider the two overlapping lines and the additional line as pathways. We then generate data from the pathway source model $\mathcal{M}(P_i, P_j)$ introduced in Section 6.3.4, and we determine the lowest IC value among all source models $M(i, j) = \mathcal{M}(P_i, P_j)$ for $i, j = 1, 2, 3$. If the edges of the pathways are non-overlapping (models (CC) and (TF)) we observe a good pathway identification, for model (LL) often only one pathway is identified correctly (Figure 6.9).

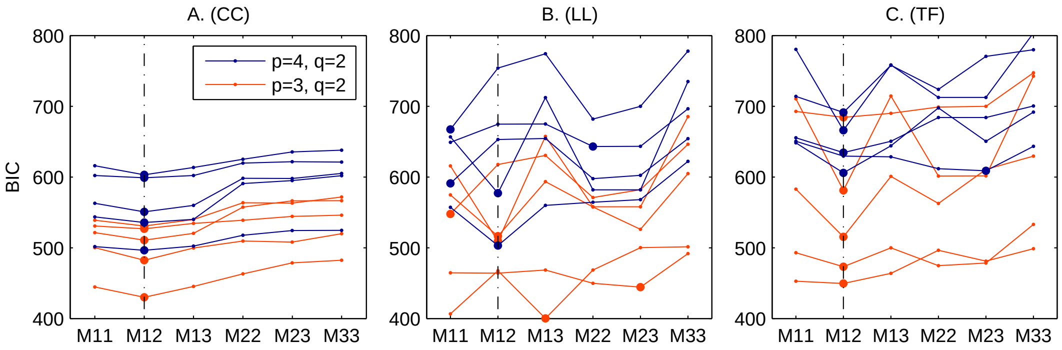

Figure 6.9: Pathway identification. We generate $q = 2$ source signals from the respective pathways P_1 and P_2 of the models (CC), (LL), and (TF). From observations of dimension $p = 4, 3$ (in different colors) we calculate the BIC of all pathway source models $M(i, j) = \mathcal{M}(P_i, P_j)$, where we assume source signals from pathways P_i and P_j $(i, j = 1, 2, 3)$. The black vertical lines show the true pathway combination and the dots indicate the selected source model in each comparison, i. e. the model with the lowest BIC value.

6.5 Conclusions

In this chapter we introduced the probabilistic BSS method `emGrade` for network data. We defined the source model in terms of a stationary Bayesian network and discussed the connection to the separation assumptions of `Grade`. We then introduced observed variables as linear mixtures of the source signals and provided explicit update rules for the parameters in an expectation-maximization scheme. Furthermore, we introduced multiple and missing observations and assumed different network structures for each source signal. We further investigated the performance of `emGrade` – on the level of parameter estimation and on the level of model selection. For the latter we evaluated the likelihood function at the parameter estimates.

In simulations we achieved good empirical convergence if the number of source signals was smaller than the number of observations. If the data was generated by the source model defined in Section 6.2.1 `emGrade` outperformed `Grade` and BSS algorithms for time series data. The improvement was more obvious if we had more observations than source signals.

In the model selection part we assumed that the correct number q of source signals is unknown and we determined the parameter estimates for different numbers $q = 1, \ldots, 5$. We then compared the information criterion values (e.g. AIC, BIC) of the estimates. Especially, if the true number of source signals was smaller than the number of observations we found the correct number of source signals. We further generated data with different pathway structures for each source signal. Comparing the information criterion values for all possible combinations of pathways, we could again determine the correct pathways. This was more obvious when the pathways in the network were non-overlapping.

In the next chapter we leave the area of easy-to-interpret simulations and apply `emGrade` to real gene expression data. The network structure is then given by a literature-derived gene regulatory network and we discuss the biological meaning of the estimation results.

7

Application: Gene expression data

In this chapter we apply `emGrade` to publicly available gene expression data from systemic inflammation in humans. We describe the experiments and the pre-processing of the data, and we compare estimation based on different network structures which are derived from online databases. The probabilistic framework of `emGrade` enables us to use model selection criteria, and we find that a proper network information indeed improves our model. We further estimate missing observation values and determine the most appropriate microarray probe set for genes that are not uniquely annotated after standard filtering. Finally, we characterize the estimated signals in terms of relevant genes and compare the gene sets from different observations. This leads the way to a biological interpretation of the estimated source signals.

Some of the results are already published in the conference proceeding

- **K. Illner**, C. Fuchs, F.J. Theis (2014). Bayesian blind source separation applied to the lymphocyte pathway. In Proc. 21st International Conference on Computational Statistics (COMPSTAT 2014), 625–632.

7.1 Experiments and data

We consider a study from Calvano *et al.* (2005) where systemic inflammation in humans is under investigation. In their experiments healthy humans were intravenously treated with bacterial endotoxin and gene expression measurements were taken from whole blood leukocytes at time points 0, 2, 4, 6, 9, and 24h after endotoxin administration. The stimulus endotoxin activates innative immune responses (Fong *et al.*, 1990). Using a literature-derived molecular network the aim was to determine changes in the expression measurements and identify important functional modules of this network. The analysis gives insights into the mechanisms of the innative immune response. In our investigation, in contrast, we consider a small sub-network and determine underlying regulatory modules based on BSS techniques. With this we want to deepen the insights about molecular signaling from the high-throughput genomic data.

7.1.1 Gene expression measurements

In the experiments from Calvano *et al.* (2005), gene expression analysis was performed using the Affimetrix chips HG-U133A and HG-U133B for the human genome. The chips contained a total of 22 338 and 22 649 probe sets, respectively. More background about the microarray technology is given in Section 2.3. With both chips expression measurements were taken from four patients treated with intravenous endotoxin and from four control patients without treatment. For each patient we have observations at time points 0, 2, 4, 6, 9, 24h, only for one control patient the measurements at time points 4h and 6h are missing. For our analysis we concentrate on the measurements from HG-U133A.

As pre-processing we perform quantile normalization introduced by Bolstad *et al.* (2003) and implemented in the R-package 'affy' (Gautier *et al.*, 2004). After normalization the distributions of probe intensities of each array are similar. In particular, they approximately share the same mean and the same quantiles. We further perform filtering using the R-package 'limma' (Smyth, 2005) and get normalized expression values of 12 683 human genes. For source separation we only consider a subset of $N = 91$ genes that are associated with a specific pathway. The derivation of the pathway is discussed in the next paragraph. We further assign the data to the eight individuals. We thus have observations LPS1-4 from the four treated patients and observations PT1-4 from the non-treated patients. Each selected gene corresponds to an observed random variable, and since we have measurements from six time points the dimension of the observed variables is $p = 6$. For patient PL2 the dimension is $p = 4$.

In simulations we found that the performance of `emGrade` increases if the variance of the observed variables has a similar range compared to the variance of the unobserved variables. Since the components of the latent variables have unit variance (Section 6.2.1) we scale the sample variance of the observed components $(x_k(1), \dots, x_k(91))$ for $k = 1, \dots, p$ to 1, accordingly.

7.1.2 Literature-derived pathways

In our BSS method we assume an initially known network that describes the dependencies between the variables (genes). To derive a network that reflects differences between control and treatment group we consider pathway information from the Genomatix Pathway System (GePS). Based on the differentially expressed genes the software provides (amongst others) biological processes from Gene Ontology where these genes are enriched. One highly significant pathway is "lymphocyte activation". In this pathway 91 from a total of 486 genes are differentially expressed which results in a p-value of 7.10e-22. As network structure net1 we consider these 91 genes together with 138 edges representing validated binding sites of transcription factors. For comparison we also investigate the less significant pathway "cell proliferation" where 158 from a total of 1610 genes (p-value 2.07e-9) are differentially expressed. If we restrict this pathway to the 91 genes from before we get a sub-network net2 with 64 edges. Both networks are shown in Figure 7.1. As required for `emGrade` the networks are directed and acyclic. Since no further information about the strength of interaction is available, we fix all egde weights at $\lambda_{ij} = 1/\#\{\text{parents of } v_j\}$. This assumption is in agreement with Kowarsch *et al.* (2010).

7.2 Model validation

We apply `emGrade` to patients LPS1-4 and PL1-4 separately and for comparison to LPS1 and PL1 jointly. In the first the dimension of observed variables equals $p = 6$ (except for PL2 where $p = 4$) and for the joint modelling of LPS1 and PL2 the dimension equals $p = 12$. The network structure is given by net1. To validate our model we consider the

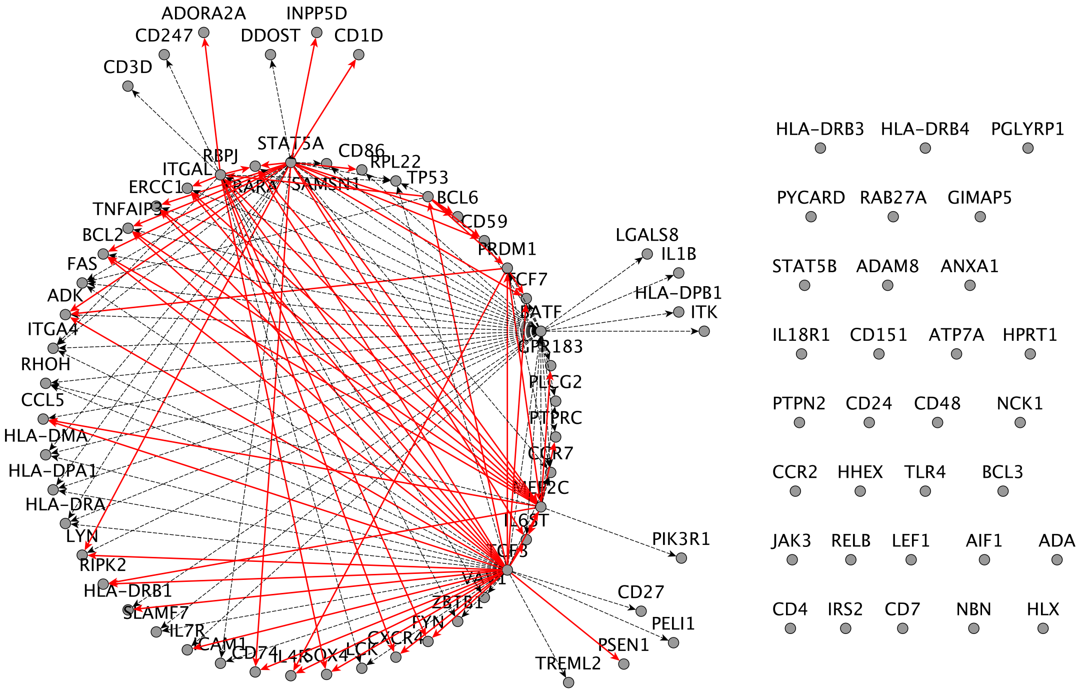

Figure 7.1: Transcriptional signaling pathways. The network net1 consists of the 91 differentially expressed genes from the pathway "lymphocyte activation" together with 138 edges (black *and* red) representing validated binding sites. The sub-network net2 consists of the signaling pathway "cell proliferation" restricted to the same set of 91 genes; it has 64 edges which are indicated in red.

coefficient of determination which is given by

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_r \sum_{i=1}^{91} \|x^r(i) - f^r(i)\|_2^2}{\sum_r \sum_{i=1}^{91} \|x^r(i) - \bar{x}^r\|_2^2} \,.$$

The total sum of squares SS_{tot} compares the data to its mean, the residual sum of squares SS_{res} compares the observation $x(i)$ to the predicted values $f(i) = \hat{A}\hat{s}(i) + \hat{\mu}$ for $i = 1, \ldots, 91$. The superscript r indicates all (separately) considered data sets, e. g. all patients LPS1-4. Figure 7.2 shows the R^2-values for all treated patients LPS1-4, for all control patients PL1-4 and for the joint modelling of LPS1 and PL1. In addition the estimated noise variance is shown. In all settings we increase the number of source signals from $q = 1, \ldots, 4$. With an increase of this number we find a better fit of the model (R^2 close to 1) and a lower estimated noise variance. Notably, there is an overall better fit of the model for the control patients PL1-4.

We further investigate different network structures. We consider net1 and net2 and for comparison the trivial network without any edges (net0). If we fix one data set, we can compare the BIC values for different networks and different numbers of source signals. Figure 7.3 indicates that for all data sets the informative net1 is more appropriate compared to net0 (lower BIC values). In contrast, the sub-network net2 does not lead to an overall better or worse performance compared to net1. Importantly, we find that for the treatment groups LPS1-4 a higher number of source signals is preferred. This indicates that in case of treatment more sub-processes of the pathway “lymphocyte activation” are active.

7.3 Results using emGrade

In the following, we present our main findings from the application of `emGrade` to the gene expression data. In particular, we estimate missing observation values and biologically evaluate the estimated source signals. In all investigations, we assume the pathway “lymphocyte activation” (net1) as underlying network structure.

