Independent Component Analysis and Slow Feature Analysis: Relations and Combination

D I S S E R T A T I O N

zur Erlangung des akademischen Grades
doctor rerum naturalium
(Dr. rer. nat.)
im Fach Physik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät I
der Humboldt-Universität zu Berlin

von
Herrn Dipl.-Phys. Tobias Blaschke
geboren am 02.10.1972 in Rüsselsheim

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Jürgen Mlynek

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät I:
Prof. Thomas Buckhout, PhD

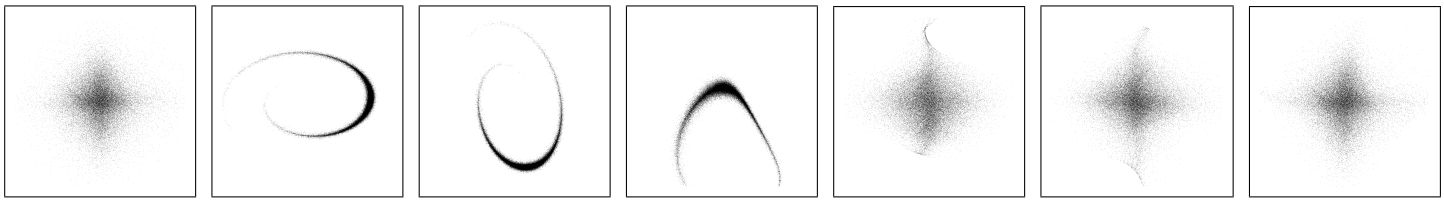
Gutachter:

1. Dr. Laurenz Wiskott
2. Prof. Dr. Klaus Obermayer
3. Prof. Dr. Lutz Schimansky-Geier

eingereicht am:	25. August 2004
Tag der mündlichen Prüfung:	02. Februar 2005

Independent Component Analysis and Slow Feature Analysis: Relations and Combination

Tobias Blaschke



Abstract

Within this thesis, we focus on the relation between independent component analysis (ICA) and slow feature analysis (SFA). To allow a comparison between both methods we introduce CuBICA2, an ICA algorithm based on second-order statistics only, i.e. cross-correlations. In contrast to algorithms based on higher-order statistics not only instantaneous cross-correlations but also time-delayed cross correlations are considered for minimization. CuBICA2 requires signal components with auto-correlation like in SFA, and has the ability to separate source signal components that have a Gaussian distribution. Furthermore, we derive an alternative formulation of the SFA objective function and compare it with that of CuBICA2. In the case of a linear mixture the two methods are equivalent if a single time delay is taken into account. The comparison can not be extended to the case of several time delays. For ICA a straightforward extension can be derived, but a similar extension to SFA yields an objective function that can not be interpreted in the sense of SFA. However, a useful extension in the sense of SFA to more than one time delay can be derived. This extended SFA reveals the close connection between the slowness objective of SFA and temporal predictability.

Furthermore, we combine CuBICA2 and SFA. The result can be interpreted from two perspectives. From the ICA point of view the combination leads to an algorithm that solves the nonlinear blind source separation problem. From the SFA point of view the combination of ICA and SFA is an extension to SFA in terms of statistical independence. Standard SFA extracts slowly varying signal components that are uncorrelated meaning they are statistically independent up to second-order. The integration of ICA leads to signal components that are more or less statistically independent.

Keywords:
Independent Component Analysis, Slow Feature Analysis, Nonlinear Blind Source Separation, Slowness

Zusammenfassung

Der Fokus dieser Dissertation liegt auf den Verbindungen zwischen ICA (Independent Component Analysis - Unabhängige Komponenten Analyse) und SFA (Slow Feature Analysis - Langsame Eigenschaften Analyse). Um einen Vergleich zwischen beiden Methoden zu ermöglichen wird CuBICA2, ein ICA Algorithmus basierend nur auf Statistik zweiter Ordnung, d.h. Kreuzkorrelationen, vorgestellt. Dieses Verfahren minimiert zeitverzögerte Korrelationen zwischen Signalkomponenten, um die statistische Abhängigkeit zwischen denselben zu reduzieren. Zusätzlich wird eine alternative SFA-Formulierung vorgestellt, die mit CuBICA2 verglichen werden kann. Im Falle linearer Gemische sind beide Methoden äquivalent falls nur eine einzige Zeitverzögerung berücksichtigt wird. Dieser Vergleich kann allerdings nicht auf mehrere Zeitverzögerungen erweitert werden. Für ICA lässt sich zwar eine einfache Erweiterung herleiten, aber ein ähnliche SFA-Erweiterung kann nicht im originären SFA-Sinne (SFA extrahiert die am langsamsten variierenden Signalkomponenten aus einem gegebenen Eingangssignal) interpretiert werden. Allerdings kann eine im SFA-Sinne sinnvolle Erweiterung hergeleitet werden, welche die enge Verbindung zwischen der Langsamkeit eines Signales (SFA) und der zeitlichen Vorhersehbarkeit desselben verdeutlich.

Im Weiteren wird CuBICA2 und SFA kombiniert. Das Resultat kann aus zwei Perspektiven interpretiert werden. Vom ICA-Standpunkt aus führt die Kombination von CuBICA2 und SFA zu einem Algorithmus, der das Problem der nichtlinearen blinden Signalquellentrennung löst. Vom SFA-Standpunkt aus ist die Kombination eine Erweiterung der standard SFA. Die standard SFA extrahiert langsam variierende Signalkomponenten die untereinander unkorreliert sind, dass heißt statistisch unabhängig bis zur zweiten Ordnung. Die Integration von ICA führt nun zu Signalkomponenten die mehr oder weniger statistisch unabhängig sind.

Schlagwörter:
Unabhängige Komponenten Analyse, Langsame Komponenten Analyse, Nicthlineare Blinde Signalquellentrennung, Langsamkeit

Acknowledgements

There are lots of people I would like to thank for a huge variety of reasons. First and foremost, I wish to thank my advisor, Laurenz Wiskott for his constant support and guidance throughout these years. Without his precise comments and questions this work would not have become like it is presented here. I will always remember: A (good) picture says more than 1000 formulas. I am also greatly indebted to Andreas Herz for creating an exciting research environment at the ITB, where I had the opportunity to benefit from the rich scientific life.

I am also grateful to Pietro Berkes for many fruitful discussions on SFA and ICA. He always listened to whatever mathematical issue I came up with. Furthermore I want to thank Tiziano Zito and Mathias Franzius my other *room mates*. There was always time for questions, discussion and for some espressi. I have also been inspired by Christian Michaelis and Thomas Voegtlin.

Special thanks go to Irina Erchova for relaxing coffee breaks, interesting discussions, and Russian music. Special thanks also to Susanne Schreiber for ice cream and help in various occasions. Of the other people at the ITB, I would especially like to thank Roberto Fernández Galán, Richard Kempter, Tim Gollisch, and Raphael Ritz for his helpful comments on my first manuscript. Thanks also to Jan Benda, Jürgen Neubauer, and Christian Zemlin: it was a pleasure to sing with you in the ITB choir. Furthermore, I would like to thank all other people at the ITB who helped creating an inspiring atmosphere at the institute. I am also indebted to the Volkswagen Foundation who supported this work.

I would like to thank Maren Gerhardt, for being there and for her support and patience in many situations; and of course Jaromir and Frida, the two most lovely kids, who sometimes reminded me of the really important things in live. I am grateful to Dana Berg, Bö Yehoash, an Karoline Körber for babysitting and beyond. I always enjoyed our weekends in the Uckermark.

Last not least, I want to thank my parents for their support throughout all these years. They helped me during my studies in many ways and where always there to give me advice.

Contents

List of Figures

List of Tables

1

Introduction

The Family Celebration Problem

Imagine you are a granddaddy, or a grandma if you like, at a family celebration. All of your near relatives, second-grade, and remote relatives are gathered together. You are having a good time. There is tasty food, nice people, good wine. Some of your remote relatives suddenly fetch their music instruments and start playing an old waltz.

For this special event you have put on your hearing aid. None of this fancy new hearing aids that can even switch between *classical music* and *single speaker* mode. You bought it a couple of years ago and since then it always worked well. At least if you are at home and listening to the television or your wife (husband).

But, here at the celebration: everyone is talking, the remote relatives are still playing tunes in three quarter time and your neighbor to the right, probably some second cousin, tries to draw your attention to his one year old crying baby.

Since your hearing aid amplifies every sound from any direction in an equivalent way all you hear is a *mixture* of all this sounds. You just do not know on which of this sounds to concentrate! There are just too many people talking, too loud music, too many signals coming from too many different directions. And of course the sweet red wine...

What you really need is a hearing aid that is able to separate the received mixture into its original source components. But, what is the clue that helps to discriminate different signals from different sources? How can we blindly separate the sources? The assumption we can make is that those signals are independent from each other. Independent means that different relatives will speak differently and different words, independently of all others. Adding the fact that different audio signals superimpose linearly we can show that the separation of the received mixture into independent components reveals the original source signals. In the following we will call the problem of finding the source signal components *Blind Source Separation (BSS)* and the method that exactly performs this separation task *Independent Component Analysis (ICA)*.

The first two parts of this work addresses the linear blind separation task as stated above and independent component analysis, its solution. To be more concise, after an introductionary part two ICA algorithms are introduced and studied. Additionally, their performance is underlined with some example simulations. We do not want to give a full overview over the complex research field of ICA. Instead, we want to concentrate on those algorithms and their mathematical foundation. A thorough introduction to ICA is given in [Lee, 1998], [Hyvärinen et al., 2001b] and [Cichocki and Amari, 2002].

ICA is a relatively young research topic. The first publication on ICA, written by Herault and Jutten, dates back to 1986. Comon [1994b] was the first to formulate ICA in a consistent mathematical framework. However, ICA became popular with the paper by Bell and Sejnowski [1995], where they analyzed ICA from an information maximization point of view (for a detailed history refer to Chapter 4. Since then it has

turned out to be a useful method in a number of application areas, including

- medical data analysis (e.g. EEG, MEG, fMRI) (e.g. [Jung et al., 2000; McKeown et al., 1998]),
- computational neuroscience (e.g. [Doi et al., 2003; Hyvärinen et al., 2003, 2001a])
- bioinformatics (e.g. [Liebermeister, 2002; Scholz et al., 2004]),
- automated music analysis (e.g. [Abdallah and Plumbley, 2003; Feng et al., 2002]).

In this work, we concentrate on cumulant-based algorithms, especially two algorithms based on higher-order cumulants and on second-order cumulants respectively. The use of higher-order cumulants in ICA has been around since the work by Comon [1994a] while the second-order cumulant-approach is due to Molgedey and Schuster [1994].

Hmm! After reading the first two parts of this work you visit your preferable hearing aid seller and ask for a hearing aid with integrated ICA algorithm. But, unfortunately, no one at the shop has ever heard something called *independent component analysis*. So you will have to wait until someone recognizes the market chance of such devices and starts to do research in that direction (there are actually attempts to use ICA in hearing aids).

What, if the voices, sounds and noise you hear are superimposed nonlinearly rather than linearly? What would be the clue to separate the source signal components in this case? Of course, it is unlikely that you will perceive this kind of mixture at the family celebration. However, there are situations where you actually observe nonlinear mixtures, for example in the field of biomedical data recording [Ziehe et al., 2000]. As we will see, this more general problem needs additional assumptions about the source signal. The independence assumption made about the source signal components in the linear case is not sufficient to extract them from a nonlinear mixture. At this point slow feature analysis (SFA) comes into play. In general, the goal of SFA is to extract a slowly varying signal out of the input signal (e.g. the mixture of sounds, voices, etc. as described above), where the input signal is usually more quickly varying as the extracted one (a detailed description of SFA will be given in Chapter 5). The integration of the independence objective of ICA and the slowness objective of SFA leads to an algorithm that is on one hand able to solve the nonlinear BSS problem. On the other hand it is an extension of SFA in the sense that the SFA output signal has statistically independent components. Studying the properties of this nonlinear BSS algorithm as well as the relations between ICA and SFA in general will be the main topic of the third part of this work.

Overview

Before we start with some basic concepts in higher-order statistics let us briefly give an overview over the different parts of this work. Part I lays the foundation of the fundamental concepts that build the basis for linear and non-linear independent component analysis and slow feature analysis. It starts with an introduction to the concept of higher-order statistics, mainly expressed in terms of cumulants and moments and their properties, in Chapter 2. Chapter 3 establishes mutual information as a fundamental measure of statistical independence, building the basis for several methods performing independent component analysis respectively blind source separation. These two concepts are then briefly described in the following chapter and a historical overview is given. Also Chapter 5 completes the first part with a review of the principles underlying slow feature analysis.

The second part of this thesis introduces two algorithms performing linear independent component analysis/blind source separation. The first, CuBICA, is an ICA algorithm based on higher-order statistics involving cumulants of order three and four (Chapter 6). The second algorithm, CuBICA2, is based on second-order statistics (Chapter 7).

The last part of this work is dedicated to the combination of SFA and ICA. In Chapter 8 SFA and ICA are compared and possible similarities and relations are pointed out. Using these relations a nonlinear BSS method is introduced in Chapter 9 that combines the two concepts of independence and slowness. The last chapter provides a summary and a discussion of the results and suggestions for future research. The

Appendices B.1 to D contain the derivation of mathematical constants that are not necessary for the reader to follow the main plot of this thesis.

Part I

Basic Concepts

2

Statistics

This chapter provides some basics in statistics that are needed to understand the second part of this thesis. We start with the definition of higher-order moments and cumulants in Section 2.2. Furthermore, we derive the relation between moments and cumulants and list some of their properties that are necessary in order to perform independent component analysis. The connection between the probability density function and cumulants is established in Section 2.3. Section 2.4 points out that the shape of the probability density function can be described by cumulants up to fourth order. Some example probability density functions and their moments and cumulants up to fourth order are given in Section 2.4.1. The chapter concludes with a short summary in Section 2.5.

2.1 Characteristic Functions

Definition 2.1.1 (Expectation) *Given a scalar random variable X with probability density function (p.d.f.) $p_X(x)$, the expectation of the function $f(x)$ is defined by*

$$\langle f(x) \rangle := \int_{-\infty}^{\infty} f(x)\, p_X(x)\, dx\,. \tag{2.1}$$

If the distribution of x is discrete, the integral is replaced by a sum. In the multivariate case the integral over $\mathbb{R}^1$ is replaced by an integral over $\mathbb{R}^p$.

Definition 2.1.2 (First Characteristic Function) *The first characteristic function $L_X(\omega)$ of the scalar random variable X is defined as the Fourier transform of the probability density function $p_X(x)$:*

$$L_X(\omega) := \int_{-\infty}^{\infty} \exp(\mathrm{i}\omega x)\, p_X(x)\, \mathrm{d}x\,, \tag{2.2}$$

where $\mathrm{i} = \sqrt{-1}$ and ω is the transformed variable (frequency) corresponding to x.

Using Definition (2.1) $L_X(\omega)$ can be written as the expected value of $\exp(\mathrm{i}\omega x)$

$$L_X(\omega) = \langle \exp(\mathrm{i}\omega x) \rangle\,. \tag{2.3}$$

In the multivariate case $L_{\boldsymbol{X}}(\boldsymbol{\omega})$, with $\omega = [\omega_1, \omega_2, \ldots, \omega_N]$, is given by

$$L_X(\boldsymbol{\omega}) = \left\langle \exp\left(\mathrm{i}\boldsymbol{\omega}^T \mathbf{x}\right) \right\rangle\,. \tag{2.4}$$

Definition 2.1.3 (Second Characteristic Function) *The second characteristic function $K_X(\omega)$ is given by the natural logarithm of the first characteristic function*

$$K_X(\omega) = \ln(L_X(\omega)) . \tag{2.5}$$

In the multivariate case $K_{\boldsymbol{X}}(\boldsymbol{\omega})$ is defined analogous to (2.4) as

$$K_X(\boldsymbol{\omega}) = \ln(L_X(\boldsymbol{\omega})) . \tag{2.6}$$

2.2 Moments and Cumulants

The concept of Moments was taken into Statistics from Mechanics by Karl Pearson when he treated the frequency-curve as the sheet enclosed by the curve and the horizontal axis (e.g. [Pearson, 1893]).

Cumulants where first defined by the Danish astronomer Thorvald Nicolai Thiele in his work from 1889 [Thiele, 1889] (the book is written in danish, a reprint of an English exposition is given in [Thiele, 1931]). Fisher rediscovered cumulants in 1929 [Fisher, 1929] and called them cumulative moment functions. He also described the computation of cumulants via sample averaging and noticed the superiority over moments. For a short overview of the history of cumulants and Thiele's contributions to statistics see [Lauritzen, 1999].

The properties of cumulants and moments are for example described in [McCullagh, 1987] and [De Lathauwer, 1997]. We briefly list the properties that are most important to us. They are the basis for all independent component analysis models throughout this thesis.

In the following we will denote a scalar random variable with X_j where we sometimes drop the index for simplicity. Vectorial random variables will be denoted by bold $\boldsymbol{X} = [X_1, , X_2, \ldots, X_N]$.

2.2.1 Moments

Definition 2.2.1 (Moments) *Moments $m_n^{(X)}$ of order n of a scalar random variable X are defined by*

$$m_n^{(X)} := \langle X^n \rangle := \int_{-\infty}^{\infty} x^n p_X(x)\,\mathrm{d}x , \tag{2.7}$$

This is just the special case of Equation (2.1) where $f(X)$ is the nth power of X. Cross-moments are defined correspondingly by

$$mom(X_1, \ldots, X_n) = \langle X_1 \cdot \ldots \cdot X_n \rangle = \int_{-\infty}^{\infty} (x_1 \cdot \ldots \cdot x_n)\, p_{\boldsymbol{X}}(x_1, \ldots, x_n)\,\mathrm{d}\mathbf{x} . \tag{2.8}$$

The first-order moment $m_1^{(X)}$ is called mean of x whereas the second-order moment is called correlation

$$m_2^{(X)} = \left\langle X^2 \right\rangle , \tag{2.9}$$

respectively cross-correlation

$$mom(X_j, X_k) = \left\langle X_j X_k \right\rangle . \tag{2.10}$$

The nth-order moments are just the partial derivatives of $L_X(\omega)$, defined in (2.3), evaluated at $\omega = 0$

$$m_n^{(X)} = \frac{1}{\mathrm{i}^n} \frac{\partial^n L_X(\omega)}{\partial \omega^n}\bigg|_{\omega=0} , \tag{2.11}$$

Cross-moments are derived correspondingly

$$mom(X_1, \ldots, X_n) = \frac{1}{\mathrm{i}^n} \frac{\partial^n L_X(\omega)}{\partial \omega_1 \ldots \partial \omega_n}\bigg|_{\omega=0} . \tag{2.12}$$

Using (2.12) we can write the expansion of the first characteristic function (2.4) as

$$L_X(\boldsymbol{\omega}) = 1 + \sum_{i=1}^{N}(\omega_i\, mom(X_i)) + \frac{1}{2!}\sum_{i=1}^{N}\sum_{j=1}^{N}(\omega_i\omega_j\, mom(X_i, X_j)) + \frac{1}{3!}\sum_{i}^{N}\sum_{j}^{N}\sum_{k}^{N}(\omega_i\omega_j\omega_k\, mom(X_i, X_j, X_k)) + \ldots . \tag{2.13}$$

Due to this fact the first characteristic function is often called the *moment generating function*.

2.2.2 Cumulants

Definition 2.2.2 (Cumulants) *Cumulants $c_n^{(X)}$ of order n are defined via the partial derivatives of the second characteristic function (2.5)*

$$c_n^{(X)} = \frac{1}{\mathrm{i}^n}\frac{\partial^n K_X(\omega)}{\partial\omega^n}\bigg|_{\omega=0} . \tag{2.14}$$

In the multivariate case the cross cumulants are defined by

$$cum(X_1, \ldots, X_n) = \frac{1}{\mathrm{i}^n}\frac{\partial^n K_X(\omega)}{\partial\omega_1 \ldots \partial\omega_n}\bigg|_{\omega=0} , \tag{2.15}$$

using the multivariate second characteristic function (2.6). Similar to Equation (2.13) we can expand the second characteristic function (2.6) using (2.15) to obtain

$$K_X(\boldsymbol{\omega}) = 1 + \sum_{i=1}^{N}(\omega_i\, cum(X_i)) + \frac{1}{2!}\sum_{i=1}^{N}\sum_{j=1}^{N}(\omega_i\omega_j\, cum(X_i, X_j)) + \frac{1}{3!}\sum_{i}^{N}\sum_{j}^{N}\sum_{k}^{N}(\omega_i\omega_j\omega_k\, cum(X_i, X_j, X_k)) + \ldots . \tag{2.16}$$

The second characteristic function is called cumulant generating function.

The first-order cumulant $c_1^{(X)}$ denotes the mean of X. The second-order cumulant corresponds to the variance

$$c_2^{(X)} = \left\langle (X - \langle X\rangle)^2 \right\rangle \tag{2.17}$$

of X. In the multivariate case second-order cross-cumulants are called covariance

$$cum(X_j, X_k) = \left\langle \left(X_j - \langle X_j\rangle\right)(X_k - \langle X_k\rangle)\right\rangle . \tag{2.18}$$

All second-order cumulants form the covariance matrix of $\boldsymbol{X} = [X_1, \ldots, X_n]^T$ given by

$$\mathbf{C}_2^{(\boldsymbol{X})} = \left\langle (\boldsymbol{X} - \langle \boldsymbol{X}\rangle)(\boldsymbol{X} - \langle \boldsymbol{X}\rangle)^T \right\rangle , \tag{2.19}$$

where the index 2 stands for cumulant of order two. In general, all auto- and cross-cumulants of nth-order together form an nth-order tensor [De Lathauwer, 1997] (the covariance matrix is a tensor of second-order).

Definition 2.2.3 (Standardized Cumulants, Skewness and Kurtosis) *A standardized cumulant of order n of a scalar random variable X is defined as*

$$\kappa_n^{(X)} = \left(\frac{1}{c_2^{(X)}}\right)^{\frac{n}{2}} c_n^{(X)} . \tag{2.20}$$

Skewness and kurtosis are defined as the standardized cumulants of order 3 *and* 4.

2.2.3 Relations between Moments and Cumulants

The Taylor expansion of $L_{\boldsymbol{X}}(\boldsymbol{\omega}) = \exp(K_{\boldsymbol{X}}(\boldsymbol{\omega}))$ is given by

$$L_{\boldsymbol{X}}(\boldsymbol{\omega}) = 1 + K_{\boldsymbol{X}}(\boldsymbol{\omega}) + \frac{1}{2!}K_{\boldsymbol{X}}(\boldsymbol{\omega})^2 + \ldots + \frac{1}{n!}K_{\boldsymbol{X}}(\boldsymbol{\omega})^n + \ldots . \tag{2.21}$$

We can now insert the expansion (2.16) of $K_{\boldsymbol{X}}(\boldsymbol{\omega})$ on the right hand side of (2.21) and the expansion of $L_{\boldsymbol{X}}(\boldsymbol{\omega})$ (2.13) on the left hand side. After combining terms and using symmetry we obtain the definition of the cumulants as functions of the moments by comparing the coefficients of both sides of Equation (2.21). We show the relations up to fourth-order

$$\begin{aligned}
cum(X_i) &= mom(X_i)\,, && (2.22)\\
cum(X_i,X_j) &= mom(X_i,X_j) - mom(X_i)\,mom(X_j)\,, && (2.23)\\
cum(X_i,X_j,X_k) &= mom(X_i,X_j,X_k) - mom(X_i)\,mom(X_j,X_k) - mom(X_j)\,mom(X_i,X_k)\\
&\quad - mom(X_k)\,mom(X_i,X_j) + 2mom(X_i)\,mom(X_j)\,mom(X_k)\,, && (2.24)\\
cum(X_i,X_j,X_k,X_l) &= mom(X_i,X_j,X_k,X_l) - mom(X_i)\,mom(X_j,X_k,X_l)\\
&\quad - mom(X_j)\,mom(X_i,X_k,X_l) - mom(X_k)\,mom(X_i,X_j,X_l)\\
&\quad - mom(X_l)\,mom(X_i,X_j,X_k) - mom(X_i,X_j)\,mom(X_k,X_l)\\
&\quad - mom(X_i,X_k)\,mom(X_j,X_l) - mom(X_i,X_l)\,mom(X_j,X_k)\\
&\quad + 2\,mom(X_i)\,mom(X_j)\,mom(X_k,X_l) + 2\,mom(X_i)\,mom(X_k)\,mom(X_j,X_l)\\
&\quad + 2\,mom(X_i)\,mom(X_l)\,mom(X_j,X_k) + 2\,mom(X_j)\,mom(X_k)\,mom(X_i,X_l)\\
&\quad + 2\,mom(X_j)\,mom(X_l)\,mom(X_i,X_k) + 2\,mom(X_k)\,mom(X_l)\,mom(X_i,X_j)\\
&\quad - 6\,mom(X_i)\,mom(X_j)\,mom(X_k)\,mom(X_l)\,. && (2.25)
\end{aligned}$$

If we consider all X_j to have zero mean ($m_1^{(X_j)} = 0$) this shortens to

$$\begin{aligned}
cum(X_i) &= 0\,, && (2.26)\\
cum(X_i,X_j) &= mom(X_i,X_j)\,, && (2.27)\\
cum(X_i,X_j,X_k) &= mom(X_i,X_j,X_k)\,, && (2.28)\\
cum(X_i,X_j,X_k,X_l) &= mom(X_i,X_j,X_k,X_l) - mom(X_i,X_j)\,mom(X_k,X_l)\\
&\quad - mom(X_i,X_k)\,mom(X_j,X_l) - mom(X_i,X_l)\,mom(X_j,X_k)\,. && (2.29)
\end{aligned}$$

Moments and cumulants are entirely equivalent, since first and second characteristic function carry the same information. Intuitively, for cumulants of a given order, redundant information of lower orders are subtracted, leading to insensitivity to lower order cumulants (see Sec. 2.2.9) as well as to different behavior with respect to partitioning and translation. This is in contrast to moments, which also contain information about lower order statistics.

2.2.4 Properties of Moments and Cumulants

At first sight, moments might seem more interesting than cumulants, due to their intuitive Definition (2.7). However, cumulants have a number of important properties that are not shared by moments and make them easier to handle. Thus they are used more frequently in higher-order statistics. We list some of the most interesting properties of moments and cumulants.

Property 2.2.1 (Scaling) *If the random variable X is multiplied by factor a an nth-order moment and cumulant transforms as*

$$\begin{aligned}
m_n^{(aX)} &= a^n m_n^{(X)}\,, && (2.30)\\
c_n^{(aX)} &= a^n c_n^{(X)}\,. && (2.31)
\end{aligned}$$

Property 2.2.2 (Sum) *Moments or cumulants of a sum are the sum of the moments or cumulants*

$$\begin{aligned} mom(X_1+Y,X_2,\ldots,X_n) &= mom(X_1,X_2,\ldots,X_n)+mom(Y,X_2,\ldots,X_n)\,, &(2.32)\\ cum(X_1+Y,X_2,\ldots,X_n) &= cum(X_1,X_2,\ldots,X_n)+cum(Y,X_2,\ldots,X_n)\,. &(2.33)\end{aligned}$$

Property 2.2.3 (Linear Transformation) *Let*

$$\boldsymbol{Y}=A\boldsymbol{X}\,, \tag{2.34}$$

be a linear transformation of a random variable $\boldsymbol{X}$ to a random variable $\boldsymbol{Y}$, where a single component Y_j depends on $\boldsymbol{X}$ like

$$Y_j=\sum_\alpha A_{j\alpha}X_\alpha\,. \tag{2.35}$$

The transformed moments up to third order are

$$\begin{aligned} mom(Y_j) &= \sum_\alpha A_{j\alpha}mom(X_\alpha)\,, &(2.36)\\ mom(Y_j,Y_k) &= \sum_{\alpha\beta} A_{j\alpha}A_{k\beta}mom\left(X_\alpha,X_\beta\right)\,, &(2.37)\\ mom(Y_j,Y_k,Y_l) &= \sum_{\alpha\beta\gamma} A_{j\alpha}A_{k\beta}A_{l\gamma}mom\left(X_\alpha,X_\beta,X_\gamma\right)\,. &(2.38)\end{aligned}$$

Cumulants transform like

$$\begin{aligned} cum(Y_j) &= \sum_\alpha A_{j\alpha}cum(X_\alpha)\,, &(2.39)\\ cum(Y_j,Y_k) &= \sum_{\alpha\beta} A_{j\alpha}A_{k\beta}cum\left(X_\alpha,X_\beta\right)\,, &(2.40)\\ cum(Y_j,Y_k,Y_l) &= \sum_{\alpha\beta\gamma} A_{j\alpha}A_{k\beta}A_{l\gamma}cum\left(X_\alpha,X_\beta,X_\gamma\right)\,. &(2.41)\end{aligned}$$

This property is called multilinearity and follows directly from the scaling property 2.2.1 and sum property 2.2.2.