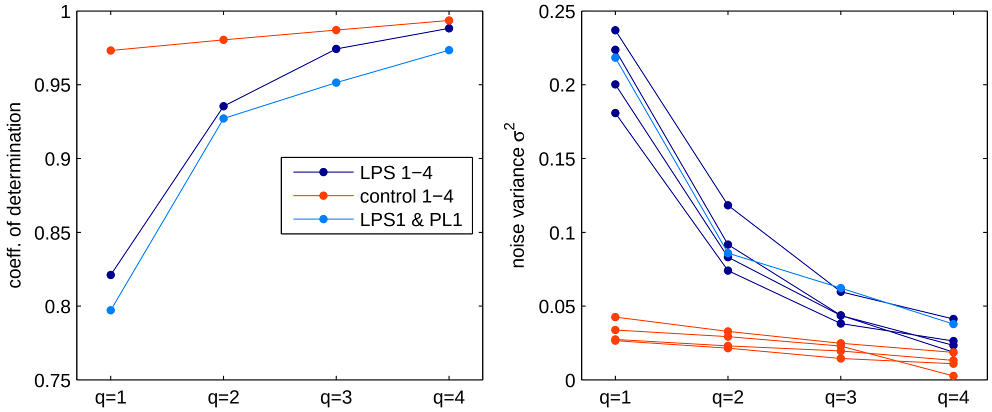

Figure 7.2: Model validation and noise variance. We apply `emGrade` to all patients LPS1-4 (treatment) and PL1-4 (control) separately and to LPS1&PL1 jointly. As network structures we consider net1. The left plot shows the coefficient of determination R^2 in case of $q = 1, 2, 3, 4$ source signals. The right plot shows the estimated noise variance of the `emGrade` model for all single estimating procedures.

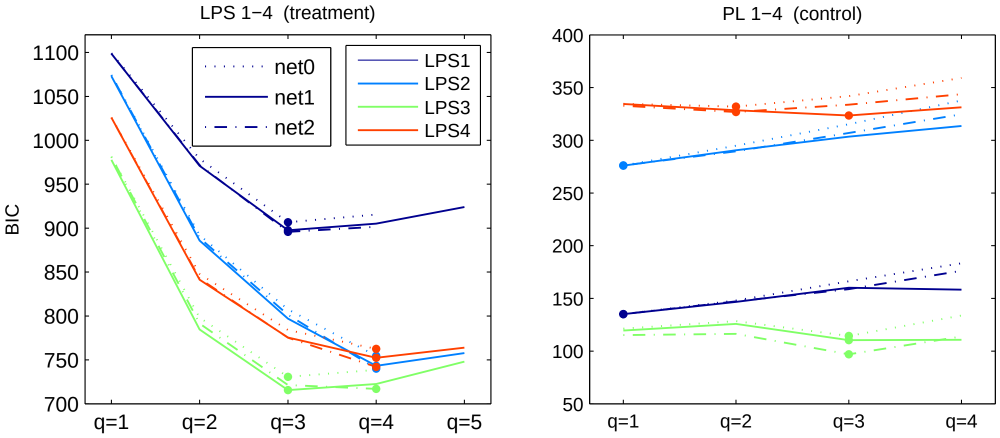

Figure 7.3: Comparison of different networks. The plots show the BIC values for patients LPS1-4 (left) and PL1-4 (right) in case of $q = 1, 2, 3$ and 4 source signals. As network structures we consider net0 and net1 and for LPS1-2 also net2. The different patients are coded in different colors and the dots indicate the value for q with the lowest BIC value.

7.3.1 Missing observation values

We first investigate the predictive power of our model for missing observation values. As stated in Section 6.3.3 we can easily treat missing observations as additional latent variables in our Bayesian network. We therefore leave out the observation value of one gene in the data set LPS1 and compare the true observation $x(i)$ to the model prediction $f(i) = \hat{A}\hat{s}(i) + \hat{\mu}$ for $i \leq 91$. Here, we distinguish between genes that are highly connected in net1(high degree) and genes that are not connected (zero degree). Figure 7.4 shows the Euclidean distances $\|x(i) - f(i)\|_2$ and the corresponding BIC values for 10 different missing genes. For comparison we estimate parameters and source signals from the complete data set. With increasing number of estimated source signals $q = 1, 2, 3, 4$ we find smaller distances between model prediction and true observation in case of complete data (blue solid line). If $x(i)$ is considered as missing, this trend is less obvious (blue stars). For highly connected genes all neighboring genes give information about the true expression value; we expect these observation values to be reconstructed more easily. In practice we find similar prediction performances on average. Nevertheless, genes with zero degree can more easily guarantee a good model fit (low BIC value) but lead to a worse predicted value at the same time. A gene with such contradictory performance is indicated by a dashed vertical line in Figure 7.4. For genes that are highly connected and, thus, obtain information from the network structure we could not find such contradictory behavior. For comparison, we repeat the investigations using the trivial network net0. Expectedly, we find similar distances $\|x(i) - f(i)\|_2$ for both networks when the genes have zero degree in net1. For genes that are highly connected in net1 the model predictions derived from both networks differ and with this the distances to the true observations. The results for the trivial network are shown in Figure 7.4 in grey.

The estimation of missing observations provides a useful feature in the present data situation: The genes HLA-DRB1 and HLA-DRB3 from net1 are annotated to 5 and 2 probe sets of the microarray chip. Gene filtering performed with the `limma` R-package omits these genes and one does not know which probe set provides the most appropriate expression values. We therefore treat both genes as missing observations and compare our estimates to the measurements of the different probe sets. Table 7.1 shows the microarray measurements of all probe sets together with our estimates. The comparison

	HLA-DRB 1						HLA-DRB 3		
time	obs.1	obs.2	obs.3	obs.4	obs.5	est.	obs.1	obs.2	est.
0h	4.99	5.23	5.26	**5.20**	2.46	5.44	5.20	**2.46**	2.52
2h	4.24	4.26	4.38	**4.11**	2.76	3.74	4.11	**2.76**	2.29
4h	4.26	4.65	4.47	**4.43**	2.54	4.66	4.43	**2.54**	2.81
6h	4.52	4.81	4.58	**4.66**	2.76	4.60	4.66	**2.76**	2.38
9h	4.66	5.21	5.22	**5.04**	2.62	5.15	5.04	**2.62**	2.61
24h	4.86	5.28	5.12	**5.23**	2.08	5.11	5.23	**2.08**	2.21

Table 7.1: Identification of the most appropriate microarray probe set. The table shows the microarray measurements from LPS1 at all probe sets that are linked to the genes HLA-DRB1 and HLA-DRB3. If we treat both genes as missing observations we get estimated observations (red). A comparison to the measurements in term of the Eukledian distance identifies the most appropriate annotated probe set for both genes (bold symbols).

suggests to use the 4th probe set as observation for HLA-DRB1 and the 2nd probe set as observation for HLA-DRB3.

7.3.2 Genes associated with source signals

In this part we determine key genes associated with the estimated source signals. Key genes characterize the respective source signals and allow a biological interpretation. Furthermore, we compare key genes of source signals that are estimated from different observations. For a source $s_k = (s_k(1), \dots, s_k(N))$ the set of key genes is given by all genes i with $|s_k(i)| > c$ for some threshold $c > 0$. According to our model assumptions the variance of each source equals 1 and we found that $c = 1$ results in an adequate number of key genes.

First, we investigate the recovery of high absolute values in case of simulated signals. We therefore generate data from net1 with random parameters A, μ, σ^2 and D. The dimension of the observed variables is fixed at $p = 6$ and the dimension of the latent variables at $q = 4$. To compare key genes we align the true and estimated source signals using the sign-changing permutation matrix P that yields the distance $\text{minDist}^\dagger(\hat{S}, S)$ from Section 4.5. The four true and estimated source signals together with the counts for

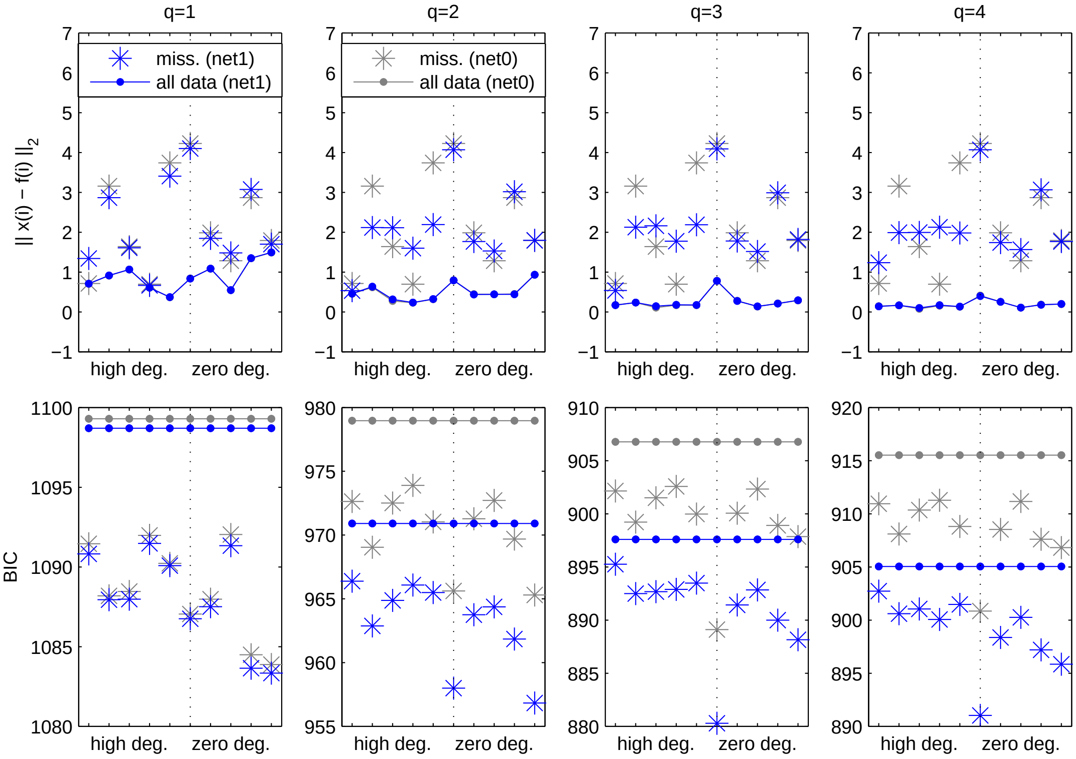

Figure 7.4: Reconstruction of missing observation values. We consider data LPS1 where we leave out the measurements of single genes with high and low connectivity in net1. The degrees of the highly connected genes are 40, 35, 26, 19 and 17 – the remaining genes have zero degree. Using `emGrade` we estimate $q = 1, 2, 3, 4$ source signals (left to right) and determine the model prediction $f(i) = \hat{A}\hat{s}(i) + \hat{\mu}$ where we assume that the observation $x(i)$ $(i \leq 91)$ is missing. *Upper plots:* The blue stars indicate the Euclidean distances $\|x(i) - f(i)\|_2$ between model prediction and true observation. The blue solid lines show the corresponding distances when $f(i)$ is estimated from the complete data set. For comparison, the gray stars and solid gray lines indicate corresponding results when the trivial network net0 is assumed. *Lower plots:* Shown are the corresponding BIC values for all above settings.

consistent key genes are shown in Figure 7.5. Moreover, for all sources the contingency tables of “true key gene/no true key gene” versus “estimated key gene / no estimated key gene” are as follows:

1.)

key gene	true	not true
estimated	17.6	3.3
not estimated	12.1	67.0

2.)

key gene	true	not true
estimated	27.5	12.1
not estimated	12.1	48.4

3.)

key gene	true	not true
estimated	16.5	12.1
not estimated	15.4	56.0

4.)

key gene	true	not true
estimated	15.4	8.8
not estimated	7.7	68.1

All counts are given in percent with respect to the total number of $N = 91$ genes. The higher percentages on the main diagonal indicate correct determination of key genes and no key genes, respectively. With these findings on simulated data, we now compare the source estimates from different treatment groups LPS1-3. We align the signals using the generalized distance function

$$\mathrm{minDist}_3^{\dagger}(S_1, S_2, S_3) = \frac{1}{\sqrt{qN}} \min_{\substack{P_1 = I_p \\ P_2, P_3 \in \mathcal{P}}} \sum_{i,j} \| P_i S_i - P_j S_j \|_{\mathrm{F}} \; . \tag{7.1}$$

Figure 7.6 illustrates the alignment of source signals and Figure 7.7 indicates that we have a higher key gene agreement for the treatment groups LPS1-3 compared to the control groups PL1-3. For the control groups only the network without edges (net0) yields a source with high agreement of key genes.

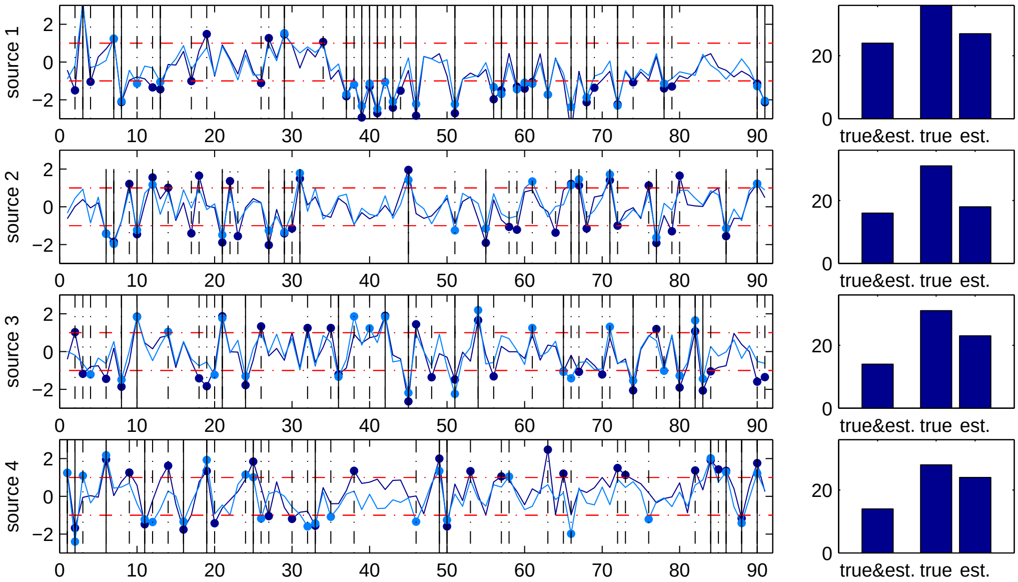

Figure 7.5: Alignment of true and estimated source signals. We generate data from network structure net1 with random parameters A, μ, σ^2 and D and determine the `emGrade` source estimates. The plots on the left show the alignment of true and estimated source signals together with the respective key genes (dots). The horizontal red lines show the thresholds for key gene selection. Solid vertical lines indicate genes that are correctly estimated key genes, dashed vertical lines indicate genes that are either true key genes or estimated key genes. The bars on the right provide the counts for these three groups.