Property 2.2.4 (Translation) *If we consider a translation of one component X_j of a vectorial random variable $\boldsymbol{X}$ by an arbitrary constant a_j such that*

$$Y_r = \delta_{rj}a_j+X_r\ \forall r\,, \tag{2.42}$$

the transformed moments are

$$\begin{aligned} mom(Y_1,Y_2,\ldots,Y_n) &= mom(X_1,X_2,\ldots,X_j+a_j,\ldots,X_n) &(2.43)\\ &= a_j mom\left(X_1,\ldots,X_{j-1},X_{j+1},\ldots,X_n\right)\\ &\quad +mom(X_1,\ldots,X_n)\,, &(2.44)\end{aligned}$$

and the cumulants are

$$\begin{aligned} cum(a_j+X_j) &= a_j+cum(X_j)\,, &(2.45)\\ cum(X_1,\ldots,X_j+a_j,\ldots,X_n) &= cum(X_1,\ldots,X_j,\ldots,X_n)\,. &(2.46)\end{aligned}$$

The translation only affects cumulants of first order. For this reason cumulants are sometimes called semi-invariant. In contrast, the translation of an nth-order moment involves moments with lower order than n. For a translation by a vector $\mathbf{a} = [a_1, \ldots, a_n]^T$ with $a_r \neq 0 \,\forall\, r \in \{1, \ldots, n\}$ moments of all orders up to n are needed to describe the transformation.

Property 2.2.5 (Symmetry) *Moments and cumulants are symmetric in their arguments, i.e.*

$$mom(X_1, X_2, \ldots, X_n) = mom(X_{P_1}, X_{P_2}, \ldots, X_{P_n}), \tag{2.47}$$

$$cum(X_1, X_2, \ldots, X_n) = cum(X_{P_1}, X_{P_2}, \ldots, X_{P_n}), \tag{2.48}$$

where $\mathbf{P}$ *is a permutation of* $[1, 2, \ldots, n]$ *and* P_i *is the* i*th element of* P*.*

Property 2.2.6 (Even Distribution) *If a random variable* X *has an even distribution* $p_X(x)$ *around the origin, all cumulants and moments of odd order vanish.*

Property 2.2.7 (Partitioning) *Given a number of random variables* $X_1, \ldots, X_n$ *that can be partitioned into two independent blocks, then all cross cumulants involving indices from both blocks are zero.*

$$cum(X_1, X_2, \ldots, X_n) = 0. \tag{2.49}$$

We give a short example. Consider four random variables X_1, X_2 and X_3, X_4, where the first two are independent of the second two. If all of them have zero mean, then their fourth order moments and cumulants can be written as

$$mom(X_1, X_2, X_3, X_4) = mom(X_1, X_2)\, mom(X_3, X_4), \tag{2.50}$$

$$\begin{aligned} cum(X_1, X_2, X_3, X_4) &= mom(X_1, X_2, X_3, X_4) - mom(X_1, X_2)\, mom(X_3, X_4) \\ &\quad - mom(X_1, X_3)\, mom(X_2, X_4) - mom(X_1, X_4)\, mom(X_2, X_3), \end{aligned} \tag{2.51}$$

$$\begin{aligned} &= mom(X_1, X_2)\, mom(X_3, X_4) - mom(X_1, X_2)\, mom(X_3, X_4) \\ &\quad - mom(X_1)\, mom(X_3)\, mom(X_2)\, mom(X_4) \\ &\quad - mom(X_1)\, mom(X_4)\, mom(X_2)\, mom(X_3) \\ &= 0. \end{aligned} \tag{2.52}$$

Here we use the fact that the joint probability function of two statistically independent variables factorizes into its marginals, e.g. $p_X(X_2, X_3) = p_X(X_2)\, p_X(X_3)$, which results in factorized cross-moments. Note, that even if X_1 and X_2 are independent

$$cum(X_1, X_2, X_1 X_2) \neq 0. \tag{2.53}$$

Property 2.2.7 is in general not shared by moments. As a consequence of this property all cross-cumulants of all orders of a vectorial random-variable $\boldsymbol{X}$ with statistically independent components X_j vanish. Thus, a tensor formed by cumulants of $\boldsymbol{X}$ of a given order $n > 1$ is diagonal. This property builds the basis of an independent component analysis algorithm developed in Chapter 6.

Property 2.2.8 (Sums of Independent Variables) *Consider two independent vector-valued random variables* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$*, where* $\boldsymbol{X}$ *has components* $X_1, \ldots, X_n$ *and* $\boldsymbol{Y}$ *has components* $Y_1, \ldots, Y_n$*. One of the most important properties of cumulants is that the cumulants of* $\boldsymbol{U} = \boldsymbol{X} + \boldsymbol{Y}$ *are just the sums of the corresponding cumulants of the individual variables,*

$$cum(U_1, \ldots, U_n) = cum(X_1, \ldots, X_n) + cum(Y_1, \ldots, Y_n). \tag{2.54}$$

This property is not shared by moments.

Property 2.2.9 (Non-Gaussianity) *Given a Gaussian random variable Y with the same mean and variance as a random variable X, then for $n \geq 3$ it holds that*

$$c_n^{(X)} = m_n^{(X)} - m_n^{(Y)} . \tag{2.55}$$

Higher order cumulants of Gaussian random variables are zero. Equation (2.55) together with Property 2.2.2 implies that higher order cumulants are insensitive to Gaussian noise. This can be used as a basis for noise reduction [Feng and Kammeyer, 1997].

2.2.5 Estimating Moments and Cumulants

Given P samples $x(t)$ $(1 \leq t \leq P)$ of a random variable X, the nth order moment can be estimated via the average $\overline{m}_n^{(X)}(t)$

$$\overline{m}_n^{(X)}(t) = \frac{1}{P} \sum_{t=1}^{P} x(t)^n . \tag{2.56}$$

For $P \to \infty$ the average $\overline{m}_n^{(X)}(t)$ converges to the moment $m_n^{(X)}$ with probability 1 [McCullagh, 1987]. Furthermore $\overline{m}_n^{(X)}(t)$ is unbiased since $\left\langle \overline{m}_n^{(X)}(t) \right\rangle = m_n^{(X)}$.

Sample cumulants can be estimated from the estimated moments via the relations (2.22 - 2.25). The third- and fourth-order sample cumulants of a mean free random variable X with unit variance are therefore given by

$$\overline{c}_3^{(X)} = \sum_{t=1}^{P} x(t)^3 , \tag{2.57}$$

$$\overline{c}_4^{(X)} = \sum_{t=1}^{P} x(t)^4 - 3 . \tag{2.58}$$

Cumulant estimates are in general biased. Compensation leads to the definition of κ-statistics [McCullagh, 1987]. Using the results of κ-statistics one can define the variance of a higher-order sample cumulant and thus the number of independent samples that is required to obtain a cumulant estimate with a given absolute precision. A detailed description goes beyond the scope of this thesis and is for example given in [McCullagh, 1987].

2.3 Gram-Charlier / Edgeworth Expansion

The relationship between cumulants and probability density function of a random variable x is often expressed via the Gram-Charlier or Edgeworth expansion of the p.d.f $p_X(x)$ around its best Gaussian approximation $\phi_X(x)$ with same mean and variance.

Definition 2.3.1 (Gram-Charlier Expansion) *The Gram-Charlier expansion of a probability density $p_X(x)$ around its best Gaussian approximation is given by*

$$\begin{aligned} \frac{p_X(x)}{\phi_X(x)} &= 1 + \frac{1}{3!} c_3^{(X)} h_3(x) + \frac{1}{4!} c_4^{(X)} h_4(x) + \frac{1}{5!} c_5^{(X)} h_5(x) \\ &\quad + \frac{1}{6!} \left(c_6^{(X)} + 10 c_3^{(X)} c_3^{(X)} \right) h_6(x) + \ldots , \end{aligned} \tag{2.59}$$

where $h_n(x)$ denotes the nth Hermite polynomial.

The standard Hermite polynomials ($c_X^{(1)} = 0$, $c_X^{(2)} = 1$) are

$$\begin{aligned} h_1(x) &= 2x, && (2.60)\\ h_2(x) &= 4x^2 - 2, && (2.61)\\ h_3(x) &= 8x^3 - 12x, && (2.62)\\ h_4(x) &= 16x^4 - 48x^2 + 12, && (2.63)\\ h_5(x) &= 32x^5 - 160x^3 + 120x, && (2.64)\\ h_6(x) &= 64x^6 - 480x^4 + 720x^2 - 120. && (2.65) \end{aligned}$$

The Hermite polynomials form a set of orthogonal polynomials. The orthogonality is defined by a scalar product with the Gaussian distribution as weighting function

$$\int \phi_X(x)\, h_i(x)\, h_j(x)\, \mathrm{d}x = \begin{cases} i!, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}. \quad (2.66)$$

Note, if X is a sum of r independent random variables normalized to unit variance with finite cumulants, then the central limit theorem [Wallace, 1958] states that the nth-order cumulant is of order $r^{\left(\frac{2-n}{2}\right)}$. Thus, the first four successive correction terms in the Gram-Charlier expansion (2.59) are of orders $r^{-1/2}$, r^{-1}, $r^{-3/2}$ and r^{-1} and these are not monotonously decreasing in r. Re-ordering the correction terms in (2.59) in decreasing order and collecting terms of same order in r leads to the Edgeworth expansion:

Definition 2.3.2 (Edgeworth Expansion) *The Edgeworth expansion of a probability density $p_{X_j}(x_j)$ around its best Gaussian approximation [McCullagh, 1987] up to order $\mathcal{O}\left(r^{-2}\right)$ is defined by*

$$\begin{aligned} \frac{p_X(x)}{\phi_X(x)} &= 1 \\ &\quad + \frac{1}{3!} c_3^{(X)} h_3(x) \\ &\quad + \left[\frac{1}{4!} c_4^{(X)} h_4(x) + 10 \frac{1}{6!} c_3^{(X)} c_3^{(X)} h_6(x) \right] \\ &\quad + \left[\frac{1}{5!} c_5^{(X)} h_5(x) + 35 \frac{1}{7!} c_3^{(X)} c_4^{(X)} h_7(x) + \frac{1}{9!} c_3^{(X)} c_3^{(X)} c_3^{(X)} h_9(x) \right] \\ &\quad + \mathcal{O}\left(r^{-2}\right). \end{aligned} \quad (2.67)$$

The correction terms are sorted in decreasing order in r. Terms that are of equal order in r are grouped together.

The infinite versions of these two expansions are identical. They only differ if they are truncated after a fixed number of terms. The Edgeworth expansion is often preferred for statistical calculations. In Chapter 3 it is used to derive a measure of statistical independence.

2.4 Cumulants and Probability Density Functions

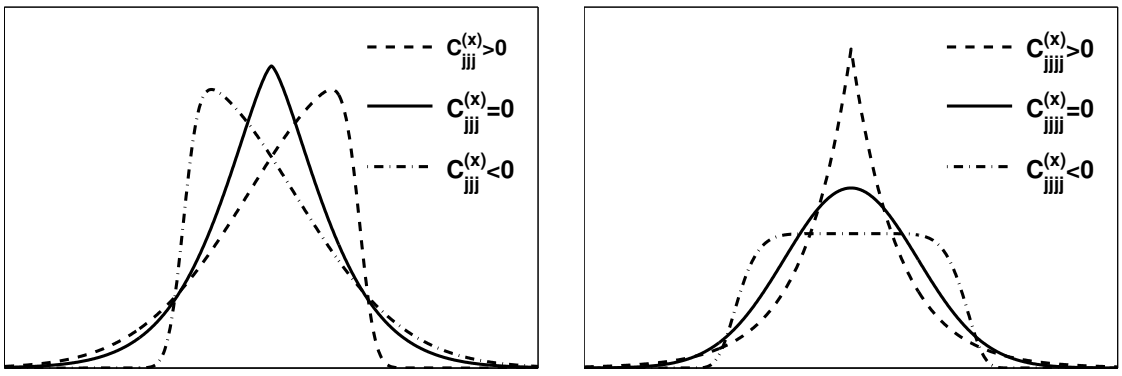
Since we will use cumulants in various situations throughout this thesis we will sometimes use a different, and simpler notation defined by

$$\begin{aligned} C_j^{(\mathbf{x})} &:= cum(X_j), && (2.68)\\ C_{jk}^{(\mathbf{x})} &:= cum(X_j, X_k), && (2.69)\\ \mathbf{C}_2^{(\mathbf{x})} &:= \mathbf{C}_2^{(\boldsymbol{X})}, && (2.70)\\ C_{jkl}^{(\mathbf{x})} &:= cum(X_j, X_k, X_l), && (2.71)\\ C_{jklm}^{(\mathbf{x})} &:= cum(X_j, X_k, X_l, X_m), && (2.72) \end{aligned}$$

where we drop the difference between a random variable and its actual realization.

We have seen in the previous section that cumulants are closely connected to the probability density function. The connection is revealed by the Edgeworth expansion. We also know from (2.16) that cumulants of a random variable X can be derived from an expansion of its second characteristic or cumulant generating function. The second characteristic function is defined as the Fourier transform of the probability density function $p_X(x)$ of x (see (2.5)). Therefore, cumulants up to order four are usually used to describe the main properties of a given probability density function $p_X(x)$ of X:

- Mean $C_j^{(\mathbf{x})}$: The auto-cumulant of first order denotes the mean of x_j.
- Variance $C_{jj}^{(\mathbf{x})}$: The auto-cumulant of second order defines the variance of x_j. The variance gives a quadratic measure of the distance between $p_X(x_j)$ evaluated at x_j and $x_j = C_j^{(\mathbf{x})}$.
- Skewness $C_{jjj}^{(\mathbf{x})}$: Due to Property 2.2.6 all cumulants of odd order of an asymmetric distribution vanish. Therefore $C_{jjj}^{(\mathbf{x})}$ is an indicator for asymmetric $p_X(x_j)$. If $C_{jjj}^{(\mathbf{x})} < 0$ ($C_{jjj}^{(\mathbf{x})} > 0$) then $p_X(x_j)$ is bend towards negative (positive) x_j (see Fig. 2.4 (a)).
- Kurtosis $C_{jjjj}^{(\mathbf{x})}$: The kurtosis $C_{jjjj}^{(\mathbf{x})}$ is usually used as a measure of the non-Gaussianity of the probability density $p_X(x_j)$. If the even part of a distribution has heavier tails than a Gaussian distribution and a peak at $C_j^{(\mathbf{x})}$ then $C_{jjjj}^{(\mathbf{x})}$ is positive. Such distributions are called super-Gaussian distributions. For flatter distributions where the even part has lighter tails $C_{jjjj}^{(\mathbf{x})}$ is negative. Such distributions are called sub-Gaussian distributions (see Fig. 2.4 (b)).



(a) (b)

Figure 2.1: Possible values for **(a)** skewness and **(b)** kurtosis. All distributions shown in **(a)** resp. **(b)** have zero mean and same variance.

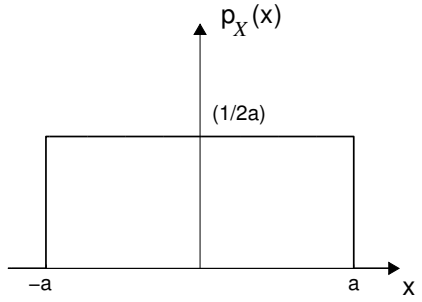
2.4.1 Examples of Probability Density Functions and their Higher-Order Cumulants

Table 2.4.1 shows examples of probability functions (uniform distribution, Gaussian distribution, and exponential distribution) and their cumulants up to fourth order.

2.5 Summary

We have seen that moments and cumulants are two different descriptions of the same reality. Nevertheless it is preferable to work with cumulants rather than with moments because:

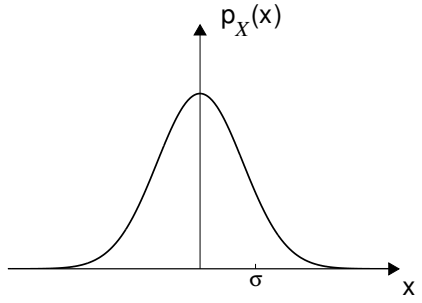
Uniform distribution



$$p_X(x) = \frac{1}{2a} \qquad (x \in [-a, a])$$

n	$m_n^{(X)}$	$c_n^{(X)}$
1	0	0
2	$a^2/3$	$a^2/3$
3	0	0
4	$a^4/5$	$-2a^4/15$

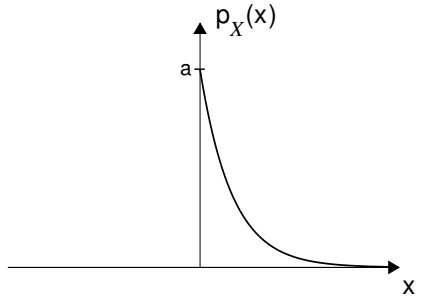
Gaussian distribution



$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

n	$m_n^{(X)}$	$c_n^{(X)}$
1	0	0
2	σ^2	σ^2
3	0	0
4	$3\sigma^4$	0

Exponential distribution



$$p_X(x) = a\exp(-ax) \qquad (x \geq 0)$$

n	$m_n^{(X)}$	$c_n^{(X)}$
1	$1/a$	$1/a$
2	$2/a^2$	$1/a^2$
3	$6/a^3$	$2/a^3$
4	$24/a^4$	$6/a^6$

Table 2.1: Example distributions and their moments and cumulants up to fourth order. From top to bottom: uniform distribution, Gaussian distribution, and exponential distribution.

a. For independent random variables the cumulants of a sum are the sum of the cumulants.

b. Cross-cumulants of statistically independent random variables are zero. Therefore cumulant tensor of independent random variables are diagonal. This is the most important property of cumulants since it forms the basis of a wide range of algorithms performing independent component analysis. One of them is introduced in Chapter 4.

c. Higher-order cumulants of a random variable with Gaussian distribution are zero.

d. Higher-order cumulants are insensitive to Gaussian noise (a consequence of a and b).

e. Cumulants change in a multilinear way under arbitrary affine transformations since second- and higher order cumulants are insensitive to the mean of a random variable.

3

Measures of Independence and their Approximations

As the term *independent component analysis* suggests, one needs a measure of statistical dependence in order to formulate an adequate algorithm. Here we introduce mutual information as a measure of the dependence between random variables. It is always non-negative, and vanishes if and only if the variables are statistically independent. In practical ICA algorithms mutual information builds the basis of an objective function, subject to optimization, such that the mutual information is minimized. In this way the original source signal can be estimated. However, mutual information is a function of the probability density function of the estimated source signal (3.11), which is usually unknown. Using the Edgeworth expansion of the probability density functions (2.3) we can approximate the mutual information as a function of higher-order cumulants. This allows the formulation of a practical measure of statistical dependence (3.27).

As a side product a relation between mutual information and negentropy (3.17) can be established leading directly to a connection between minimization of mutual information and maximization of kurtosis.

3.1 Entropy

Shannon [Shannon, 1948] developed the concept of entropy to measure the uncertainty of a discrete random variable.

Definition 3.1.1 (Entropy, Joint Entropy) *The differential entropy $H(X)$ of a random variable X with probability density $p_X(x)$ is defined by [Shannon, 1948]*

$$H(X) = -\int p_X(x) \log p_X(x)\, dx\,. \tag{3.1}$$

In the multidimensional case the differential entropy of a vectorial random variable $\boldsymbol{X} = [X_1, \ldots, X_N]^T$ with joint probability density $p_{\boldsymbol{X}}(\mathbf{x})$ is defined as

$$H(\boldsymbol{X}) = -\int p_{\boldsymbol{X}}(\mathbf{x}) \log p_{\boldsymbol{X}}(\mathbf{x})\, d\mathbf{x}\,. \tag{3.2}$$

The differential entropy can be interpreted as the degree of information the observation of a random variable gives. For a fixed variance it is maximal if $p_{\boldsymbol{X}}(\mathbf{x})$ is a Gaussian probability density [Cover and Thomas, 1991]. For any other distribution, the differential entropy is strictly smaller. The transformation property of entropy can be expressed by

Property 3.1.1 (Entropy Under Linear Transforamtion) *If the random variable $\boldsymbol{X}$ with entropy $H(\boldsymbol{X})$ is transformed by an invertible linear transformation $\boldsymbol{U} = \mathbf{R}\boldsymbol{X}$ the entropy transforms according to*

$$H(\boldsymbol{U}) = H(\boldsymbol{X}) - \log|\det\mathbf{R}|, \tag{3.3}$$

which can be directly verified using (3.2) and the relation $p_{\boldsymbol{U}}(\mathbf{u}) = p_{\boldsymbol{X}}(\mathbf{x})/|\det\mathbf{R}|$.

Definition 3.1.2 (Marginal Entropy) *The marginal entropy $H(X_1)$ of a scalar random variable X_1 out of $\boldsymbol{X} = [X_1, X_2]$ is defined by*

$$H(X_1) = -\int p_{\boldsymbol{X}}(\mathbf{x}) \log p_{\boldsymbol{X}}(x_1)\,\mathrm{d}\mathbf{x} = -\int p_{\boldsymbol{X}}(x_1) \log p_{\boldsymbol{X}}(x_1)\,\mathrm{d}x_1,, \tag{3.4}$$

where $p_{\boldsymbol{X}}(x_1)$ is the marginal probability density of X_1.

3.2 Kullback Leibler Divergence and Mutual Information

While the entropy of a variable is a measure of the uncertainty in its distribution, the relative entropy or Kullback Leibler divergence is a measure of the statistical distance between two distributions.

Definition 3.2.1 (Kullback Leibler Divergence) *The Kullback Leibler (KL) divergence $KL(p_X(x), p_X(x))$ or relative entropy between two probability density functions $p_X(x)$ and $q_X(x)$ is defined as*

$$KL(p_X(x), q_X(x)) = \int p_X(x) \log \frac{p_X(x)}{q_X(x)}\mathrm{d}x\,. \tag{3.5}$$

The KL divergence is not a real distance measure because it is not symmetric ($KL(p_X(x), q_X(x)) \neq KL(q_X(x), p_X(x))$). In statistics the KL divergence appears as the expected value of the log likelihood ratio. The KL divergence is always nonnegative, and zero if and only if the two distributions are equal. This fact can be shown using Jensen's inequality [Rudin, 1987]

$$\langle f(y)\rangle \geq f(\langle y\rangle)\,. \tag{3.6}$$

Setting $f(y) = -\log(y)$ and $y = q_X(x)/p_X(x)$ in Equation (3.6) we can derive

$$\begin{aligned}
\langle f(y)\rangle &= -\int p_X(x) \log \frac{q_X(x)}{p_X(x)}\mathrm{d}x && (3.7)\\
&= KL(p_X(x), q_X(x)) && (3.8)\\
&\geq \log\left(\int p_X(x)\,\frac{q_X(x)}{p_X(x)}\mathrm{d}x\right) && (3.9)\\
&= \log\left(\int q_X(x)\,\mathrm{d}x\right) = \log 1 = 0\,, && (3.10)
\end{aligned}$$

where we used the inequality (3.6) from (3.8) to (3.9). Since the logarithm is a strictly concave function, the equality holds if and only if $p_X(x)/q_X(x) = \text{const.}$, i.e. $p_X(x) = q_X(x)$.

In the case of independent component analysis we can use the fact that the probability density of a random variable $\boldsymbol{U}$ with independent components U_j factorizes, i.e. $p_{\boldsymbol{U}}(\mathbf{u}) = \prod_j p_{U_j}(u_j)$. The KL divergence $KL\left(p_{\boldsymbol{U}}(\mathbf{u}), \prod_j p_{U_j}(u_j)\right)$ therefore vanishes and is thus a good measure for statistical independence.

Definition 3.2.2 (Mutual Information) *The mutual information $I(X_j, X_k)$ between two random variables X_j and X_k is the KL divergence between their joint distribution and the product of their marginals*

$$I(X_j, X_k) = \int p_{\boldsymbol{X}}(\mathbf{x}) \log \frac{p_{\boldsymbol{X}}(\mathbf{x})}{p_{\boldsymbol{X}}(x_j)\,p_{\boldsymbol{X}}(x_k)} d\mathbf{x}\,, \tag{3.11}$$

with $\boldsymbol{X} = [X_j, X_k]^T$.

The mutual information is symmetric, i.e. $I(X_j, X_k) = I(X_k, X_j)$. Since the mutual information is derived directly from the KL divergence it is always positive and it is equal zero if and only if X_j and X_k are statistically independent [Cover and Thomas, 1991]. The extension to the multidimensional case is straightforward and called multi information.

Definition 3.2.3 (Multi Information) *Given a random vector $\boldsymbol{X} = [X_1, \ldots, X_N]$ the multi information is given by the KL divergence between the joint distribution $p_{\boldsymbol{X}}(\mathbf{x})$ and the product of the marginal distributions $p_{\boldsymbol{X}}(x_j)$*

$$I(\boldsymbol{X}) = \int p_{\boldsymbol{X}}(\mathbf{x}) \log \frac{p_{\boldsymbol{X}}(\mathbf{x})}{\prod_j p_X(x_j)} d\mathbf{x}. \tag{3.12}$$

Rewriting Equation 3.12 using the definition of entropy 3.1 we can derive

$$I(\boldsymbol{X}) = \sum_j H(X_j) - H(\boldsymbol{X}), \tag{3.13}$$

where $H(X_j)$ are the entropies of the marginals X_j (3.4).

Note, that by Property 3.1.1, we have for an invertible linear transformation $\boldsymbol{U} = \mathbf{R}\boldsymbol{X}$

$$I(U_1, U_2, \ldots, U_n) = \sum_i H(U_i) - H(\mathbf{R}\boldsymbol{X}) = \sum_i H(U_i) - H(\boldsymbol{X}) - \log|\det \mathbf{R}|. \tag{3.14}$$

This property is for example used in the ICA algorithms introduced by [Bell and Sejnowski, 1995] and [Lee et al., 1999], where $\mathbf{R}$ is learned, such that it minimizes (3.14). This results in statistically independent U_i.

3.3 Negentropy and Mutual Information

Definition 3.3.1 (Negentropy) *The negentropy J of a vectorial random variable $\boldsymbol{X}$ with probability density $p_{\boldsymbol{X}}(\mathbf{x})$ is defined as*

$$J(\boldsymbol{X}) = H(\boldsymbol{X}^{gauss}) - H(\boldsymbol{X}), \tag{3.15}$$

where $\boldsymbol{X}^{gauss}$ is a Gaussian random variable with same covariance matrix as $\boldsymbol{X}$.

Since the entropy of a Gaussian variable is maximal, the negentropy is always nonnegative and zero if $\boldsymbol{X}$ has Gaussian components, too. Thus, it is a measure of non-Gaussianity of the probability density $p_{\boldsymbol{X}}(\mathbf{x})$. Additionally, it has the important property of being invariant under linear invertible transformations which can be seen by using Property 3.1.1 in (3.15).