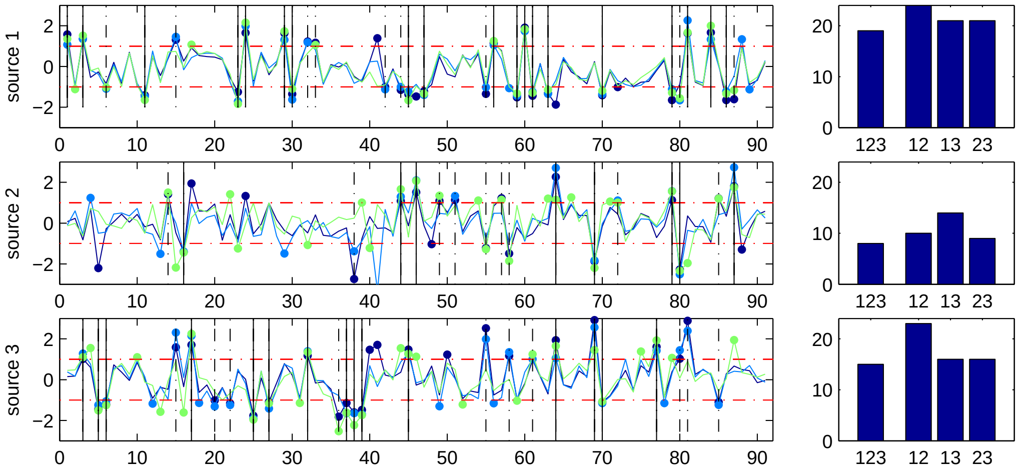

Figure 7.6: Alignment of source signals and intersection of key genes for LPS. For patients LPS1, LPS2 and LPS3 and network structure net1 we determine the `emGrade` source estimates for $q = 3$. The plots on the left show the aligned source signals (different patients in different colors) together with the respective key genes (dots). The horizontal red lines show the threshold for key gene selection. Solid vertical lines indicate genes that are key genes in the aligned sources from all patients, dashed vertical lines indicate genes that are key genes in the aligned sources from at least two patients. The bars on the right provide the counts of key genes in all estimates (123) and the counts of key genes in two estimates (12), (13) and (23).

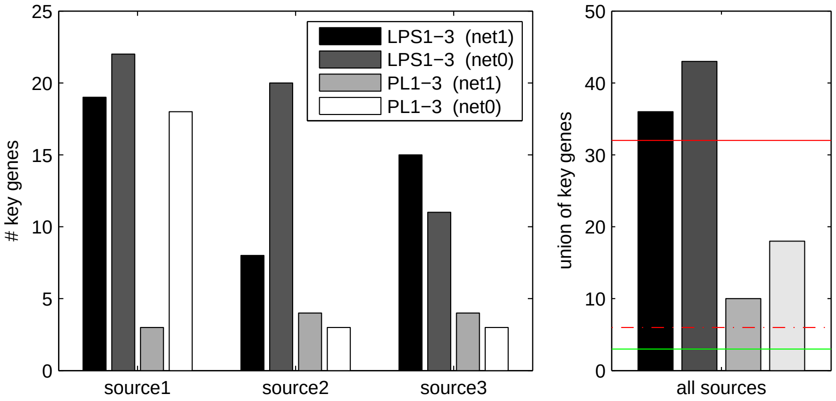

Figure 7.7: Intersection of key genes for LPS and PL. For patients LPS1-3 and controls PL1-3 and network structures net1 and net0 we determine the `emGrade` source estimates. The left figure shows the counts of genes that are key genes in all aligned source estimates from LPS1-3 or PL1-3, respectively. The right figure gives the total number (union) of key genes in all sources. The solid red line indicates the count of identical key genes in LPS1-3 (net1) and LPS1-3 (net0), the dashed red line indicates the corresponding count for PL1-3. The green line is the count of identical key genes in all four groups.

7.3.3 Algorithm comparison

In this part we compare `emGrade` to other BSS methods in terms of biological interpretability of the results. In particular we consider the algorithms `Grade`, `SOBI` (using lags 1-2 and lags 1-10) and `fastICA`. All these algorithms have been applied to simulated data from networks in Section 6.4.2. Here, we consider the gene expression data from patients LPS1 and PL1. For `emGrade` we set the number of source signals equal to $q = 4$, for the other algorithms the number of source signals equals the number of observations, i. e. $q = 12$. To illustrate some obvious differences between the algorithms Figure 7.8 shows four estimated source signals from `emGrade` and `fastICA`. The latter determines source signals with a low overall variability but with high peaks for individual genes.

The observations LPS1 and PL1 contain information from a treated and non-treated patient. To investigate whether the algorithms identify differences between treatment and control group we determine the correlation between the columns of the mixing estimate and the index vector $[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]$. A high correlation indicates that the respective source signal has a different impact on treatment and control group. Figure 7.9 shows the correlation for all algorithms. Here, only `emGrade` determines one mixing column with very high correlation with the index vector.

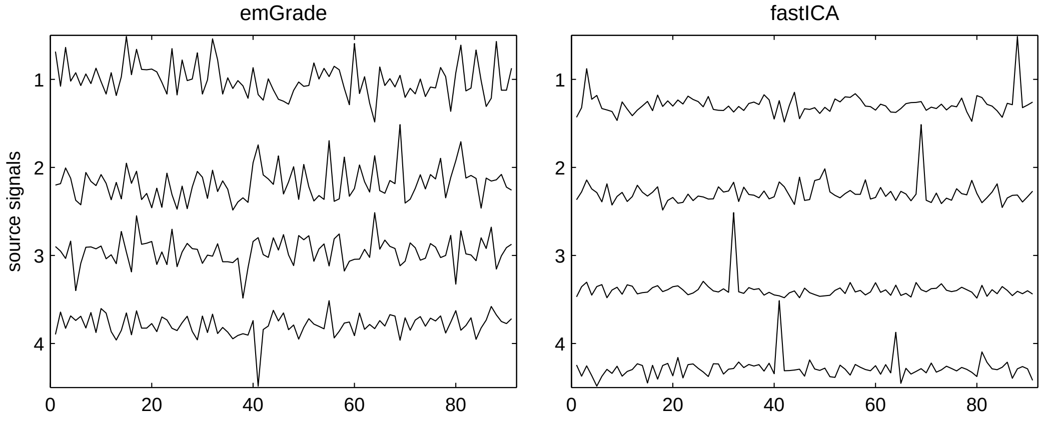

Figure 7.8: Comparison of estimated source signals using `emGrade` and `fastICA`. We consider the gene expression data LPS1 and PL1. The left plot shows the estimated source signals derived from `emGrade` when we assume $q = 4$ source signals, the right plot shows the first four source estimates derived from `fastICA`.

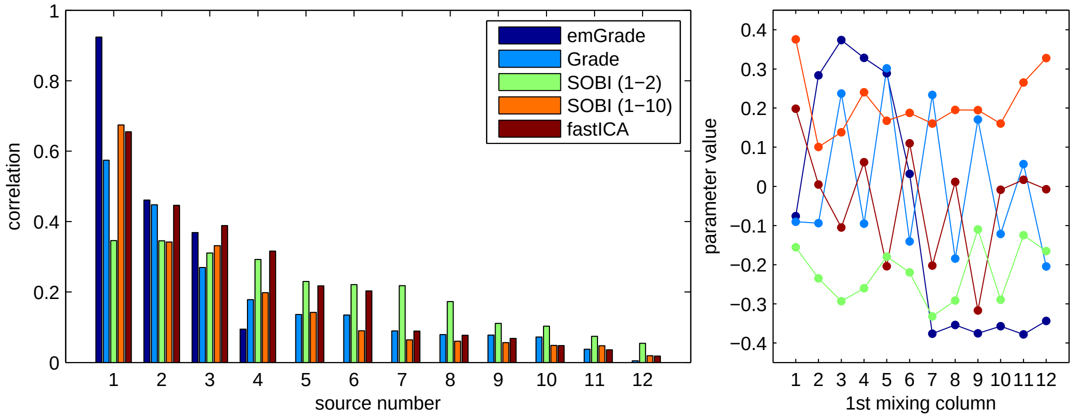

Figure 7.9: Difference of mixing columns between treatment and control. We consider the gene expression data LPS1 and PL1 and determine the mixing estimate using `emGrade`, `Grade`, `SOBI` (with lags 1-2 and 1-10), and `fastICA`. The left plot shows the correlation of all mixing columns with the index vector $[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]$. The number of mixing columns equals the number of estimated source signals. In case of `emGrade` we estimate $q = 4$ source signals, all other algorithms determine $q = 12$ source signals. The right plot shows the mixing columns of all algorithms with highest correlation with the index vector.

7.3.4 Biological interpretation

As stated in the last section, `emGrade` jointly applied to LPS1 and PL1 determines a source signal that highly differs between treatment and control; in the analysis we assumed $q = 4$ source signals. To understand the biological meaning of this specific signal and the three remaining signals we show the key genes of each source signal in the pathway “lymphocyte activation”. In Figures 7.10 and 7.11 the value for each gene in a specific source signal is color-coded; blue indicates positive values and red indicates negative values. We define key genes as genes with absolute value larger than one; these genes are highlighted in the networks in dark blue and dark red. Since source signals are unique only up to sign (and permutation) the meaning of blue and red is exchangeable. We focus on the connected part of the network and find that different source signals affect different parts. The second source signal, for example, contains many key genes and these genes are spread over the network; the third signal, in contrast, concentrates on a few highly-connected genes. Furthermore, the second signal affects many genes in the lower part of the network whereas the last signal is not present at all in this part. With this, our approach motivates the definition of different sub-modules of the original pathway; the functionality of these sub-modules might be experimentally further validated. In addition, we performed enrichment analysis of the key genes from each source signal using Gene Ontology (GO) terms. These terms refer, amongst others, to biological processes and provide lists of involved genes (Section 2.1). Using Fisher's exact test (Fisher, 1922) one can determine biological processes where a considered gene set is overrepresented. Tables 7.2 and 7.3 show the top 15 associated biological processes for the key genes of each source signal. Since the genes are from one specific pathway the enrichment is strongly biased and we find only little difference between the source signals.

7.4 Conclusions

In this chapter we applied `emGrade` to gene expression data. With this we wanted to identify informative source signals that represent active biological processes in the data. We discussed the pre-processing of publicly available microarray data consisting of treatment data LPS1-4 and control data PL1-4. From the Genomatix database we derived the pathway "lymphocyte activation" which reflects differences between the control and treatment group; this yielded the network structure in our BSS method.

In comparison to a network without egdes, the pathway information improved our estimates and we found lower BIC values – this was true for the treatment and the control group. Nevertheless, the pathway information played a major role particularly for the treatment group where more source signals were preferred. We further investigated the estimation of missing observation values. For two genes from the lymphocyte pathway standard annotation to one unique microarray probe set failed. When treated as missing observations `emGrade` could identify the most appropriate annotated probe set in both cases. In addition, we investigated the prediction performance of missing observations dependent on the connectivity of the respective gene. Here, we found that the network structure indeed improved the prediction; if genes with zero degree were treated as missing observations this could yield a good model fit but a bad prediction of the actual measurements at the same time. Furthermore, we characterized the estimated source signals in terms of key genes, i. e. genes with high absolute value in the respective source signals. We found a high number of key genes (per source) that were in agreement with LPS1-3. For PL1-3 these numbers were lower. This might again indicate that the pathway “lymphocyte activation” better explains the dynamics in the treatment group compared to the control group.

In the last part, we considered data LPS1 and PL1. In contrast to other BSS algorithms, only `emGrade` could identify a signal with significantly different weightings in treatment and control. This means, the signal represents regulatory processes that differ between treatment and control. To understand the biological meaning of the estimated source signals we mapped all signals derived from `emGrade` to the lymphocyte pathway. We found that each signal is present in a different part of the network. This indicates that the signals explain different regulatory submodules of the pathway. Finally, these submodules give an idea about the composition of the pathway “lymphocyte activation”.