The relation between negentropy and mutual information can be established by noting that

$$\begin{aligned} J(\boldsymbol{X}) - \sum_j J(X_j) &= H(\boldsymbol{X}^{gauss}) - H(\boldsymbol{X}) - \sum_j H\left(X_j^{gauss}\right) + \sum_j H(X_j) \\ &= I(\boldsymbol{X}) + H(\boldsymbol{X}^{gauss}) - \sum_j H\left(X_j^{gauss}\right), \end{aligned} \tag{3.16}$$

where we have used (3.13) and $H(X_j)$ are the marginal entropies (3.4). The marginal negentropies $J(X_j)$ are defined correspondingly. If $\boldsymbol{X}$ has uncorrelated components then $H(\boldsymbol{X}^{gauss}) = \sum_j H\left(X_j^{gauss}\right)$ and (3.16) simplifies to

$$I(\boldsymbol{X}) = J(\boldsymbol{X}) - \sum_j J(X_j). \tag{3.17}$$

Taking into account that the first term on the right hand side in Equation (3.17) is constant under invertible linear transformations [Comon, 1994b], minimization of $I(\boldsymbol{X})$ is equivalent to maximizing the marginal negentropies of the components X_j. This is equivalent to maximizing the non-Gaussianity of the marginal probabilities $p_{\boldsymbol{X}}(x_j)$. Thus, minimizing the mutual information is equality to the maximization of the non-Gaussianity of the estimated signal components (see e.g. (6.3)).

3.4 Approximation of the Negentropy

All measures presented up to now use statistical information of all orders since they all use the exact probability densities. Normally we do not know these distributions and thus such measures are not useful for real applications. However, we can use the approximations to a probability density introduced in Section 2.3 to define an approximate measure of independence that is applicable. This measure can then be used to form the basis of a possible ICA algorithm. We will show an example at the end of this section.

In Equation (3.15) the negentropy is defined as a function of entropies. Inserting the definition of entropy (3.1) into (3.15) we derive

$$J(X_j) = -\int \phi_{X_j}(x_j) \log \phi_{X_j}(x_j)\,\mathrm{d}x_j + \int p_{X_j}(x_j) \log p_{X_j}(x_j)\,\mathrm{d}x_j\,. \tag{3.18}$$

The probability densities in (3.18) can be expanded using the Edgeworth expansion (2.67). We consider the first three terms of the Edgeworth expansion of $p_{X_j}(x_j)$. These are given by

$$\begin{aligned} p_{X_j}(x_j) &\approx \phi_{X_j}(x_j)\left(1 + \frac{1}{3!}c_3^{(x_j)} h_3(x_j) + \frac{1}{4!}c_4^{(x_j)} h_4(x_j)\right) \\ &=: \phi_{X_j}(x_j)\left(1 + \nu(x_j)\right), \end{aligned} \tag{3.19}$$

where we introduced the definition in the second line for simplification. Inserting (3.19) into (3.18) we can derive

$$\begin{aligned} J(X_j) &\approx -\int \phi_{X_j}(x_j) \log \phi_{X_j}(x_j)\,\mathrm{d}x_j + \int \phi_{X_j}(x_j)(1+\nu) \log\left(\phi_{X_j}(x_j)(1+\nu)\right)\mathrm{d}x_j \\ &= -\int \phi_{X_j}(x_j) \log \phi_{X_j}(x_j)\,\mathrm{d}x_j + \int \phi_{X_j}(x_j)(1+\nu) \log \phi_{X_j}(x_j)\,\mathrm{d}x_j \\ &\quad + \int \phi_{X_j}(x_j)(1+\nu)\log(1+\nu)\,\mathrm{d}x_j \qquad (3.20) \\ &= \int \phi_{X_j}(x_j)\,\nu \log \phi_{X_j}(x_j)\,\mathrm{d}x_j + \int \phi_{X_j}(x_j)(1+\nu)\log(1+\nu)\,\mathrm{d}x_j\,, \qquad (3.21) \end{aligned}$$

where from (3.20) to (3.21) we summed up the first and the third term on the right hand side. The first term on the right hand side of Equation (3.21) vanishes which can be proved by simply inserting the definitions of Hermite polynomials (2.65) and using the fact that $\log \phi_{X_j}(x_j)$ is a polynomial of second order.

To simplify the second term on the right hand side of (3.21) we make a further approximation. Since the expansion of $p_{X_j}(x_j)$ is taken in the vicinity of its best Gaussian approximation $\phi_{X_j}(x_j)$ and we assume the correction terms to be small we can approximate

$$\log(1+\nu) \approx \nu - \frac{\nu^2}{2}\,. \tag{3.22}$$

Using approximation (3.22) we can rewrite (3.21) as

$$J(X_j) \approx \int \phi_{X_j}(x_j)(1+\nu)\left(\nu - \frac{\nu^2}{2}\right)\mathrm{d}x_j\,. \tag{3.23}$$

After reinserting the expressions from (3.19) for ν this integral can be calculated explicitly. For further simplification we consider higher order terms to be negligible and take only terms into account with cumulants up to order $O\left(r^{-2}\right)$ (see central limit theorem in Section (2.3)). Furthermore we use the orthogonality property 2.66 plus two additional properties of Hermite polynomials derived by [Comon, 1994b]

$$\int \phi_{x_j}(x_j)\, h_3^2(x_j)\, h_4(x_j)\,\mathrm{d}x_j = (3!)^3\,, \tag{3.24}$$

$$\int \phi_{x_j}(x_j)\, h_3^3(x_j)\,\mathrm{d}x_j = 0\,. \tag{3.25}$$

Inserting (3.19) into (3.23) and using the Properties 3.24 and 3.25 the negentropy can be expressed as

$$J(X_j) = \frac{1}{12}\left(c_3^{(X_j)}\right)^2 + \frac{1}{48}\left(c_4^{(X_j)}\right)^2 - \frac{1}{8}\left(c_3^{(X_j)}\right)^2 c_4^{(X_j)} + O\left(i^{-2}\right), \tag{3.26}$$

where we only considered terms up to $O\left(i^{-2}\right)$. Finally we use relation (3.17) and arrive at an approximated formula for mutual information

$$I(\boldsymbol{X}) \approx J(\boldsymbol{X}) - \frac{1}{12}\sum_j \left(c_3^{(X_j)}\right)^2 - \frac{1}{48}\sum_j \left(c_4^{(X_j)}\right)^2 + \frac{1}{8}\sum_j \left(c_3^{(X_j)}\right)^2 c_4^{(X_j)}. \tag{3.27}$$

Since in the area of standard ICA we are mostly interested in finding linear invertible transformations $J(\boldsymbol{X})$ is a constant in such cases and can be neglected. We therefore do not compute an approximated version of $J(\boldsymbol{X})$.

We will conclude with a simple example: Consider two random variables X_1 and X_2 and their realizations x_1 and x_2. Assume we want to minimize the mutual information between x_1 and x_2 via a linear transformation $\mathbf{Q}$ such that $\mathbf{y} = \mathbf{Q}\mathbf{x}$, where $\mathbf{x} = [x_1, x_2]^T$. The mutual information can be written as

$$I(Y_1, Y_2) = J(\boldsymbol{Y}) - \frac{1}{12}\sum_{j=1}^{2} \left(c_3^{(Y_j)}\right)^2 - \frac{1}{48}\sum_{j=1}^{2} \left(c_4^{(Y_j)}\right)^2 + \frac{1}{8}\sum_{j=1}^{2} \left(c_3^{(Y_j)}\right)^2 c_4^{(Y_j)}. \tag{3.28}$$

The first term on the right hand side of Equation (3.28) is a constant and can be neglected in the minimization procedure. Furthermore instead of minimizing $I(Y_1, Y_2)$ we can maximize $-I(Y_1, Y_2)$. Thus we arrive at an objective, subject to maximization

$$\begin{aligned} \Psi &= \frac{1}{12}\sum_{j=1}^{2} \left(c_3^{(Y_j)}\right)^2 + \frac{1}{48}\sum_{j=1}^{2} \left(c_4^{(Y_j)}\right)^2 - \frac{1}{8}\sum_{j=1}^{2} \left(c_3^{(Y_j)}\right)^2 c_4^{(Y_j)} \\ &= \frac{1}{12}\sum_{j=1}^{2} \left(C_{jjj}^{(\mathbf{y})}\right)^2 + \frac{1}{48}\sum_{j=1}^{2} \left(C_{jjjj}^{(\mathbf{y})}\right)^2 - \frac{1}{8}\sum_{j=1}^{2} \left(C_{jjj}^{(\mathbf{y})}\right)^2 C_{jjjj}^{(\mathbf{y})}. \end{aligned} \tag{3.29}$$

This is essentially the objective function used in the ICA algorithm introduced in Chapter 6.

3.5 Conclusion

We have derived an approximated version of the mutual information $I(\boldsymbol{X})$ that allows to define a simple measure of statistical independence between them. Usually, it is used to measure the mutual independence between two scalar random variables X_j and X_k. The expression for $I(X_j, X_k)$ is given by Equation (3.28). The approximation (3.19) of course only holds when $p_{X_j}(x_j)$ is not far from the Gaussian distribution $\phi_{X_j}(x_j)$. Therefore other approximations for mutual information that are not based on higher-order cumulants, but use more general measures of non-Gaussianity have been developed [Hyvärinen, 1997].

4

Blind Source Separation and Independent Component Analysis

In this chapter we like to introduce the basic concepts of independent component analysis (ICA) and blind source separation (BSS). We start with a short introduction to principal component analysis in the next section, since it is the most common method using second-order statistics and builds the core of the first step of the ICA and BSS algorithms introduced in this thesis. Thereafter we give a brief introduction and historical overview of BSS and ICA in Section 4.3. ICA algorithms often use a two-stage approach where the second stage consists of optimizing a contrast function. This method is explained in detail in Section 4.3. A different ICA approach, based on second-order statistics, will be described in Section 4.4. A nonlinear extension to the standard linear BSS problem is introduced in Section 4.5. In Section 4.6 we address the problem of matrix-diagonalization, an important tool used in several ICA/BSS algorithm. We conclude with the definition of a performance measure in Section 4.7. Such a measure allows to quantify performances of ICA/BSS algorithms based on the knowledge of the underlying mixing-matrix.

4.1 Principal Component Analysis

Principal component analysis (PCA), sometimes called Karhunen-Loève transform, is the most common method used in signal processing, statistics, and neural computing. It is a second-order statistical method based on the work by Pearson [1901]. PCA finds a linear representation $\mathbf{y}$ of an observed signal $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ such that the components y_i are uncorrelated and have variances that are extremes (maxima and minima) along the new coordinate axes. The name PCA comes from the principal axes of an ellipsoid which are just the coordinate axes in question.

Given the covariance matrix $\mathbf{C}_2^{(\mathbf{x})}$ of $\mathbf{x}$ as defined in 2.70 we can calculate an orthogonal basis by finding its n eigenvalues λ_i and corresponding eigenvectors $\mathbf{v}_i$. They are solutions of the eigenvalue problem

$$\mathbf{C}_2^{(\mathbf{x})}\mathbf{v}_i = \lambda_i \mathbf{v}_i \,, \tag{4.1}$$

where we assume that the λ_i are distinct. Ordering the eigenvectors by descending eigenvalues (starting with the largest), we can build an orthogonal basis with the first eigenvector having the direction of largest variance of the data and each succeeding eigenvector accounts for as much of the remaining variance, as possible. Since we will use it in the next section we also define the eigenvalue problem in matrix formulation

$$\mathbf{C}_2^{(\mathbf{x})}\mathbf{V} = \mathbf{V}\mathbf{D} \,, \tag{4.2}$$

where $\mathbf{V}$ is an $N \times N$ matrix with the eigenvectors of the covariance matrix as its column vectors $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$. $\mathbf{D}$ is a diagonal matrix with the ordered eigenvalues on its main diagonal.

Now, consider a transformation

$$\mathbf{y} := \mathbf{V}^T \mathbf{x}, \tag{4.3}$$

where we assume, that $\mathbf{x}$ has zero mean. The components of $\mathbf{y}$ are the coordinates in the orthogonal base, and thus, the components y_i are uncorrelated. We can reconstruct the original data $\mathbf{x}$ from $\mathbf{y}$ by

$$\mathbf{x} = \mathbf{V}\mathbf{y}, \tag{4.4}$$

since for orthogonal matrices $\mathbf{V}^T = \mathbf{V}^{-1}$.

Instead of using all n eigenvectors of the covariance matrix, we may think of a matrix $\mathbf{V}_l$ with only the first l eigenvectors as its column vectors. A similar transformation can be created

$$\mathbf{y} = \mathbf{V}_l^T \mathbf{x}. \tag{4.5}$$

The inverse transformation yields

$$\tilde{\mathbf{x}} = \mathbf{V}_l \mathbf{y}. \tag{4.6}$$

Thus, we project the original N-dimensional data vector $\mathbf{x}$ on an l-dimensional coordinate system and map the vector back by a transformation, which is a linear combination of the orthogonal basis vectors $\mathbf{v}_i$, to obtain $\tilde{\mathbf{x}}$. PCA minimizes the mean-square error between the original data $\mathbf{x}$ and the representation $\tilde{\mathbf{x}}$ with less eigenvectors.

This operation provides a way to compress data without loosing much information. By taking the eigenvectors with largest variance as little information as possible, in the mean-square sense, is lost [Jolliffe, 1986].

It is important to notice that, since $\mathbf{y}$ has zero mean and the covariance matrix of $\mathbf{y}$ is diagonal, the components of $\mathbf{y}$ are already statistically independent up to second order (cf. Prop. 2.2.7).

Thus, PCA consists of computing the eigenvectors and eigenvalues of the covariance matrix of the observed signal and a transformation given by Equation (4.3).

4.2 Linear Blind Source Separation and Independent Component Analysis

In signal processing one often has to deal with high dimensional data such as a vectorial signal $\mathbf{x}(t) = [x_1(t)(t), \ldots, x_M(t)(t)]^T$. To facilitate the interpretation of such a signal a useful representation of the data in terms of a linear or nonlinear transformation has to be found. Depending on the purpose there exist many different, mostly linear, transformations e.g. Fourier transformation, principal component analysis, and factor analysis. Relatively new are two additional methods: Blind source separation and independent component analysis.

The vectorial signal $\mathbf{x}(t)$ to be analyzed, which we will refer to as input signal, is often a mixture of some underlying signal components $s_i(t)$ coming from different sources. For instance, the sound we hear is usually a superposition of several sound sources, such as a person speaking and a phone ringing. In a simple model of this data generation process, it is assumed that there are as many sources as input signal components, that at most one $s_i(t)$ is normally distributed, and that the mixing is linear and noise free, yielding the relation

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \tag{4.7}$$

with an invertible $N \times N$ mixing matrix $\mathbf{A}$ and source signal $\mathbf{s}(t) = [s_1(t)(t), \ldots, s_N(t)(t)]^T$. In the following we will drop the reference to time, and simply assume some sets of source and input data related by (4.7). We also assume for simplicity that the source signal and input signal have zero mean.

The goal of blind source separation (BSS) is to recover the unknown source signal $\mathbf{s}(t)$ from the observable $\mathbf{x}(t)$ without any prior information, this is the reason why it is called blind. The only assumption is that the source signal components (person and phone) are statistically independent. Given only the observed signal $\mathbf{x}(t)$ we want to find a matrix $\mathbf{R}$ such that the components of

$$\mathbf{u}(t) \quad = \quad \mathbf{R}\mathbf{x}(t), \tag{4.8}$$

are mutually statistically independent.

The method of finding a representation of the observed data such that the components are mutually statistically independent is called independent component analysis (ICA). It has been proven that ICA solves the linear BSS problem, apart from the fact that the source signal components can only be recovered up to a multiplication of the source signal components $x_i(t)$ by constants (or multiplication of $\mathbf{x}(t)$ by a diagonal matrix $\mathbf{\Lambda}$) and permutation (multiplication with a permutation matrix$\mathbf{P}$) [Comon, 1994b]. Thus the relation between $\mathbf{R}$ and mixing matrix is given by $\mathbf{R} = \mathbf{\Lambda P A}$.

The seminal work on independent component analysis was by Herault and Jutten [1986]. They introduced a simple neural network with feedback that was able to separate unknown independent source signal components and were the first to introduce the term *independent components analysis* (see [Comon, 1994b]). They further developed the orginial model [Jutten and Herault, 1991] and gave a first mathemtaical interpretation [Comon et al., 1991]. Comon [1994b] was the first to give the concept its statistical framework and introduced the concept of contrast functions.

Parallel to these studies on independent component analysis, mostly coming from France, Bell and Sejnowski [1995] put the the blind source separation problem into an information-theoretic framework by proposing an algorithm based on the Infomax principle introduced by Linsker [1989]. The objective of [Bell and Sejnowski, 1995] was to maximize the mutual information between the inputs and outputs of a neural network. This work has been subject of many scientific publications and led to further improvements, for example the concept of natural-gradient introduced by Amari [1998] which significantly improved the convergence of the Infomax learning rule.

Since then a large amount of algorithms has been proposed with extended Infomax [Lee et al., 1999], FastICA [Hyvärinen, 1999], JADE [Cardoso and Souloumiac, 1993], SOBI [Belouchrani et al., 1997], and TDSEP [Ziehe and Müller, 1998] being the most popular. Here, we do not give a generall overview over ICA, rather introduce the basic ICA-knowledge needed in the subsequent chapters. For a good introduction to ICA refer to the text-books [Lee, 1998], [Hyvärinen et al., 2001b] and [Cichocki and Amari, 2002].

It is important to make a clear distinction between ICA and BSS: ICA finds a representation $\mathbf{u}(t)$ of an observed signal $\mathbf{x}(t)$ such that the signal components $u_i(t)$ are mutually statistically independent. BSS finds a representation $\mathbf{u}(t)$ such that the signal components $u_i(t)$ coincide with the original source signal components $\mathbf{s}(t)\,i$, underlying the observed signal. This can be done up to permutation and scaling. In the linear case ICA and BSS are analogous methods. But, considering the more general BSS/ICA problem, where the observed signal $\mathbf{x}(t)$ is the result of a nonlinear transformation of the source signal $\mathbf{s}(t)$, the two methods (ICA and BSS) are no longer equivalent. This is because the independence assumption is not enough to regain the original source signal out of a nonlinear mixture. We will discuss this issue in Section 4.5. Thus, the notions BSS and ICA denote two distinct methods in the nonlinear case, although the extracted source signal components have to be mutually independent in the linear as well as in the nonlinear case.

4.3 Linear Independent Component Analysis

The goal of ICA is to find a representation of an observed signal such that the components of the transformed signal are mutually statistically independent. It can be understood as an extension to PCA. ICA not only decorrelates the observed signal, which requires second-order statistics, but it also minimizes higher-order statistical dependencies, forcing all signal components to be as independent as possible.

4.3.1 A two-stage Approach

The classical approach to ICA proceeds in two steps: Sphering/whitening of the input signal $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ with a whitening matrix $\mathbf{W}$ is followed by an orthogonal transformation (rotation) $\mathbf{u}(t) = \mathbf{Q}\mathbf{y}(t)$. Thus the unmixing matrix $\mathbf{R}$ is $\mathbf{R} = \mathbf{QW}$ and the ICA model (4.8) can be written as

$$\mathbf{u}(t) = \mathbf{R}\mathbf{x}(t) = \mathbf{QW}\mathbf{x}(t) = \mathbf{Q}\mathbf{y}(t)\,, \tag{4.9}$$

where $\mathbf{y}(t)$ denotes the whitened signal. Besides this classical approach there exist algorithms that estimate the (non-orthogonal) unmixing transformation in a single step without a prewhitening (see e.g. [Akuzawa, 2001; De Lathauwer et al., 1995]). But, since the ICA algorithms introduced in this thesis follow the classical approach we will focus on it and give a short introduction in the following.

Whitening

A common linear preprocessing step in many ICA algorithms as well as in linear SFA is the whitening of the input signal $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$, where we assume from now on, without loss of generality, that $\mathbf{x}(t)$ has zero mean components. Whitening is a linear transformation $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ resulting in a signal with mutually uncorrelated components $y_k(t)$ with unit variance. Thus they fulfill the constraints

$$\langle y_k(t) \rangle = 0 \qquad \text{(zero mean)}, \tag{4.10}$$

$$\left\langle (y_k(t))^2 \right\rangle = 1 \qquad \text{(unit variance)}, \tag{4.11}$$

$$\langle y_k(t)\, y_l(t) \rangle = 0 \qquad \text{(decorrelation)}. \tag{4.12}$$

Therefore, the components $y_k(t)$ of the whitened signal $\mathbf{y}(t)$ have

- vanishing first-order cumulants

$$C_k^{(\mathbf{y})} = \langle y_k(t) \rangle = 0 \quad \forall k \in \{1, \ldots, N\} \tag{4.13}$$

- vanishing second-order cross-cumulants

$$C_{kl}^{(\mathbf{y})} = \langle [y_k(t) - \langle y_k(t) \rangle]\, [y_l(t) - \langle y_l(t) \rangle] \rangle = 0 \quad \forall k, l \in \{1, \ldots, N\} \wedge k \neq l \tag{4.14}$$

Thus, $\mathbf{y}(t)$ has statistically independent components up to second order (cf. Prop. 2.2.7).

Because whitening is decorrelation plus normalization it is in fact similar to PCA. Using the matrix of eigenvectors $\mathbf{V}$ and the diagonal matrix $\mathbf{D}$ with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ on the main diagonal as computed by PCA (cf. (4.2)), the whitening matrix is given by

$$\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{V}^T. \tag{4.15}$$

It can be easily shown that $\mathbf{W}$ defines indeed a whitening transformation. Using (4.15) and $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ we can derive

$$\left\langle \mathbf{y}(t)\, \mathbf{y}(t)^T \right\rangle = \mathbf{W} \left\langle \mathbf{x}(t)\, \mathbf{x}(t)^T \right\rangle \mathbf{W}^T \tag{4.16}$$

$$= \mathbf{D}^{-1/2}\mathbf{V}^T \left\langle \mathbf{x}(t)\, \mathbf{x}(t)^T \right\rangle \mathbf{V}\mathbf{D}^{-1/2} \tag{4.17}$$

$$= \mathbf{D}^{-1/2}\mathbf{V}^T \mathbf{C}_2^{(\mathbf{x})} \mathbf{V}\mathbf{D}^{-1/2} \tag{4.18}$$

$$\left(\mathbf{C}_2^{(\mathbf{x})} = \left\langle \mathbf{x}(t)\, \mathbf{x}(t)^T \right\rangle \text{ (cf. (2.19))} \right)$$

$$= \mathbf{D}^{-1/2}\mathbf{V}^T \mathbf{V}\mathbf{D}\mathbf{V}^T \mathbf{V}\mathbf{D}^{-1/2} = \mathbf{I} \tag{4.19}$$

$$\left(\mathbf{C}_2^{(\mathbf{x})} = \mathbf{V}\mathbf{D}\mathbf{V}^T \text{ (cf. (4.2))} \right),$$

where $\mathbf{I}$ denotes the $N \times N$ identity matrix. Thus the covariance matrix of $\mathbf{y}(t)$ is the identity matrix, and therefore $\mathbf{y}(t)$ is white.

This exact diagonalization of the covariance matrix of $\mathbf{y}(t)$ is preserved under any orthogonal transformations $\mathbf{Q}$, i.e. a pure rotation possibly plus reflections, because

$$\left\langle \mathbf{u}(t)\, \mathbf{u}(t)^T \right\rangle = \mathbf{Q} \left\langle \mathbf{y}(t)\, \mathbf{y}(t)^T \right\rangle \mathbf{Q}^T = \mathbf{Q}\mathbf{I}\mathbf{Q}^T = \mathbf{I}, \tag{4.20}$$

where we have used the orthogonality of $\mathbf{Q}$. Since $\left\langle \mathbf{u}(t)\,\mathbf{u}(t)^T \right\rangle = \mathbf{I}$ also the unit variance of the whitened data $\mathbf{y}(t)$ is preserved. An additional property of $\mathbf{W}$ can be derived using the relation

$$\left\langle \mathbf{y}(t)\,\mathbf{y}(t)^T \right\rangle = \mathbf{W}\left\langle \mathbf{x}(t)\,\mathbf{x}(t)^T \right\rangle \mathbf{W}^T = \mathbf{W}\mathbf{A}\left\langle \mathbf{s}(t)\,\mathbf{s}(t)^T \right\rangle \mathbf{A}^T\mathbf{W}^T = \mathbf{I}. \tag{4.21}$$

If we assume that the source signal components have unit variance this implies $\mathbf{W}\mathbf{A}\mathbf{A}^T\mathbf{W}^T = \mathbf{I}$.

Sometimes whitening is referred to as sphering. The term sphering originates from probability density functions that have spherical symmetry (e.g. a random variable with multivariate Gaussian probability density function with zero mean and unit covariance matrix). However, it is also used for probability density functions that show no spherical symmetry.

Usually $\mathbf{W}$ is an $N \times N$ matrix. But the whitening step can also be used to reduce the dimensionality of the observed signal. This can be done like in PCA (cf. (4.5)). Instead of using all eigenvectors and -values only the first l are taken into account. $\mathbf{V}_l$ is therefore an $N \times l$ matrix and $\mathbf{D}$ an $l \times l$ matrix resulting in a whitening transformation $\mathbf{W}$ which is an $l \times N$ matrix. Thus, we derive an l-dimensional whitened signal $\mathbf{y}(t)$.

Determining Q from Higher-Order Statistics

It can be shown that after the whitening step an orthogonal transformation $\mathbf{Q}$ on $\mathbf{y}(t)$ is sufficient to yield independent components [Comon, 1994b]. Thus the linear unmixing of $\mathbf{x}(t)$ can be achieved by the two transformations $\mathbf{W}$ and $\mathbf{Q}$ yielding the desired independent estimated source-signal-components

$$\mathbf{u}(t) = \mathbf{Q}\mathbf{W}\mathbf{x}(t) = \mathbf{Q}\mathbf{y}(t)\,. \tag{4.22}$$

The ICA algorithm presented in Chapter 6 is a typical example for such a two-step ICA algorithm. After the whitening of the observed signal the orthogonal transformation $\mathbf{Q}$ is computed via an approximate diagonalization of higher-order cumulant-tensors. The connection between $\mathbf{Q}$ and cumulants in this case is established via the multilinearity property 2.41 of cumulants under linear transformations.

4.3.2 Contrast Function

Often, the linear transformation defining the unmixing is estimated through optimization of a contrast function. Gassiat [1988] first introduced contrast functions in the context of scalar blind deconvolution. It has been adapted by Comon [1994b] in the framework of independent component analysis. There are several ICA algorithms based on contrast optimization the first being from Comon [1994b]. For an overview of possible contrasts for ICA see [Comon, 2004].

Definition 4.3.1 (Contrast Function) *A contrast function for ICA is a mapping Ψ from the set of probability density functions $\{p_{\mathbf{x}}(\mathbf{x}) | \mathbf{x} \in \mathbb{R}^N\}$ to $\mathbb{R}$ such that the following holds:*

a. $\Psi(p_{\mathbf{x}}(\mathbf{x}))$ is invariant under permutation and scale changing of the components of $\mathbf{x}$

$$\Psi(p_{\mathbf{x}}(\mathbf{x})) = \Psi(p_{\mathbf{\Lambda P x}}(\mathbf{\Lambda P x}))\,, \tag{4.23}$$

with $\mathbf{\Lambda}$ being a diagonal matrix and $\mathbf{P}$ a permutation matrix.

b. If $\mathbf{x}$ has independent components x_i, then

$$\Psi(p_{\mathbf{x}}(\mathbf{x})) \geq \Psi(p_{\mathbf{A x}}(\mathbf{A x}))\,, \tag{4.24}$$

for all invertible matrices $\mathbf{A}$.

c. If $\mathbf{x}$ has independent components x_i, then

$$\Psi(p_{\mathbf{x}}(\mathbf{x})) = \Psi(p_{\mathbf{A x}}(\mathbf{A x})) \tag{4.25}$$

if and only if $\mathbf{A}$ is of the form $\mathbf{A} = \mathbf{\Lambda P}$ with $\mathbf{\Lambda}$ and $\mathbf{P}$ as defined in Property 1.

In the following we will use the abbreviated notation $\Psi(\mathbf{x})$ instead of $\Psi(p_{\mathbf{x}}(\mathbf{x}))$.