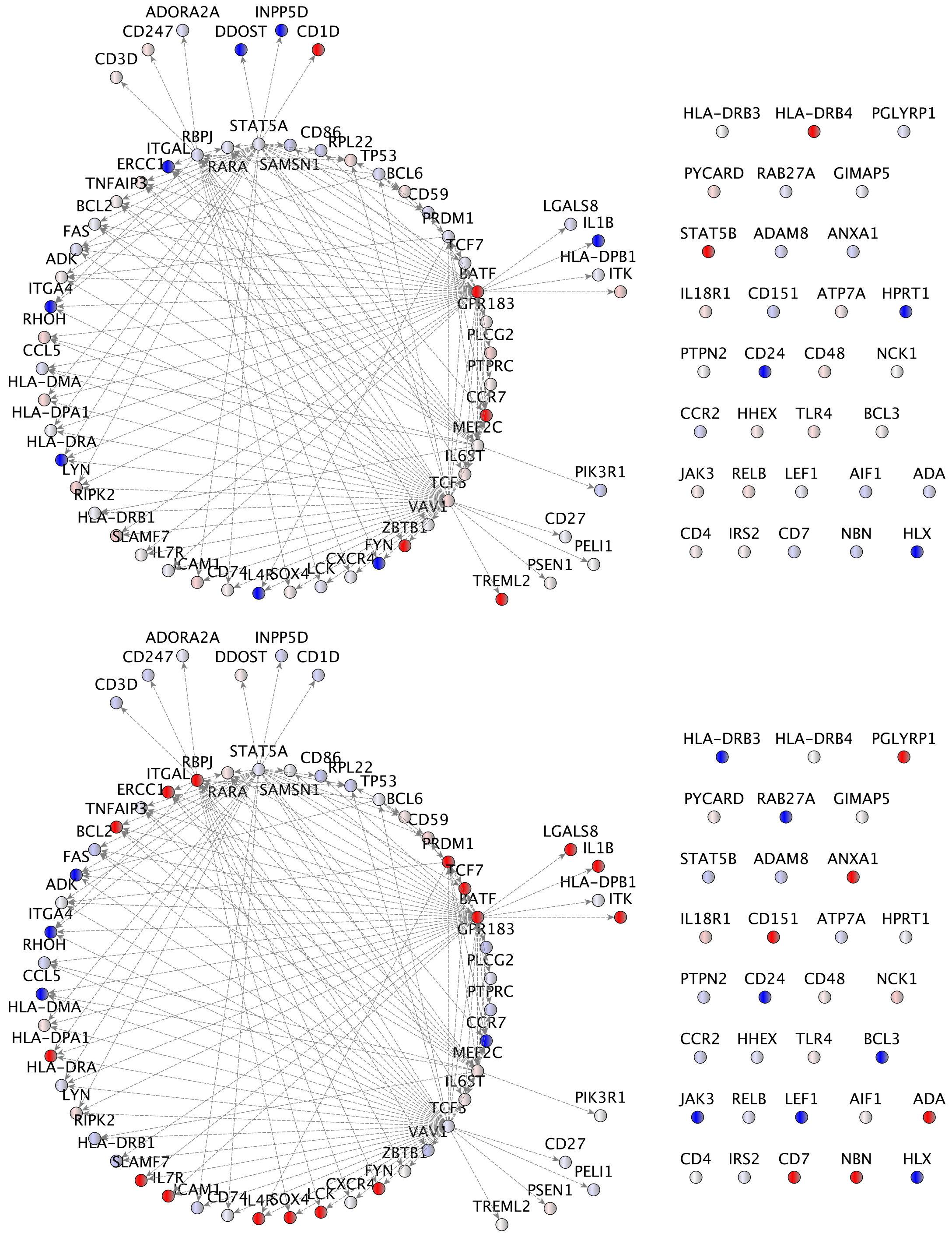

Figure 7.10: Key genes derived from `emGrade`, 1st and 2nd source signal. We applied `emGrade` jointly to data LPS1 and PL1; the estimated source signals are color-coded in the network with positive values in blue and negative values in red. The upper plot shows the results of the 1st source signal, the lower plot the result of the 2nd source signal.

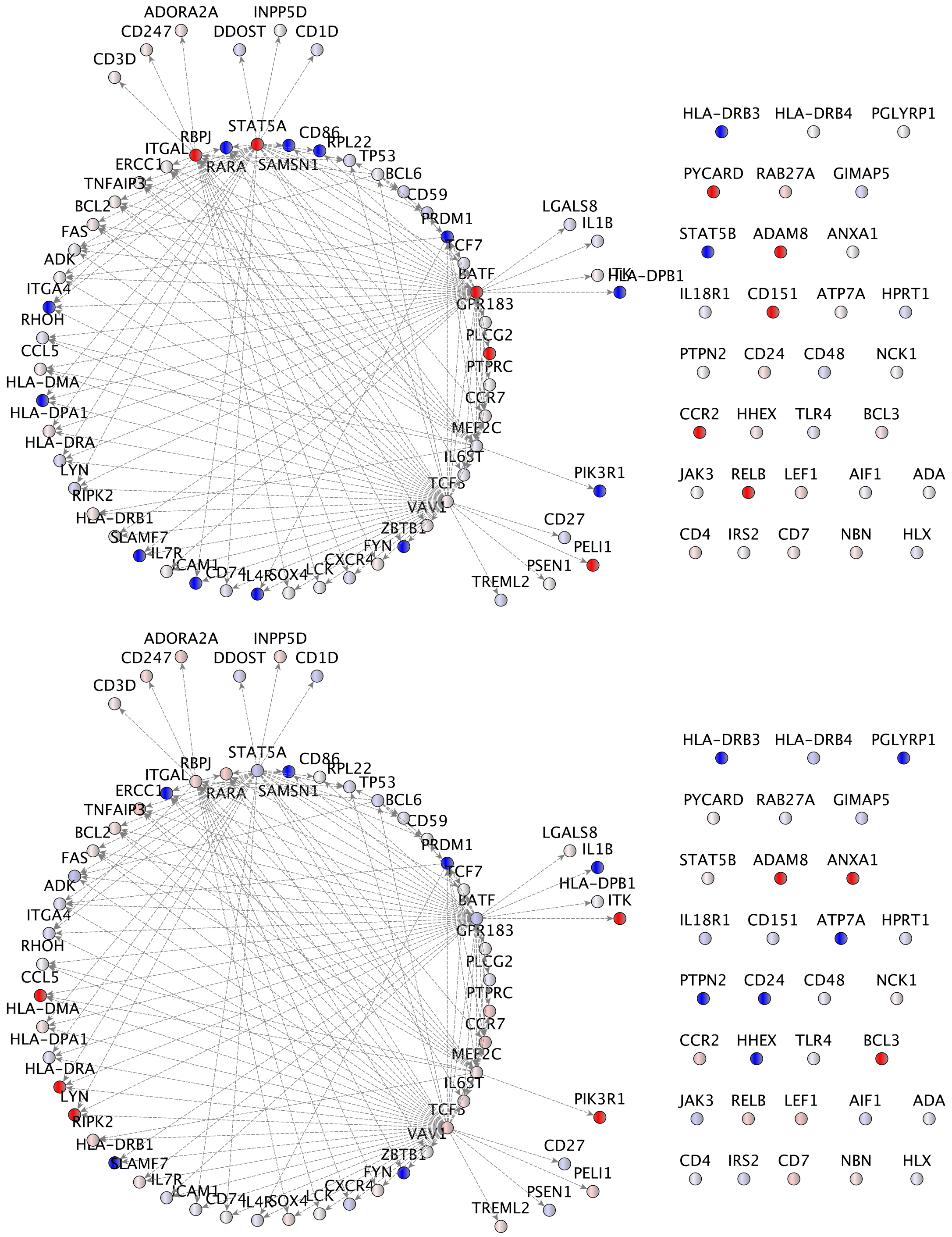

Figure 7.11: Key genes derived from `emGrade`, 3rd and 4th source signal. We applied `emGrade` jointly to data LPS1 and PL1; the estimated source signals are color-coded in the network with positive values in blue and negative values in red. The upper plot shows the results of the 3rd source signal, the lower plot the result of the 4th source signal.

GOBPID	p-value	count	size	term
GO:0046649	1.19e-22	18	451	lymphocyte activation
GO:0045321	2.37e-21	18	531	leukocyte activation
GO:0001775	6.79e-19	18	724	cell activation
GO:0042110	1.13e-17	15	331	T cell activation
GO:0051251	1.41e-16	13	198	positive regulation of lymphocyte activation
GO:0002696	4.21e-16	13	215	positive regulation of leukocyte activation
GO:0050867	7.68e-16	13	225	positive regulation of cell activation
GO:0050870	9.80e-16	12	157	positive regulation of T cell activation
GO:0051249	9.86e-15	13	273	regulation of lymphocyte activation
GO:0050863	3.48e-14	12	210	regulation of T cell activation
GO:0002694	5.46e-14	13	311	regulation of leukocyte activation
GO:0050865	1.62e-13	13	338	regulation of cell activation
GO:0002376	8.62e-12	18	1785	immune system process
GO:0050776	1.48e-11	14	642	regulation of immune response
GO:0006955	1.89e-11	16	1112	immune response

GOBPID	p-value	count	size	term
GO:0046649	2.51e-43	33	451	lymphocyte activation
GO:0045321	6.60e-41	33	531	leukocyte activation
GO:0001775	2.41e-36	33	724	cell activation
GO:0042110	1.32e-30	26	331	T cell activation
GO:0002376	3.21e-23	33	1785	immune system process
GO:0030098	2.05e-22	20	234	lymphocyte differentiation
GO:0006955	1.08e-21	29	1112	immune response
GO:0051249	4.81e-21	20	273	regulation of lymphocyte activation
GO:0002694	6.80e-20	20	311	regulation of leukocyte activation
GO:0050865	3.66e-19	20	338	regulation of cell activation
GO:0002521	4.92e-19	20	343	leukocyte differentiation
GO:0030217	2.07e-16	15	157	T cell differentiation
GO:0002252	6.11e-16	20	490	immune effector process
GO:0002682	8.07e-16	24	930	regulation of immune system process
GO:0051251	7.27e-15	15	198	positive regulation of lymphocyte activation

Table 7.2: Enrichment analysis for key genes derived from `emGrade`, 1st and 2nd source signal. We applied `emGrade` jointly to data LPS1 and PL1 and determined $q = 4$ source signals. Listed are the top 15 biological processes derived from Gene Ontology where the key genes of the 1st and 2nd source signal are enriched (upper/lower table). Key genes are defined as genes with absolute value > 1 and enrichment was performed using Fisher's exact test.

GOBPID	p-value	count	size	term
GO:0046649	6.66e-31	24	451	lymphocyte activation
GO:0045321	3.68e-29	24	531	leukocyte activation
GO:0001775	7.23e-26	24	724	cell activation
GO:0042110	5.55e-20	18	331	T cell activation
GO:0030098	1.16e-18	16	234	lymphocyte differentiation
GO:0051249	1.43e-17	16	273	regulation of lymphocyte activation
GO:0006955	5.17e-17	22	1112	immune response
GO:0002694	1.18e-16	16	311	regulation of leukocyte activation
GO:0002376	2.30e-16	24	1785	immune system process
GO:0050865	4.51e-16	16	338	regulation of cell activation
GO:0002521	5.71e-16	16	343	leukocyte differentiation
GO:0051251	6.32e-16	14	198	positive regulation of lymphocyte activation
GO:0002696	2.05e-15	14	215	positive regulation of leukocyte activation
GO:0050867	3.91e-15	14	225	positive regulation of cell activation
GO:0002682	4.38e-15	20	930	regulation of immune system process

GOBPID	p-value	count	size	term
GO:0046649	2.98e-25	20	451	lymphocyte activation
GO:0045321	8.34e-24	20	531	leukocyte activation
GO:0001775	4.53e-21	20	724	cell activation
GO:0042110	2.92e-18	16	331	T cell activation
GO:0051249	1.62e-17	15	273	regulation of lymphocyte activation
GO:0002694	1.18e-16	15	311	regulation of leukocyte activation
GO:0050865	4.17e-16	15	338	regulation of cell activation
GO:0051251	1.68e-13	12	198	positive regulation of lymphocyte activation
GO:0050863	3.43e-13	12	210	regulation of T cell activation
GO:0002376	3.63e-13	20	1785	immune system process
GO:0002696	4.56e-13	12	215	positive regulation of leukocyte activation
GO:0050867	7.92e-13	12	225	positive regulation of cell activation
GO:0050870	9.25e-13	11	157	positive regulation of T cell activation
GO:0002682	9.80e-13	17	930	regulation of immune system process
GO:0002684	1.36e-12	15	579	positive regulation of immune system process

Table 7.3: Enrichment analysis for key genes derived from `emGrade`, 3rd and 4th source signal. We applied `emGrade` jointly to data LPS1 and PL1 and determined $q = 4$ source signals. Listed are the top 15 biological processes derived from Gene Ontology where the key genes of the 3rd and 4th source signal are enriched (upper/lower table). Key genes are defined as genes with absolute value > 1 and enrichment was performed using Fisher's exact test.

8

Discussion and outlook

In this thesis, we presented probabilistic BSS methods that explicitly make use of the structural information of the data. We demonstrated the power of probabilistic modeling in various simulation scenarios and in an application to gene expression data. In this chapter, we shortly review the proposed methods and their applicability and summarize our results. We then come forward with proposals about other areas of application and possible future research.

8.1 Summary

In the first part, we considered BSS models for weakly stationary time series; here, a mixing estimate can be derived using joint diagonalization of autocovariances. The recently and partly in joint work published algorithms `SOBIdef` and `SOBIsym` are the first to provide mixing estimates together with limiting distributions. Both algorithms represent different conceptual approaches of determining an (un-)mixing estimate. To evaluate the performance of these relatively new algorithms we provided an extense simulation study; for comparison, we used the joint diagonalization algorithms `SOBI` and `ACDC` and we invented further pairwise variants. In conclusion, we found that particularly `SOBIsym` provides a reasonable alternative to other joint diagonalization algorithms; it determines the same mixing estimates as `SOBI` with only slightly longer runtimes.