An example for a possible contrast function is given by

$$\Psi(\mathbf{x}) := -I(\mathbf{x}) , \tag{4.26}$$

with $I(\mathbf{x})$ being the multi information (3.12) of the components x_i of the random variable $\mathbf{x}$. $\Psi(\mathbf{x})$ defines a contrast over all orthogonal matrices with unit covariance [Comon, 1994b]. Since this contrast uses the information of all higher-order statistics it is not useful as a basis for ICA algorithms. In Chapter 3 we introduced an approximated version of (4.26) which allows the formulation of a simple intuitive contrast function (3.29). This contrast builds the basis for the ICA algorithm derived in the second part of this thesis.

The concept of contrast functions can only be understood in the framework of ICA since it needs a proper definition of statistical independence. In cases where Definition (4.3.1) can not be applied we will use the term objective function instead. For example the objective function of CuBICA2 (7.13), a method for solving the ICA problem based on second-order statistics, has not been proven to fulfill the conditions (4.23-4.25).

4.4 Independent Component Analysis Based on Second-Order Statistics

There exists a variety of algorithms performing ICA and therefore BSS. They can be divided into two classes [Cardoso, 2001]: (i) independence is achieved by optimizing a criterion that requires higher order statistics; (ii) the optimization criterion requires auto-correlations or non-stationarity of the source signal components. For the second class of ICA algorithms second-order statistics is sufficient. In the following, we will call these kind of algorithms second-order ICA.

Consider a signal component without any temporal auto-correlation (e.g. white noise) and a second signal component which is the first signal component shifted slightly in time. Applying the measure of independence as mentioned above (3.17), the two signal components appear independent, although they are intuitively strongly dependent. By using a different measure based upon cross correlations instead, one will discover that these two components are dependent.

The second-order independent component analysis algorithm which we will discuss in Chapter 7 uses this latter measure of independence. Two signal components are considered statistically independent if they have zero time-delayed cross-correlations. There are several algorithms performing second-order ICA ([Belouchrani et al., 1997; Molgedey and Schuster, 1994; Nuzillard and Nuzillard, 2003; Zibulevsky and Pearlmutter, 2000; Ziehe and Müller, 1998]). Note, that the source signal components need to have auto-correlations in order to recover them from an unknown mixture. But, since the method does not depend on higher-order statistics it is also able to recover signal components with a Gaussian distribution.

Usually the algorithms follow the same two-stage approach like those based on higher-order statistics, i.e. first the input signal is whitened, and then the orthogonal matrix is estimated by optimizing an objective function leading to mutually independent signal components (see Sec. 4.3.1). However, there are also algorithms that estimate the unmixing matrix in a single step, e.g. [Yeredor, 2002] and [Ziehe et al., 2003a].

4.5 Nonlinear Blind Source Separation

An obvious extension to the linear mixing model (4.7) has the form

$$\mathbf{x}(t) = F(\mathbf{s}(t)) , \tag{4.27}$$

with a nonlinear function $F(\cdot)\,\mathbb{R}^N \rightarrow \mathbb{R}^M$ that maps N-dimensional source vectors $\mathbf{s}(t)$ onto M-dimensional signal vectors $\mathbf{x}(t)$. The components $x_i(t)$ of the observable are a nonlinear mixture of the sources and like in the linear case source signal components $s_i(t)$ are assumed to be mutually statistically independent. Extracting the source signal is in general only possible if $F(\cdot)$ is an invertible function.

The equivalence of BSS and ICA in the linear case does generally not hold for a nonlinear function $F(\cdot)$ [Hyvärinen and Pajunen, 1999; Jutten and Karhunen, 2003]. For example, given statistically independent components $u_1(t)$ and $u_2(t)$, any nonlinear functions $h_1(u_1(t))$ and $h_2(u_2(t))$ also lead to components that are statistically independent. Figure 4.1 gives a simple example of this indeterminacy.

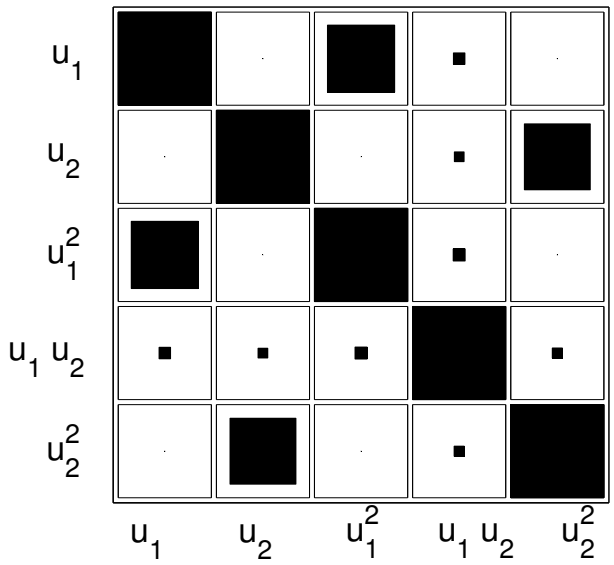


Figure 4.1: Mutual information between components of the signal $\mathbf{z}(t) = \mathbf{h}(\mathbf{u}(t)) := \left[u_1(t), u_2(t), u_1^2(t), u_1(t)u_2(t), u_2^2(t)\right]$. Large filled boxes denote high values of mutual information between the respective signal components (the components are statistically dependent, e.g. $u_1(t)$ and $u_1^2(t)$), small filled boxes denote small values of mutual information, (the components are mutually independent, e.g. $u_1(t)$ and $u_2(t)$). Thus, $u_1(t)$ is independent of $u_2(t)$ and also of $u_2^2(t)$. Assume, that $u_1(t)$ and $u_2(t)$ are the solutions to the nonlinear BSS problem (4.27). A nonlinear ICA algorithm is not able to distinguish between $u_1(t)$ resp. $u_2(t)$ and the squared components $u_1^2(t)$ resp. $u_2(t)^2$ because it is solely based on the assumption of mutual statistical independence between source signal components. ICA will therefore equally likely find the squared signal components as well as the original source signal components. To resolve this indeterminacy additional information about the source signal or the nonlinear mapping (4.27) is needed.

Thus, mutual independence of the extracted signal components is a necessary but not a sufficient condition to solve the nonlinear BSS problem. Moreover, a nonlinear mixture of $u_1(t)$ and $u_2(t)$ can still have statistically independent components (for an example see [Jutten and Karhunen, 2003]). To solve the nonlinear BSS problem additional assumptions about the mapping $F(\cdot)$ or the source signal are needed. We list some of the known methods:

- Constraints on the mapping $F(\cdot)$:
  - $F(\cdot)$ is a smooth mapping [Almeida, 2004; Hyvärinen and Pajunen, 1999]
  - $F(\cdot)$ is a post nonlinear (PNL) mapping [Taleb, 2002; Taleb and Jutten, 1997, 1999; Yang et al., 1998; Ziehe et al., 2003b]
- Prior information about the source signal components:
  - source signal components are bounded [Babaie-Zadeh et al., 2002]
  - source signal components have time-delayed cross-correlations (referred to as temporal correlations) [Hosseini and Jutten, 2003]

– source signal components are those that are extracted in the presence of injected noise [Harmeling et al., 2003]

4.6 Diagonalization Scheme

Assume, a vectorial signal $\mathbf{u}(t)$ with statistically independent components $u_i(t)$ is given. Furthermore, assume that all sample cumulants $C^{(\mathbf{u})}_{\cdots}$ of all orders can be computed. We know from Section 2.2.2 that all cumulants of a given order n form an nth-order tensor. Additionally, the Property 2.2.7 states that such a tensor with $n > 1$ is diagonal, e.g. if the signal components $u_i(t)$ have zero mean and are uncorrelated (all cross-cumulants or covariances of second-order vanish) their covariance matrix (the corresponding second-order tensor) is diagonal. Since ICA algorithms search for signals with independent components it is therefore useful to define a diagonalizing scheme for cumulant tensors of different orders. Several ICA algorithms adopt such a scheme. The basic ICA algorithm presented in Chapter 6 for example jointly diagonalizes the third- and fourth-order cumulant-tensor of the whitened observed signal. JADE, an ICA algorithm by Cardoso and Souloumiac [1993] diagonalizes several fourth-order cumulant-matrices and CuBICA2 (see Chapter 7) as well as most ICA methods based on second-order statistics e.g. [Belouchrani et al., 1997; Ziehe and Müller, 1998] jointly diagonalize time delayed correlation matrices of the observed signal.

The Diagonalization scheme for all those algorithms is based on successive applications of Givens or plane rotations. The scheme for diagonalizing a single matrix is based on work by Jacobi, an extension to several matrices has been introduced by Cardoso and Souloumiac [1996]. Comon [1994a] extended the method to the diagonalization of higher-order cumulant-tensors. All these methods have in common that they need to compute the rotation angle that defines each Givens rotation.

In the following we introduce the concept of Givens rotations and the Jacobi method to diagonalize a matrix. Furthermore, we derive a way to calculate the rotation angle of a Givens rotation in a linear way, independent of whether we want to calculate Givens rotations for matrices or third- or fourth-order cumulants. This is in contrast to the methods introduced by [Cardoso and Souloumiac, 1996] and [Comon, 1994a]. In [Cardoso and Souloumiac, 1996] a two-dimensional eigenvalue problem has to be solved to calculate a single rotation angle. This includes finding the roots of polynomials of degree two. In the case of fourth-order tensor diagonalization (cf. [Comon, 1994a]) the roots of degree four polynomials have to be computed.

4.6.1 Givens Rotations

Definition 4.6.1 (Givens Rotation) *A Givens rotation is a rotation around the origin within the plane of two selected components μ and ν and has the matrix form*

$$\mathbf{Q}^{\mu\nu} := \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos(\theta) & \cdots & \sin(\theta) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin(\theta) & \cdots & \cos(\theta) & \cdots & \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}, \tag{4.28}$$

with entries defined by

$$Q_{ij}^{\mu\nu} := \begin{cases} \cos(\phi) & for\ (i,j) \in \{(\mu,\mu),(\nu,\nu)\} \\ \sin(\phi) & for\ (i,j) \in \{(\mu,\nu)\} \\ -\sin(\phi) & for\ (i,j) \in \{(\nu,\mu)\} \\ \delta_{ij} & otherwise \end{cases} \tag{4.29}$$

with Kronecker symbol δ_{ij} and rotation angle ϕ.

Any orthogonal $N \times N$ matrix such as $\mathbf{Q}$ can be written as a product of $N(N-1)/2$ (or more) Givens rotation matrices $\mathbf{Q}^{\mu\nu}$ (for the rotation part) as defined above and a diagonal matrix with diagonal elements ± 1 (for the reflection part). Since reflections do not matter in our case we only consider the Givens rotations.

4.6.2 Jacobi Method

The Jacobi method of diagonalizing a matrix $\mathbf{M}$ can be described as the iterative optimization of a diagonality criterion under Givens rotations. We first define the diagonality criterion:

Definition 4.6.2 (Off) *The* **off** *of an $N \times N$ matrix $\mathbf{M}$ is defined by*

$$\mathbf{off}(\mathbf{M}) \quad = \quad \sum_{\substack{i,j=1 \\ i \neq j}}^{N} |M_{ij}|^2, \tag{4.30}$$

where M_{ij} are the matrix entries.

In the ideal case if $\mathbf{off}(\mathbf{M}) = 0$, matrix $\mathbf{M}$ is diagonal, since the sum of the squared off-diagonal entries vanishes. Using (4.30) we can define an objective, subject to minimization, that diagonalizes matrix $\mathbf{M}$ when applying a rotation matrix $\mathbf{Q}$

$$\Psi_{\text{diag}} = \mathbf{off}\left(\mathbf{Q}\mathbf{M}\mathbf{Q}^T\right). \tag{4.31}$$

Minimization is achieved by successively applying Givens rotations $\mathbf{Q}^{\mu\nu}$ in all possible planes. The global rotation $\mathbf{Q}$ is given by

$$\mathbf{Q} = \prod_{\mu,\nu} \mathbf{Q}^{\mu\nu}, \tag{4.32}$$

where usually several sweeps of Givens rotations $\mathbf{Q}^{\mu\nu}$ have to be applied.

The calculation of the optimal rotation angle for each Givens rotation is straightforward. First, consider for simplicity $N = 2$. The Givens rotation matrix $\mathbf{Q}^{1,2} = \mathbf{Q}$ looks like

$$\mathbf{Q}(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{4.33}$$

Write the diagonalization objective function as

$$\Psi_{\text{diag}} \quad = \quad \mathbf{off}\left(\mathbf{Q}\mathbf{M}\mathbf{Q}^T\right) \tag{4.34}$$

$$\quad = \quad \sum_{\substack{i,j=1 \\ i \neq j}}^{2} \left(\sum_{kl}^{2} Q_{ik} Q_{jl} M_{kl} \right)^2. \tag{4.35}$$

Now insert (4.33) into (4.35) to obtain

$$\Psi_{\text{diag}} = \sum_{i=0}^{2} c_i \left(\cos(\phi)^{4-i} \sin(\phi)^i + (-1)^i \sin(\phi)^{4-i} \cos(\phi)^i \right). \tag{4.36}$$

The derivation of this relation is described and the constants therein are defined in Appendix A.1. Further simplifications lead to

$$\Psi_{\text{diag}}(\phi) \quad = \quad A_0 + A_4 \cos(4\phi + \phi_4), \tag{4.37}$$

with constants defined in Appendix A.1. The Givens rotation angle which minimizes $\Psi_{\text{diag}}(\phi_m in)$ is therefore given by

$$\phi_{\min} = -\phi_4/4. \tag{4.38}$$

Thus, the optimal angle minimizing (4.35) can be computed in closed form in a linear fashion, which is in contrast to the formulation of [Cardoso and Souloumiac, 1996].

In Chapter 6 we derive a similar Givens-rotation-objective as (4.37) for the simultaneous diagonalization of third- and fourth-order cumulant-tensors. To diagonalize $N \times N$ matrices we can construct the global rotation matrix according to (4.32). In Chapter 7 we will use this method in order to diagonalize several matrices simultaneously. The extension to the case of simultaneous diagonalization is straightforward and will be explained there.