Based on the limiting distributions of the estimates we elaborated methods to identify the mixing pattern and to decide, for example, whether entries in the mixing estimate are actually zero. First, we introduced a family of linear hypothesis tests; in 80% of our simulations we could correctly reject the hypothesis that an estimated mixing entry equals zero if the true value equals at least 0.02. In addition, we provided a setup to perform model selection. We considered model selection criteria with different penalties for complexity (including AIC and BIC) and we introduced variants with lower computational costs. In simulations with different time series models we found very high recovery rates of the true zero entries using BIC and hypothesis testing on single entries. The other proposed methods with differing penalizing constants tended to underestimate the number of true zero-entries. In Appendix A we further show that penalized optimization of the joint diagonalization problem fails. This justifies the use of advanced methods like the proposed ones.

In the second part we moved on to more general network structures. We proposed a new BSS method where we assumed a (known) weighted directed acyclic graph as underlying data structure. The variable model was realized in form of a Bayesian network and the unknown source signals were represented by latent variables. Continuing the approach of `Grade` (Kowarsch *et al.*, 2010), we substantiated the definition of weak stationarity for regulatory data. Together with the Markov property of Bayesian networks we could define a descriptive *source model* for regulatory data; it depends on a single parameter for each source signal. To infer the model parameters, we provided an expectation-maximization scheme with explicit update formulas; this also yields the name `emGrade` for our method.

In contrast to `Grade`, we assume a directed *acyclic* graph and the distribution assumptions are more specific and in a sense restricting. However, the true power of `emGrade` originates from the flexible modeling in terms of a Bayesian network. The model is, for example, capable to include repeated observations and missing observed components. Moreover, the source signals can be modeled individually assuming different network structures on the same set of nodes. This is particularly interesting, when the network naturally splits into substructures and one is interested in the impact of these substructures in explaining the data.

We evaluated the performance of `emGrade` and considered three different network structures inspired by biological network motifs. Expectedly, we found that the estimation performance increases with the number of repeated observations and decreases with the amount of missing observed components. In comparison to other BSS methods, `emGrade` showed the most accurate estimation results; the difference to the other methods was clearer in overdetermined settings with more observations than source signals. In particular, `emGrade` outperformed `Grade` when the data was generated using our source model; the loss is a much higher runtime due to the iterative expectation-maximization scheme. In simulations with `emGrade` we could further identify the true number of source signals using the BIC. Additionally, we divided the networks into three subnetworks each and modeled the source signals individually. Again, model selection with the BIC identified the true combination of source signals among all possible combinations in most of the simulations.

In the last part, we applied `emGrade` to gene expression data. In the considered experiments systemic inflammation in humans was under investigation and the data consisted of patients treated with intravenous endotoxin and a control group. As network information we used the pathway "lymphocyte activation" which is known to reflect differences between both groups. In comparison to the trivial network, we found that the pathway information indeed improved the model performance. Furthermore, the prediction of missing observations was more reliable for genes that were highly connected; this indicates that the genes obtain information from their neighbouring genes. The most important results of our investigations using `emGrade` are as follows. When treating observations as missing, we could identify the most appropriate probe set on the microarray chip for two genes that were not uniquely annotated after standard annotation. In comparison to other BSS algorithms, `emGrade` could identify a source signal with significantly different regulatory impact in treatment and control group. And finally, the estimated source signals represented different parts in the lymphocyte pathway and thus deepened the understanding of regulatory submodules in the pathway.

8.2 Outlook

For BSS models one can generally consider a variety of modifications and model relaxations. These include, amongst others, a non-linear mixing, a non-instantaneous (or *convolutive*) mixing or for weakly stationary data the relaxation to piecewise stationarity. A further obvious generalization for the method `emGrade` is a fully Bayesian treatment such that, for example, parameter priors and Bayesian model selection become accessible. Besides this, we see some major aspects where the newly proposed model not yet mirrors completely the nature of possible data. In the following, we list the current model limitations and suggest reformulations that allow the application to a broader range of regulatory data.

The first point is about loops and directed cycles in the network. In gene-regulatory networks, for example, self-activating genes and positive or negative feedback loops play an important regulatory role. So far, we used a Bayesian network to describe the source signals. Thus, we implicitly assumed that the network is a directed and acyclic graph. The Markov property together with the stationarity assumption uniquely defined the joint distribution of all variables. To incorporate directed cycles the distribution of the respective variables needs to be redefined. This is mainly a matter of definition but we will loose the Markov property at least at one point in each directed cycle. The original algorithm `Grade`, in contrast, places no restrictions on the network structure; the algorithm is applicable to any directed network.

Besides loops and cycles, the assumption of directed edges can be restricting. In gene-regulatory networks transcription factors activate or inhibit target genes; thus, the edges are directed by definition. If we consider metabolomics data, in contrast, the egdes reflect (reversible) chemical reactions between metabolomic compounds. One can use partial correlations to learn such network structures. Here, the edges are undirected and a random assignment of edge directions leads to different source models (Section 6.2.2). To provide a consistent source model for undirected networks, one could simply define the covariance of two random variables as a combination of a known weight, e. g. from a partial correlation network, the distance of the variables in the network and a source specific scaling parameter. The drawback is on the computational side since we lose the facility of belief propagation in Bayesian networks. Without further assumptions, one needs to invert the joint covariance matrix of all random variables in each expectation step. However, one might exploit structural aspects to overcome this computational issue.

Finally, the proposed model can be seen as starting point for heterogeneous data analysis. This has been an emerging topic in systems biology during the last years (Jeong *et al.*, 2010; Sass *et al.*, 2013). Heterogeneous data analysis means that different types of data, e. g. DNA methylation, gene expression and protein levels, from several experimental layers are modeled jointly to provide new and meaningful insights. A promising approach can be to integrate the interacting layers of the data into our model. Here, the relaxation to piecewise stationarity and possibly different parallel mixing processes can be important. Nevertheless, the flexible Bayesian network structure is capable to model a large variety of dependencies beween different types of variables and, thus, makes our model applicable in a large variety of data situations.

Appendix A

Model selection using penalized optimization

In the maximization problem (5.2) from Chapter 5 we jointly diagonalize autocovariances $M_1, \ldots, M_K$ under the constraint $WM_0W' = I_p$. To perform model selection we consider the following penalized version

$$f_{\text{pen}}(W) = \sum_{j=1}^{p} \sum_{k=1}^{K} (w_j' M_k w_j)^2 - \lambda \sum_{i,j} |a_{ij}| \ ,$$

where $\lambda > 0$ is a constant that forces entries of $A = W^{-1}$ to zero. The case $\lambda = 0$ corresponds to the original unpenalized problem. Like before the maximization problem is constrained by $WM_0W' = I_p$. In case of pre-whitened data ($M_0 = I_p$), we penalize mixing entries after back-transformation. With this, both problems become equivalent but we observe a better numerical performance on whitened data. In all simulations we consider the MATLAB optimizer `fmincon` for constraint non-linear multivariate problems.

In Figures A.1 and A.2 we generated data from the AR(4)-model (*i*) and the ARMA-model (*ii*) from Section 5.3. The true mixing matrix contains $1, 2, \ldots, 6$ zero-entries and the other entries are chosen randomly from $\pm\mathcal{U}[0.1, 1]$. The plots show the MDI values of the mixing estimate of the penalized optimization problem for increasing constant λ. The best performance is achieved for small values of λ. For larger values ($\lambda \geq 10$) the performance becomes very poor. Unexpectedly, the weighting parameter does not really scale with the number of true zero-entries and it is unclear how to appropriately choose it.

Furthermore, we do not observe a subsequent reduction of the mixing entries to zero like in other L_1-penalized maximization problems (Figure A.3). The reason is the constraint $WM_0W' = I_p$. To perform model selection we therefore need to define a threshold and set (small) mixing entries equal to zero. In Figures A.4 and A.5 we consider different threshold values $c = 0.0005$, 0.001, 0.005 and 0.01. The plots show the percentage of correctly determined zero-patterns for time series models (i) and (ii). A high percentage of correct pattern identification is only achieved for $\lambda = 0$ (i.e. in case of the un-penalized problem) or for high values $\lambda \geq 10$. As stated above, high values λ correspond to poor mixing estimates. All in all, the penalizing term does not seem to increase the performance if we are interested in pattern identification.

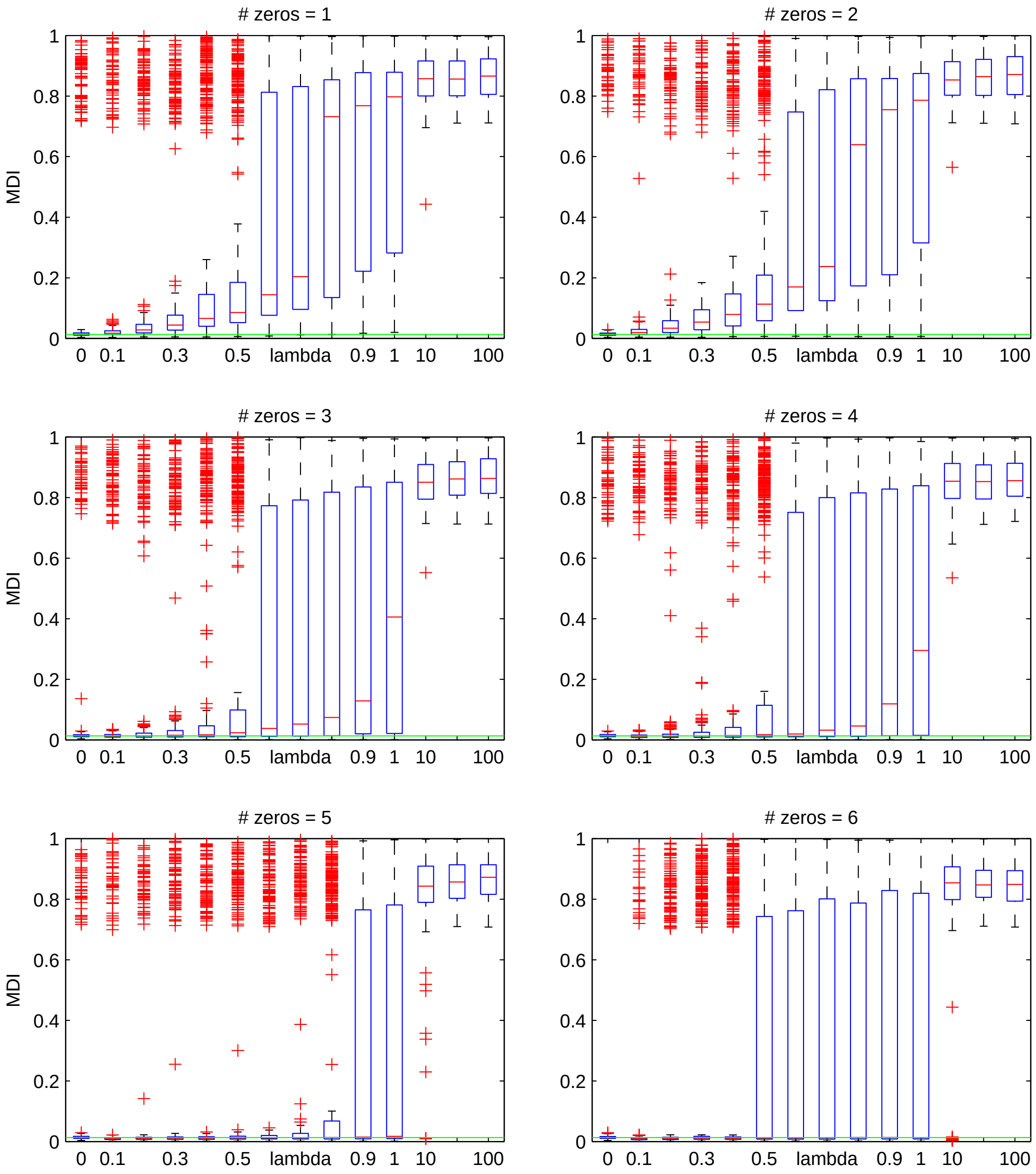

Figure A.1: Joint diagonalization with penalty term (MDI), model (i). The data is generated using the AR(4)-model (i) with a time series length of $T = 10\,000$. The true mixing matrix contains $1, 2, \ldots, 6$ zero-entries. The plots show the numerical optimization performance of the penalized joint diagonlization problem for different constants λ over 500 runs. For comparison, the horizontal green line indicates the median over all SOBI estimates for $\lambda = 0$.

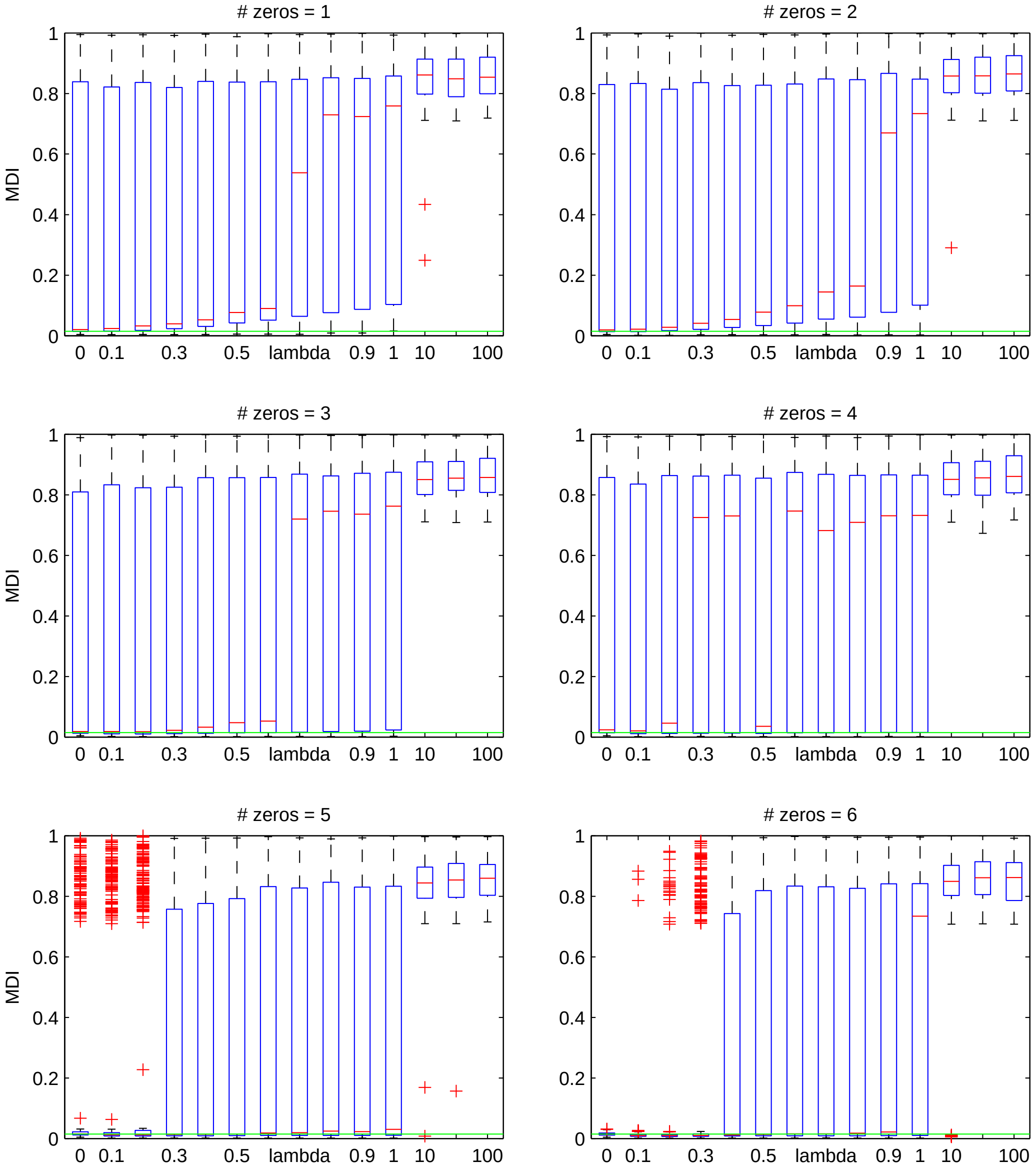

Figure A.2: Joint diagonalization with penalty term (MDI), model (ii). The data is generated using the ARMA-model (ii) with a time series length of $T = 10\,000$. The true mixing matrix contains $1, 2, \dots, 6$ zero-entries. The plots show the numerical optimization performance of the penalized joint diagonlization problem for different constants λ over 500 runs. For comparison, the horizontal green line indicates the median over all SOBI estimates for $\lambda = 0$.

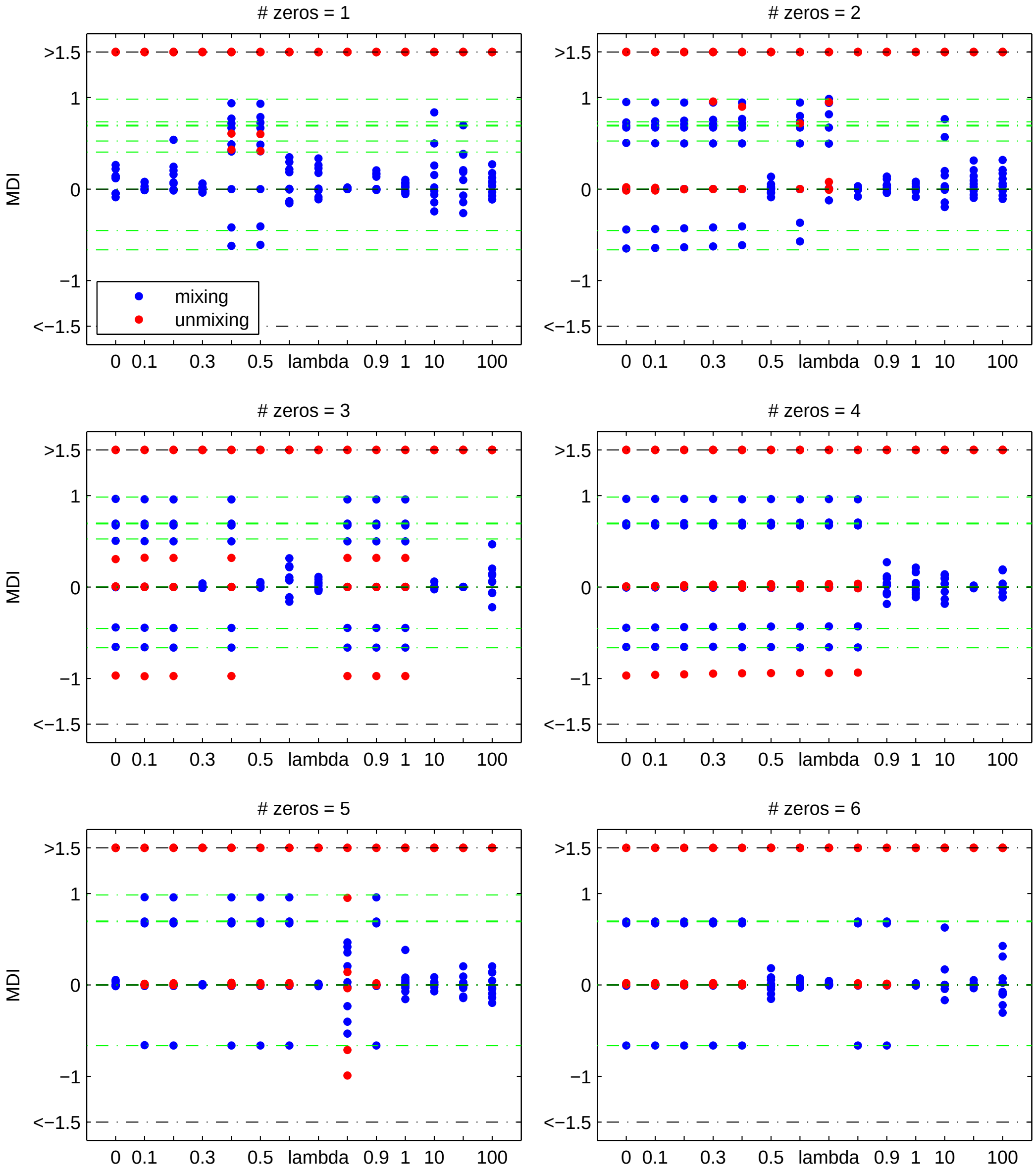

Figure A.3: Joint diagonalization with penalty term (mixing matrix). The data is generated using the AR(4)-model (i) with a time series length of $T = 10\,000$. The true mixing matrix contains $1, 2, \ldots, 6$ zero-entries. The plots show all entries of the mixing estimates (blue) and the corresponding unmixing estimates (red) from numerical optimization of the penalized joint diagonlization problem where we consider different constants λ. The horizontal green lines indicate the SOBI mixing estimates for $\lambda = 0$.

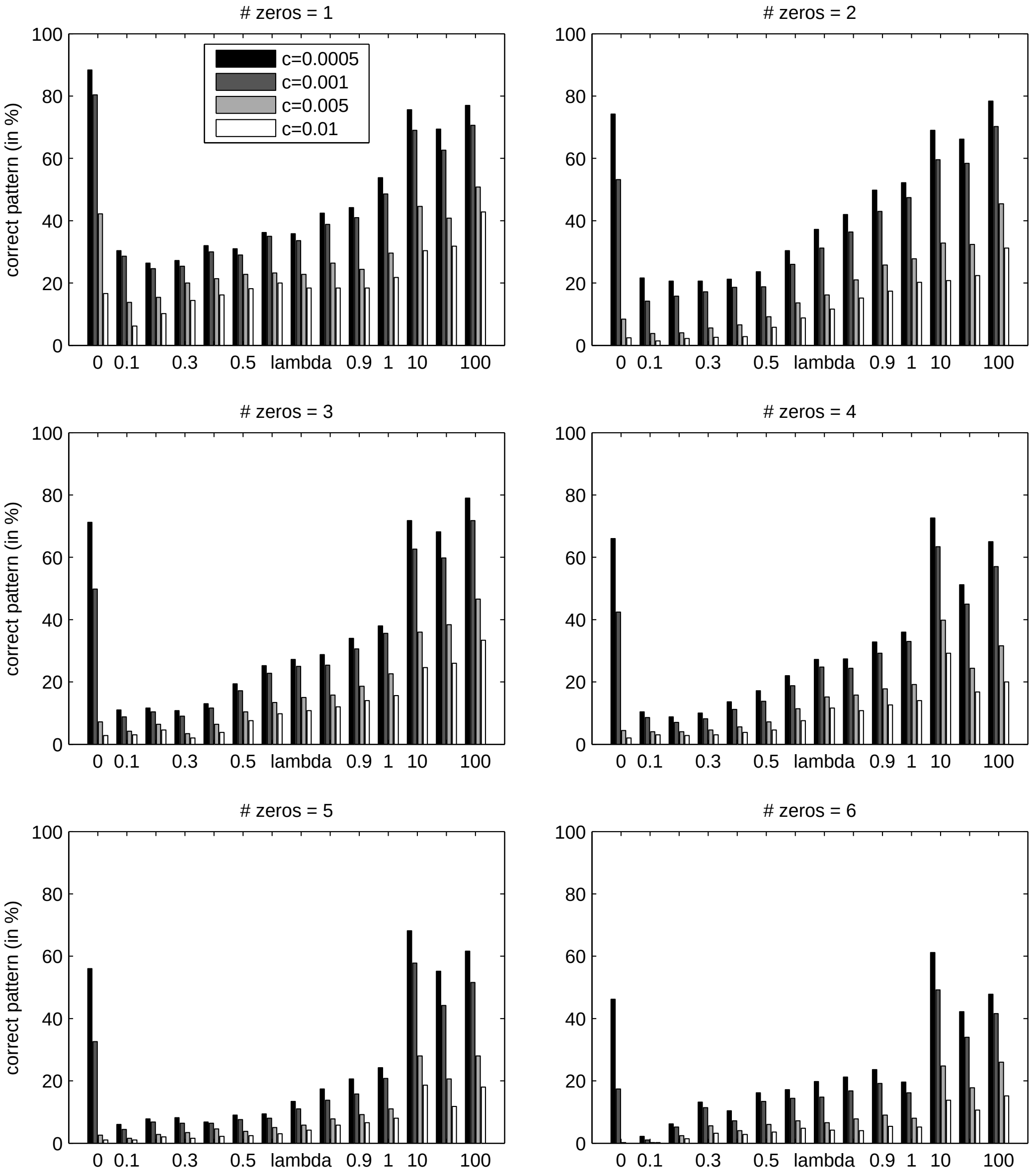

Figure A.4: Pattern identification using penalized optimization, model (i). The data is generated using the AR(4)-model (i) with a time series length of $T = 10\,000$. The true mixing matrix contains $1, 2, \ldots, 6$ zero-entries. To determine the true zero-pattern from the mixing estimates we consider different threshold values c. The plots show the percentage of correctly determined zero-patterns over 500 runs dependent on the threshold c and the penalizing constant λ.

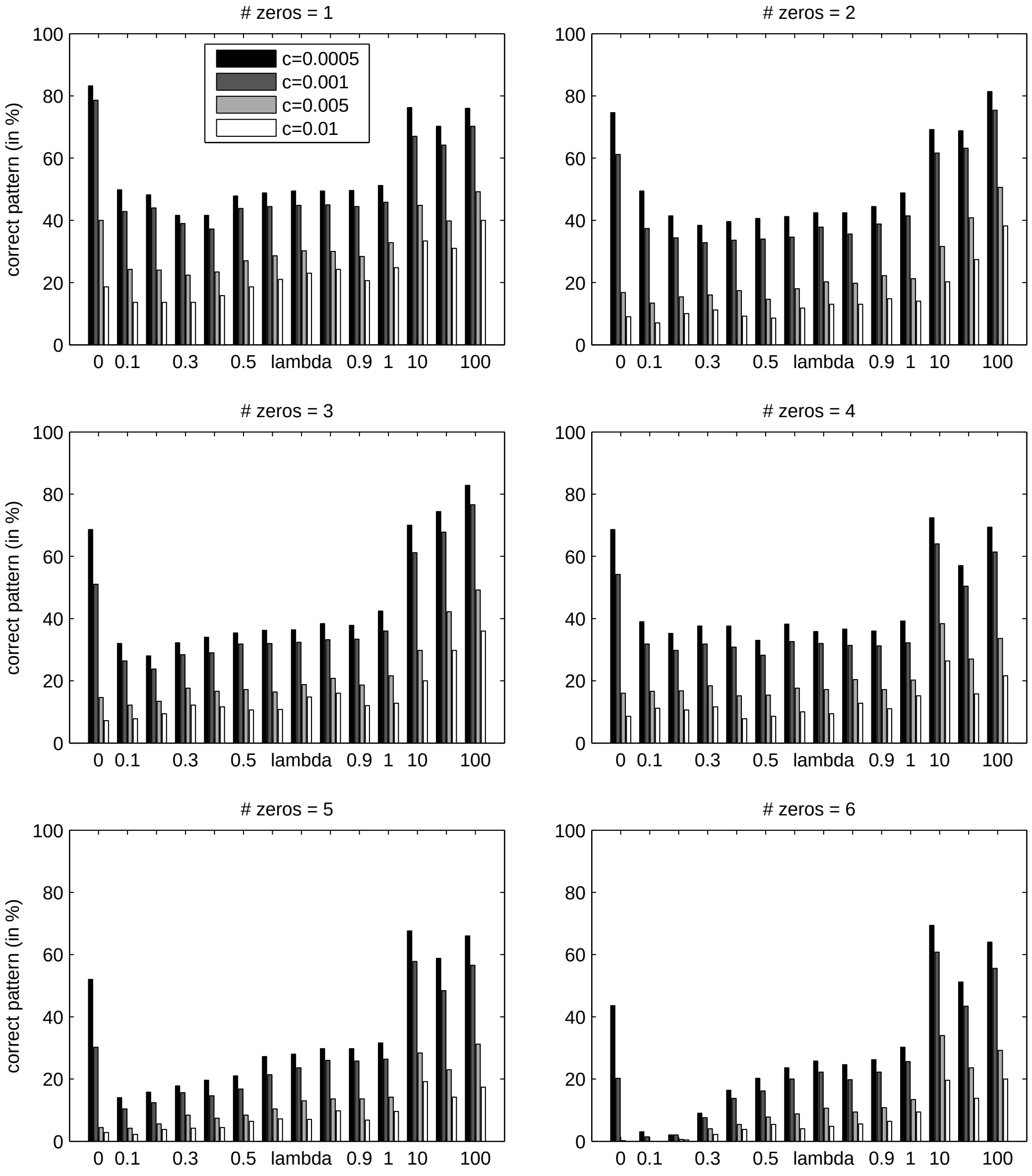

Figure A.5: Pattern identification using penalized optimization, model (ii). The data is generated using the ARMA-model (ii) with a time series length of $T = 10\,000$. The true mixing matrix contains $1, 2, \ldots, 6$ zero-entries. To determine the true zero-pattern from the mixing estimates we consider different threshold values c. The plots show the percentage of correctly determined zero-patterns over 500 runs dependent on the threshold c and the penalizing constant λ.

Appendix B

Derivation of parameter updates for emGrade

In Section 6.3.2 we introduced an expectation-maximization scheme to perform parameter inference in the model of `emGrade`. In the E-step we maximize the expected complete data log-likelihood $E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S} \mid \theta)\big]$ with respect to all model parameters $\theta = (A, \mu, \sigma^2, D)$. In the following we provide the derivation of the update rules for the parameters A, μ and σ^2.

Let therefore $A_\mu = (A, \mu)$ and $\boldsymbol{s}_*(i) = (\boldsymbol{s}(i)', 1)'$ for $i = 1, \ldots, N$. Let further $\mathrm{Exx} = \sum_{i=1}^N \boldsymbol{x}(i)\boldsymbol{x}(i)'$, $\mathrm{Esx} = \sum_{i=1}^N E[\boldsymbol{s}_*(i)]\boldsymbol{x}(i)'$, and $\mathrm{Ess} = \sum_{i=1}^N E[\boldsymbol{s}_*(i)\boldsymbol{s}_*(i)']$. The part of $E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S} \mid \theta)\big]$ dependent on the parameters A, μ and σ^2 is given in (6.8); with the above definitions it simplifies to

$$\begin{aligned}
&E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X} \mid \boldsymbol{S}, A, \mu, \sigma^2)\big] \\
&\quad = -\frac{Np}{2}\ln(2\pi) - \frac{Np}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^N \Big[\mathrm{Tr}\left(E[\boldsymbol{x}(i)\boldsymbol{x}(i)']\right) \\
&\qquad - 2\,\mathrm{Tr}\left(E[\boldsymbol{s}_*(i)\boldsymbol{x}(i)']\,A_\mu\right) + \mathrm{Tr}\left(E[\boldsymbol{s}_*(i)\boldsymbol{s}_*(i)']\,A_\mu' A_\mu\right)\Big] \\
&\quad = -\frac{Np}{2}\ln(2\pi) - \frac{Np}{2}\ln(\sigma^2) \\
&\qquad - \frac{1}{2\sigma^2}\big[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}\,A_\mu) + \mathrm{Tr}(\mathrm{Ess}\,A_\mu' A_\mu)\big]
\end{aligned}$$

Petersen & Pedersen (2008) provided useful formulas for the derivation of matrices. With this, we determine the partial derivatives of the above term with respect to A_μ and σ^2 and obtain

$$\begin{aligned}\frac{\partial}{\partial A_\mu}\,E_{S|X,\theta}[\,\ell_c\,] &= -\frac{1}{2\sigma^2}\big[-2(\mathrm{Esx})' + A_\mu((\mathrm{Ess})' + \mathrm{Ess})\big]\\ &= -\frac{1}{\sigma^2}[-(\mathrm{Esx})' + A_\mu \mathrm{Ess}\,]\\ \frac{\partial}{\partial \sigma^2}\,E_{S|X,\theta}[\,\ell_c\,] &= -\frac{Np}{2\sigma^2} + \frac{1}{2\sigma^4}\big[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}A_\mu) + \mathrm{Tr}(\mathrm{Ess}A_\mu' A_\mu)\big]\end{aligned}$$

A necessary condition for extremal point is that the partial derivates equal zero. With this, the first expression yields an update rule for A_μ

$$\begin{aligned} 0 \ &= -(\mathrm{Esx})' + A_\mu \mathrm{Ess}\\ \Leftrightarrow \quad A_\mu &= (\mathrm{Esx})'(\mathrm{Ess})^{-1}\end{aligned}$$

Using this updated value A_μ we get from the second expression an update rule for σ^2

$$\begin{aligned} 0 \ &= Np\,\sigma^2 - \big[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}A_\mu) + \mathrm{Tr}(\mathrm{Ess}A_\mu' A_\mu)\big]\\ \Leftrightarrow \quad \sigma^2 &= \frac{1}{Np}\big[\mathrm{Tr}(\mathrm{Exx}) - 2\mathrm{Tr}(\mathrm{Esx}A_\mu) + \mathrm{Tr}(\mathrm{Ess}A_\mu' A_\mu)\big]\end{aligned}$$

The validation of these possible extremal points using higher-order partial derivatives is not straight-forward. The reason is the multi-dimensionality in form of matrices. In praxis we therefore test whether a step into the proposed direction leads to an increase of the cost-function $E_{S|X,\theta}\big[\ln \mathrm{p}(\boldsymbol{X}, \boldsymbol{S} \mid \theta)\big]$; this was always true in our simulations.

References

Abbi, R., El-Darzi, E., Vasilakis, C. & Millard, P. (2008). Analysis of stopping criteria for the em algorithm in the context of patient grouping according to length of stay. In *4th International IEEE Conference on Intelligent Systems (IS)*, vol. 1, 3–9, IEEE. 38, 100

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716–723. 39

Albera, L., Ferréol, A., Comon, P. & Chevalier, P. (2004). Blind identification of overcomplete mixtures of sources (BIOME). *Linear algebra and its applications*, **391**, 3–30. 42

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. & Walter, P. (2008). *Molecular biology of the cell*, vol. 5. Garland Science. 13

Amari, S.I., Cichocki, A., Yang, H.H. *et al.* (1996). A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, 757–763. 53

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. & Sherlock, G. (2000). Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, **25**, 25–29. 14

Banerjee, O., Ghaoui, L.E., d'Aspremont, A. & Natsoulis, G. (2006). Convex optimization techniques for fitting sparse gaussian graphical models. In *Proc. 23rd International Conference on Machine Learning*, 89–96, ACM. 5

Beckmann, C.F. & Smith, S.M. (2004). Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, **23**, 137–152. 58

Belouchrani, A. & Cardoso, J.F. (1994). Maximum likelihood source separation for discrete sources. In *Proc. 7th European Signal Processing Conference (EUSIPCO 1994)*, Citeseer. 47

Belouchrani, A., Abed-Meraim, K., Cardoso, J.F. & Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, **45**, 434–444. 3, 21, 49

Berg, J.M., Tymoczko, J.L. & Stryer, L. (2011). *Biochemistry: international edition*. WH Freeman & Company Limited. 13

Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P. & Plemmons, R.J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, **52**, 155–173. 53

Bolstad, B.M., Irizarry, R.A., Åstrand, M. & Speed, T.P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**, 185–193. 114

Brockwell, P.J. & Davis, R.A. (2009). *Time series: theory and methods*. Springer. 27, 29

Bunse-Gerstner, A., Byers, R. & Mehrmann, V. (1993). Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, **14**, 927–949. 49

Calvano, S.E., Xiao, W., Richards, D.R., Felciano, R.M., Baker, H.V., Cho, R.J., Chen, R.O., Brownstein, B.H., Cobb, J.P., Tschoeke, S.K. *et al.* (2005). A network-based analysis of systemic inflammation in humans. *Nature*, **437**, 1032–1037. 113, 114

Cao, X.R. & Liu, R.W. (1996). General approach to blind source separation. *IEEE Transactions on Signal Processing*, **44**, 562–571. 42

Cardoso, J.F. (1990). Localisation et identification par la quadricovariance. *Traitement du signal*, **7**, 397–406. 46

Cardoso, J.F. & Souloumiac, A. (1993). Blind beamforming for non-gaussian signals. In *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, 362–370, IET. 46

Cardoso, J.F. & Souloumiac, A. (1996). Jacobi angles for simultaneous diagonalization. *SIAM journal on matrix analysis and applications*, **17**, 161–164. 49

Choudrey, R.A. & Roberts, S.J. (2001). Flexible bayesian independent component analysis for blind source separation. In *Proc. 3rd International Conference on Independent Component Analysis and Signal Separation (ICA 2001)*, 90–95. 4, 42

Cichocki, A. & Amari, S.I. (2002). *Adaptive blind signal and image processing: learning algorithms and applications*, vol. 1. John Wiley & Sons. 2

Collins, F.S., Morgan, M. & Patrinos, A. (2003). The human genome project: lessons from large-scale biology. *Science*, **300**, 286–290. 13

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, **36**, 287–314. 45

Crick, F. (1970). Central dogma of molecular biology. *Nature*, **227**, 561–563. 13

Dahm, R. (2008). Discovering DNA: Friedrich Miescher and the early years of nucleic acid research. *Human genetics*, **122**, 565–581. 13

Eastaway, M., Steinbauer, J., Qian, A. & Dayal, A. (2015). Blind source separation via ICA: Implementation. OpenStax-CNX, January 9, 2015, http://legacy.cnx.org/content/m15639/1.2/. 2

Eriksson, J. & Koivunen, V. (2004). Identifiability, separability, and uniqueness of linear ica models. *Signal Processing Letters, IEEE*, **11**, 601–604. 45

Fischer, G. (2013). *Lineare Algebra: Eine Einführung für Studienanfänger*. Springer Spektrum, 18th edn. 19

Fisher, R.A. (1922). On the interpretation of $\chi 2$ from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 87–94. 128

Fong, Y.M., Marano, M.A., Moldawer, L.L., Wei, H., Calvano, S.E., Kenney, J.S., Allison, A.C., Cerami, A., Shires, G.T. & Lowry, S.F. (1990). The acute splanchnic and peripheral tissue metabolic response to endotoxin in humans. *Journal of Clinical Investigation*, **85**, 1896. 114

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, **303**, 799–805. 30

Friedman, N., Linial, M., Nachman, I. & Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of computational biology*, **7**, 601–620. 5, 90

Gao, Y. & Church, G. (2005). Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, **21**, 3970–3975. 52, 53

Gautier, L., Cope, L., Bolstad, B.M. & Irizarry, R.A. (2004). Affy—analysis of affymetrix genechip data at the probe level. *Bioinformatics*, **20**, 307–315. 114

Grimmett, G. & Stirzaker, D. (2001). *Probability and Random Processes*. Oxford University Press. 22

Grove, C.A. & Walhout, A.J.M. (2008). Transcription factor functionality and transcription regulatory networks. *Molecular BioSystems*, **4**, 309–314. 16

Halder, G., Callaerts, P. & Gehring, W.J. (1995). Induction of ectopic eyes by targeted expression of the eyeless gene in Drosophila. *Science*, **267**, 1788–1792. 15

Hansen, L.K. (2000). Blind separation of noisy image mixtures. In *Advances in Independent Component Analysis*, 161–181, Springer. 47

Hartemink, A.J., Gifford, D.K., Jaakkola, T. & Young, R.A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pacific symposium on biocomputing*, vol. 6, 266. 90

He, Z., Xie, S., Zhang, L. & Cichocki, A. (2008). A note on lewicki-sejnowski gradient for learning overcomplete representations. *Neural computation*, **20**, 636–643. 42

Held, L. (2008). Methoden der statistischen inferenz. *Likelihood und Bayes. Heidelberg: Spektrum Akad. Verl.* 34

Højen-Sørensen, P.A.d.F.R., Winther, O. & Hansen, L.K. (2002). Mean-field approaches to independent component analysis. *Neural Computation*, **14**, 889–918. 47

Hong, B. & Calhoun, V.D. (2004). Source density driven adaptive independent component analysis approach for fMRI signal analysis. In *Proc. 14th IEEE Signal Processing Society Workshop: Machine Learning for Signal Processing*, 463–472, IEEE. 58

Hyvärinen, A. & Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural computation*, **9**, 1483–1492. 46

Hyvarinen, A., Karhunen, J. & Oja, E. (2002). Independent component analysis. *Studies in informatics and control*, **11**, 205–207. 4, 45

Illner, K., Fuchs, C. & Theis, F.J. (2012). Blind source separation using latent gaussian graphical models. In *Proc. 9th International Workshop on Computational Systems Biology (WCSB 2012)*, 43–46.

Illner, K., Fuchs, C. & Theis, F.J. (2014a). Bayesian blind source separation applied to the lymphocyte pathway. In *Proc. 21st International Conference on Computational Statistics (COMPSTAT 2014)*, 625–632.

Illner, K., Fuchs, C. & Theis, F.J. (2014b). Bayesian blind source separation for data with network structure. *Journal of Computational Biology*, **21**, 855–865.

Illner, K., Miettinen, J., Fuchs, C., Taskinen, S., Nordhausen, K., Oja, H. & Theis, F.J. (2015). Model selection using limiting distributions of second-order blind source separation algorithms. *Signal Processing*, **113**, 95–103.

Ilmonen, P., Nordhausen, K., Oja, H. & Ollila, E. (2010). A new performance index for ICA: properties, computation and asymptotic analysis. In *Latent Variable Analysis and Signal Separation*, 229–236, Springer. 54

Ilmonen, P., Oja, H. & Serfling, R. (2012). On invariant coordinate system (ICS) functionals. *International Statistical Review*, **80**, 93–110. 44

Imoto, S., Goto, T., Miyano, S. *et al.* (2002). Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. In *Pacific Symposium on Biocomputing*, vol. 7, 175–186, World Scientific. 90, 94

JEFFREYS, H. (1961). Theory of probability. 40

JEONG, J., LI, L., LIU, Y., NEPHEW, K.P., HUANG, T.H.M. & SHEN, C. (2010). An empirical Bayes model for gene expression and methylation profiles in antiestrogen resistant breast cancer. *BMC medical genomics*, **3**, 55. 139

JOTHI, R., BALAJI, S., WUSTER, A., GROCHOW, J.A., GSPONER, J., PRZYTYCKA, T.M., ARAVIND, L. & BABU, M.M. (2009). Genomic analysis reveals a tight link between transcription factor dynamics and regulatory network architecture. *Molecular systems biology*, **5**. 17

JOYCE, C.A., GORODNITSKY, I.F. & KUTAS, M. (2004). Automatic removal of eye movement and blink artifacts from eeg data using blind component separation. *Psychophysiology*, **41**, 313–325. 2

KANEHISA, M., GOTO, S., SATO, Y., FURUMICHI, M. & TANABE, M. (2011). KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, **40**, D109–D114. 14

KARVANEN, J., ERIKSSON, J. & KOIVUNEN, V. (2000). Maximum likelihood estimation of ICA model for wide class of source distributions. In *Neural Networks for Signal Processing X, 2000. Proc. of the 2000 IEEE Signal Processing Society Workshop*, vol. 1, 445–454, IEEE. 54

KASS, R.E. & RAFTERY, A.E. (1995). Bayes factors. *Journal of the american statistical association*, **90**, 773–795. 40

KIM, H. & PARK, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, **23**, 1495–1502. 52, 53

KOWARSCH, A., BLÖCHL, F., BOHL, S., SAILE, M., GRETZ, N., KLINGMÜLLER, U. & THEIS, F.J. (2010). Knowledge-based matrix factorization temporally resolves the cellular responses to il-6 stimulation. *BMC bioinformatics*, **11**, 585. 3, 5, 6, 8, 13, 41, 51, 89, 91, 94, 96, 115, 136

KRUMSIEK, J., SUHRE, K., ILLIG, T., ADAMSKI, J. & THEIS, F.J. (2011). Gaussian graphical modeling reconstructs pathway reactions from high-throughput metabolomics data. *BMC Systems Biology*, **5**, 21. 5, 33

KRUMSIEK, J., SUHRE, K., ILLIG, T., ADAMSKI, J. & THEIS, F.J. (2012). Bayesian independent component analysis recovers pathway signatures from blood metabolomics data. *Journal of proteome research*, **11**, 4120–4131. 3

LAURITZEN, S.L. (1996). *Graphical models*. Oxford University Press. 30, 32, 34

LEE, D.D. & SEUNG, H.S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**, 788–791. 52

LEE, D.D. & SEUNG, H.S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, 556–562. 53

LEE, T.W., LEWICKI, M.S., GIROLAMI, M. & SEJNOWSKI, T.J. (1999). Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters*, **6**, 87–90. 2

LIN, Q.H., YIN, F.L., MEI, T.M. & LIANG, H. (2006). A blind source separation based method for speech encryption. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **53**, 1320–1328. 2, 42

LÓPEZ, G.P.M., LOZANO, H.M., SÁNCHEZ, F. & MORENO, L.N.O. (2011). Blind source separation of audio signals using independent component analysis and wavelets. In *21st International Conference on Electrical Communications and Computers (CONIELECOMP 2011)*, 152–157, IEEE. 2

LUTTER, D., UGOCSAI, P., GRANDL, M., ORSO, E., THEIS, F.J., LANG, E.W. & SCHMITZ, G. (2008). Analyzing M-CSF dependent monocyte/macrophage differentiation: Expression modes and meta-modes derived from an independent component analysis. *BMC Bioinformatics*, **9**, 100. 3

MCKEOWN, M.J., MAKEIG, S., BROWN, G.G., JUNG, T.P., KINDERMANN, S.S., BELL, A.J. & SEJNOWSKI, T.J. (1998). Analysis of fMRI data by blind separation into independent spatial components. *Human Brain Mapping*, **6**, 160–188. 58

MCLACHLAN, G. & KRISHNAN, T. (2007). *The EM algorithm and extensions*, vol. 382. John Wiley & Sons. 34

MIETTINEN, J., NORDHAUSEN, K., OJA, H. & TASKINEN, S. (2012). Statistical properties of a blind source separation estimator for stationary time series. *Statistics & Probability Letters*. 58

MIETTINEN, J., NORDHAUSEN, K., OJA, H. & TASKINEN, S. (2013). BSSasymp: Asymptotic covariance matrices of some BSS mixing and unmixing matrix estimates. R-package version 1.0-0. http://cran.r-project.org/web/packages/BSSasymp. 77

MIETTINEN, J., NORDHAUSEN, K., OJA, H. & TASKINEN, S. (2014). Deflation-based separation of uncorrelated stationary time series. *Journal of Multivariate Analysis*, **123**, 214–227. 7, 8, 57, 59, 60, 61, 74, 75

MIETTINEN, J., ILLNER, K., NORDHAUSEN, K., OJA, H., TASKINEN, S. & THEIS, F.J. (2015). Separation of uncorrelated stationary time series using autocovariance matrices. Accepted for *Journal of Time Series Analysis.* 7, 8, 57, 59, 60, 63, 74, 75

Murphy, K. *et al.* (2001). The bayes net toolbox for matlab. *Computing science and statistics*, **33**, 1024–1034. 99

Neapolitan, R.E. *et al.* (2004). *Learning bayesian networks*, vol. 38. Prentice Hall Upper Saddle River. 99

Nordhausen, K., Oja, H. & Ollila, E. (2008). Robust independent component analysis based on two scatter matrices. *Austrian Journal of Statistics*, **37**, 91–100. 54

Nordhausen, K., Ollila, E. & Oja, H. (2011). On the performance indices of ica and blind source separation. In *12th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2011)*, 486–490, IEEE. 53

Olby, R.C. (1974). *The path to the double helix: the discovery of DNA*. Courier Dover Publications. 13

Ollila, E. & Kim, H.J. (2011). On testing hypotheses of mixing vectors in the ica model using fastica. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 325–328. 58, 78

Parra, L. & Spence, C. (2000). Convolutive blind separation of non-stationary sources. *IEEE Transactions on Speech and Audio Processing*, **8**, 320–327. 43

Pauca, V.P., Shahnaz, F., Berry, M.W. & Plemmons, R.J. (2004). Text mining using non-negative matrix factorizations. In *SIAM International Conference on Data Mining (SDM 2004)*, 452–456, SIAM. 52

Pearson, H. (2006). Genetics: what is a gene? *Nature*, **441**, 398–401. 14

Pe'er, D., Regev, A., Elidan, G. & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17**, S215–S224. 5, 90

Petersen, K.B. & Pedersen, M.S. (2008). The matrix cookbook. *Technical University of Denmark*, 7–15. 149

Sass, S., Buettner, F., Müller, N.S. & Theis, F.J. (2013). A modular framework for gene set analysis integrating multilevel omics data. *Nucleic acids research*, **41**, 9622–9633. 139

Schäfer, J. & Strimmer, K. (2005). Learning large-scale graphical gaussian models from genomic data. In *AIP Conference Proceedings*, vol. 776, 263. 5, 33

Schiessl, I., Schöner, H., Stetter, M., Dima, A. & Obermayer, K. (2000). Regularized second order source separation. In *Proc. 2nd International Workshop on Independent Component Analysis and Blind Signal Separation*, 111–116, Citeseer. 49

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, **6**, 461–464. 39

SEGAL, E., SHAPIRA, M., REGEV, A., PE'ER, D., BOTSTEIN, D., KOLLER, D. & FRIEDMAN, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature genetics*, **34**, 166–176. 17

SMYTH, G.K. (2005). Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, 397–420, Springer. 114

SUN, W. & YUAN, Y.X. (2006). *Optimization theory and methods: nonlinear programming*, vol. 1. springer. 61

TESCHENDORFF, A.E., JOURNÉE, M., ABSIL, P.A., SEPULCHRE, R. & CALDAS, C. (2007). Elucidating the altered transcriptional programs in breast cancer using independent component analysis. *PLoS Computational Biology*, **3**, e161. 3

THEIS, F.J. (2003). *Geometric source separation: Algorithms and applications*. Ph.D. thesis, University de Granada. 46

THEIS, F.J. & GRUBER, P. (2005). On model identifiability in analytic postnonlinear ICA. *Neurocomputing*, **64**, 223–234. 45

THEIS, F.J. & INOUYE, Y. (2006). On the use of joint diagonalization in blind signal processing. In *IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, 4–pp, IEEE. 49

THEIS, F.J., LANG, E.W. & PUNTONET, C.G. (2004a). A geometric algorithm for overcomplete linear ICA. *Neurocomputing*, **56**, 381–398. 54

THEIS, F.J., MEYER-BAESE, A. & LANG, E.W. (2004b). Second-order blind source separation based on multi-dimensional autocovariances. In *Independent Component Analysis and Blind Signal Separation*, 726–733, Springer. 3, 49

THEIS, F.J., GRUBER, P., KECK, I.R. & LANG, E.W. (2008). A robust model for spatiotemporal dependencies. *Neurocomputing*, **71**, 2209–2216. 2, 3, 49

THEIS, F.J., MÜLLER, N.S., PLANT, C. & BÖHM, C. (2010). Robust second-order source separation identifies experimental responses in biomedical imaging. In *Latent Variable Analysis and Signal Separation*, 466–473, Springer. 2

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 267–288. 53

TIPPING, M.E. & BISHOP, C.M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **61**, 611–622. 4

TONG, L., SOON, V.C., HUANG, Y.F. & LIU, R. (1990). Amuse: A new blind identification algorithm. *IEEE International Symposium on Circuits and Systems*, 1784–1787. 3, 42, 48, 49, 50

VAQUERIZAS, J.M., KUMMERFELD, S.K., TEICHMANN, S.A. & LUSCOMBE, N.M. (2009). A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, **10**, 252–263. 16

WALHOUT, A.J.M. (2011). What does biologically meaningful mean? A perspective on gene regulatory network validation. *Genome biology*, **12**, 109. 17

WALHOUT, M., VIDAL, M. & DEKKER, J. (2013). *Handbook of systems biology: concepts and insights*. Academic Press. 13

WANG, W., CHAMBERS, J.A. & SANEI, S. (2003). A joint diagonalization method for convolutive blind separation of nonstationary sources in the frequency domain. In *Proc. 4th International Conference on Independent Component Analysis and Signal Separation (ICA 2003)*, 939–944. 43

WATSON, J.D. & CRICK, F.H.C. (1953). Molecular structure of nucleic acids. *Nature*, **171**, 737–738. 13

YANG, Z., ZHOU, G., WU, Z. & ZHANG, J. (2008). New method for signal encryption using blind source separation based on subband decomposition. *Progress in Natural Science*, **18**, 751–755. 2, 42

YEN, K., NARASIMHAN, S.D. & TISSENBAUM, H.A. (2011). DAF-16/Forkhead box O transcription factor: many paths to a single fork (head) in the road. *Antioxidants & redox signaling*, **14**, 623–634. 16

YEREDOR, A. (2002). Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, **50**, 1545–1553. 3, 49, 50

ZIBULEVSKY, M. (2003). Blind source separation with relative newton method. In *Proc. 4th International Conference on Independent Component Analysis and Signal Separation (ICA 2003)*, 897–902. 2

ZIEHE, A. & MÜLLER, K.R. (1998). TDSEP - an efficient algorithm for blind separation using time structure. In *Proc. International Conference on Artificial Neural Networks (ICANN 1998)*, 675–680, Citeseer. 49

ZIEHE, A., LASKOV, P., MÜLLER, K.R. & NOLTE, G. (2003). A linear least-squares algorithm for joint diagonalization. In *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2003)*, 469–474. 49