4.6.3 Invariances under Givens Rotations

Applying a Givens rotation $\mathbf{Q}^{\mu\nu}$ in the $\mu\nu$-plane changes all covariances $C^{(\mathbf{u})}_{\cdot\cdot}(\tau)$ with at least one of the indices equal to μ or ν. There exist two invariances under such transformation which can be described as

$$\left(\mathbf{C}^{(\mathbf{u})}_{\mu i}(\tau)\right)^2 + \left(\mathbf{C}^{(\mathbf{u})}_{\nu i}(\tau)\right)^2 \quad = \quad \text{const.} \; \forall \, \text{i} \notin \{\mu, \nu\}, \tag{4.39}$$

$$\left(\mathbf{C}^{(\mathbf{u})}_{\mu\mu}(\tau)\right)^2 + \left(\mathbf{C}^{(\mathbf{u})}_{\mu\nu}(\tau)\right)^2 + \left(\mathbf{C}^{(\mathbf{u})}_{\nu\mu}(\tau)\right)^2 + \left(\mathbf{C}^{(\mathbf{u})}_{\nu\nu}(\tau)\right)^2 \quad = \quad \text{const.}. \tag{4.40}$$

See Appendix A.2 for the exact derivation of the two invariances.

4.7 Performance Measure

A perfect performance in the sense of ICA is achieved if the product of the estimated unmixing matrix $\mathbf{R}$ and the mixing matrix $\mathbf{A}$ equals the identity matrix plus arbitrary permutations and scaling

$$\mathbf{RA} = \mathbf{P\Lambda}, \tag{4.41}$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{\Lambda}$ a diagonal matrix. The product of these two matrices is called the performance matrix $\mathbf{M}^{\text{perf}} = \mathbf{RA}$ [Amari et al., 1995].

To quantify the performances we slightly modified an error measure proposed by Amari et al. [1995] and define the unmixing error

$$E = \frac{1}{N^2}\left(\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{\left|M^{\text{perf}}_{ij}\right|}{\max_k\left|M^{\text{perf}}_{ik}\right|} - 1\right) + \sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{\left|M^{\text{perf}}_{ij}\right|}{\max_k\left|M^{\text{perf}}_{kj}\right|} - 1\right)\right), \tag{4.42}$$

where $M^{\text{perf}}_{\cdot\cdot}$ denote the entries of $\mathbf{M}^{\text{perf}}$. Unmixing error E measures the difference between performance matrix $\mathbf{M}^{\text{perf}}$ and a permutation matrix, where the entries of $\mathbf{M}^{\text{perf}}$ are normalized to take scaling into account. E indicates good unmixing by low values and vanishes for perfect unmixing.

Of course the mixing matrix $\mathbf{A}$ must be known if we want to apply this performance measure. Thus, it only works for artificial mixtures. Since we know that independence is a sufficient criterion for recovering the original source signal in the linear case, we can apply a measure of statistical independence, like mutual information, to quantify the unmixing performance for real world data. In Chapter 9 we use a crude approximation of mutual information (9.12), namely the sum over squared cross-cumulants of fourth order, for this purpose. This kind of performance measure is not applicable in the nonlinear case.

5

Slow Feature Analysis

Slow Feature Analysis (SFA) is a method to determine functions that extract slowly varying signals from quickly varying, observed signals. This is generally achieved in a non-linear fashion. SFA has been developed for unsupervised learning of invariances in the framework of biological modeling [Wiskott and Sejnowski, 2002]. Early descriptions of the principle are given in [Hinton, 1989], [Földiák, 1991], and [Mitchison, 1991].

There exist several applications: In [Wiskott and Sejnowski, 2002] SFA has been used in a simple hierarchical model of the visual system and was able to learn invariances like translation, scale or contrast invariance. Berkes and Wiskott [2003] have trained SFA on image sequences and learned functions that share many properties with complex cells of the primary visual cortex. Additionally SFA was successfully applied to pattern recognition [Berkes, 2004] and to the estimation of driving forces underlying non-stationary time series [Wiskott, 2003a]. A toolkit for SFA implemented in Matlab is available online [Berkes, 2003].

This chapter gives a short description of the method as developed in [Wiskott and Sejnowski, 2002]. The next chapter will point out the relation between SFA and second-order ICA as discussed in [Blaschke et al., 2004], providing the means to find a simple objective function for our nonlinear BSS method.

5.1 Mathematical Formulation

Assume a vectorial input signal $\mathbf{x}(t) = [x_1(t), \ldots, x_M(t)]^T$ is given. The objective of SFA is to find an in general nonlinear input-output function $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_L(\mathbf{x})]^T$ such that the components of $\mathbf{u}(t) = \mathbf{g}(\mathbf{x}(t))$ are varying as slowly as possible. This can be achieved by minimizing the objective function

$$\Delta(u_i) \quad := \quad \left\langle \dot{u}_i^2(t) \right\rangle, \tag{5.1}$$

successively for each $u_i(t)$ under the constraints

$$\langle u_i(t) \rangle = 0 \qquad \text{(zero mean)}, \tag{5.2}$$

$$\left\langle (u_i(t))^2 \right\rangle = 1 \qquad \text{(unit variance)}, \tag{5.3}$$

$$\left\langle u_i(t)\, u_j(t) \right\rangle = 0 \ \ \forall\, j < i \qquad \text{(decorrelation and order)}, \tag{5.4}$$

where $\langle \cdot \rangle$ denotes averaging over time. As a measure of slowness we use the variance of the first derivative of $u_i(t)$ (5.1). Minimal $\Delta(u_i)$ therefore indicates a signal component with on average small slope. Constraints (5.2) and (5.3) ensure that the solution will not be the trivial solution $u_i(t) = \text{const}$. Constraint (5.4) provides uncorrelated output signal components and thus guarantees that different components carry different information.

To make the optimization problem easier to solve we consider the components $g_i(\cdot)$ of the input-output function to be a linear combination of a finite set of nonlinear functions. We can then split the optimization procedure into two parts: (i) nonlinear expansion of the input signal $\mathbf{x}(t)$ into a high-dimensional feature space, and (ii) solving the optimization problem in the feature space linearly.

5.1.1 Nonlinear Expansion

A common method to make nonlinear problems solvable in a linear fashion is nonlinear expansion. The observed signal components $x_i(t)$ are mapped into a high-dimensional feature-space according to

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)). \tag{5.5}$$

The dimension L of $\mathbf{z}(t)$ is typically much larger than that of the original signal. A common mapping is given by the monomials of degree one and two

$$\mathbf{h}(\mathbf{x}) = [x_1, \ldots, x_M, x_1x_1, x_1x_2, \ldots, x_Mx_M]^T - \mathbf{h}_0, \tag{5.6}$$

when given an M-dimensional signal $\mathbf{x}(t)$. The dimensionality of the feature space for the monomials of first and second degree is $L = M + M(M+1)/2$. Of course it is possible to take monomials of higher degree or other nonlinear functions. The constant vector $\mathbf{h}_0$ can be used to make the expanded signal mean free.

5.1.2 Solution of the Linear Optimization Problem

Given the nonlinear expansion, the nonlinear input-output function $\mathbf{g}(\mathbf{x})$ can be written as

$$\mathbf{g}(\mathbf{x}) \quad = \quad \mathbf{R}\mathbf{h}(\mathbf{x}) = \mathbf{R}\mathbf{z}, \tag{5.7}$$

where $\mathbf{R}$ is an $L \times L$ matrix which is subject to optimization. To simplify the optimization procedure we (i) choose the nonlinearities $\mathbf{h}(\cdot)$ such that $\mathbf{z}(t)$ is mean free and (ii) first find a transformation $\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t)$ to obtain mutually decorrelated components $y_i(t)$ with zero mean. Matrix $\mathbf{W}$ is a whitening matrix as in normal ICA:

$$\mathbf{u}(t) = \mathbf{Q}\mathbf{y}(t) = \mathbf{Q}\mathbf{W}\mathbf{z}(t) = \mathbf{R}\mathbf{z}(t) = \mathbf{g}(\mathbf{x}(t)), \tag{5.8}$$

where $\mathbf{y}(t)$ is the nonlinearly expanded and whitened signal with $\left\langle \mathbf{y}(t)\,\mathbf{y}(t)^T \right\rangle = \mathbf{I}$. To fulfill constraint (5.2) an appropriate constant term $\mathbf{h}_0$ can be chosen. The constraints (5.3), and (5.4) are fulfilled trivially if the transformation $\mathbf{Q}$, subject to learning, is an orthogonal matrix. This can be shown as follows. Define $\mathbf{q}_i := [Q_{i1}, Q_{i2}, \ldots, Q_{iN}]^T$ to be the ith row vector of $\mathbf{Q}$. Using (5.8) we can derive

$$\langle u_i u_j \rangle = \mathbf{q}_i^T \left\langle \mathbf{y}(t)\,\mathbf{y}(t)^T \right\rangle \mathbf{q}_j = \mathbf{q}_i^T \mathbf{q}_j = \delta_{ij} \tag{5.9}$$

where we have used the orthonormality of the $\mathbf{q}_i$. Thus, constraints (5.4) and (5.4) are fulfilled.

To solve the optimization problem we rewrite the slowness objective (5.1)

$$\Delta(u_i) = \mathbf{q}_i^T \langle \dot{\mathbf{y}}\dot{\mathbf{y}}^T \rangle \mathbf{q}_i =: \mathbf{q}_i^T \mathbf{E}\mathbf{q}_i. \tag{5.10}$$

For this optimization problem there exists a unique solution. For $i = 1$ the optimal weight vector is the normalized eigenvector that corresponds to the smallest eigenvalue of $\mathbf{E}$. The eigenvectors of the next higher eigenvalues produce the next slow components ($u_2, u_3, \ldots$ and so forth).

Thus, to extract all slow components the minimization problem can be formulated as an eigenvalue problem

$$\mathbf{E}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda} \tag{5.11}$$

where $\mathbf{\Lambda}$ denotes a diagonal matrix with Λ_{ii} the ith eigenvalue belonging to the eigenvector $\mathbf{q}_i$.

With this we can define a simple schedule for an SFA algorithm

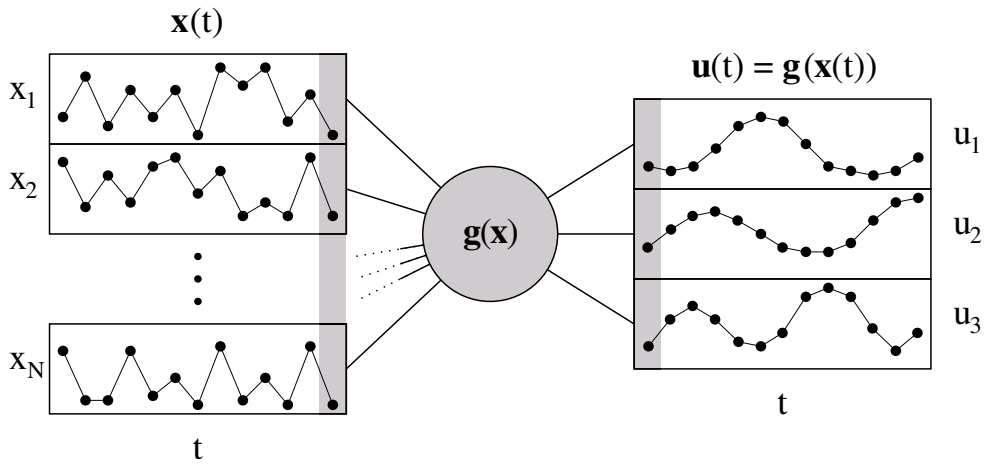


Figure 5.1: Illustration of the optimization problem solved by slow feature analysis. The observed signal $\mathbf{x}(t)$ varies quickly. Slow feature analysis finds the optimal input-output function $\mathbf{g}(\mathbf{x})$ such that the output signal $\mathbf{u}(t) = \mathbf{g}(\mathbf{x}(t))$ varies slowly. Figure courtesy of Dr. Laurenz Wiskott.

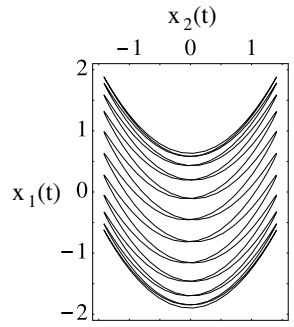
- Nonlinearly expand the input signal $\mathbf{x}$ into the feature space to obtain $\mathbf{z}$. Choose the nonlinearity such, that all components z_i have zero mean.
- Apply a whitening transformation $\mathbf{y} = \mathbf{W}\mathbf{z}$.
- Solve the eigenvalue problem $\mathbf{E}\mathbf{Q} = \langle \dot{\mathbf{y}}\dot{\mathbf{y}}^T \rangle \mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$ to obtain the slowly varying signal components $\mathbf{u} = \mathbf{Q}\mathbf{y}$.
- Sort the components of $\mathbf{u}$ by slowness.

5.2 Simple Example

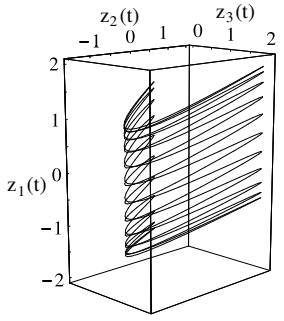
To illustrate the SFA procedure we will give a simple example adopted from [Wiskott and Sejnowski, 2002]. Assume the input signal $\mathbf{x}$ to SFA is given with components defined by

$$x_1(t) = \sin(t) + \cos^2(11t), \tag{5.12}$$
$$x_2(t) = \cos(11t), \tag{5.13}$$

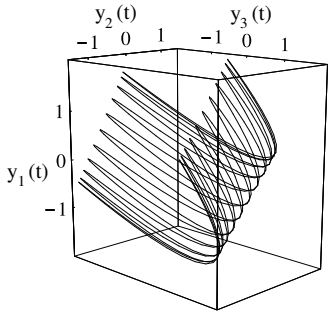
with $t \in [0, 2\pi]$. The signal $\sin(t)$ is the slowly varying signal we want to extract from $\mathbf{x}(t)$. The SFA procedure is illustrated in Figure 5.2.



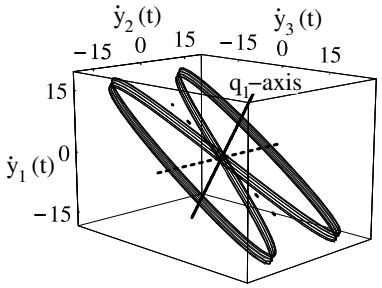
(a) Input signal $\mathbf{x}(t)$



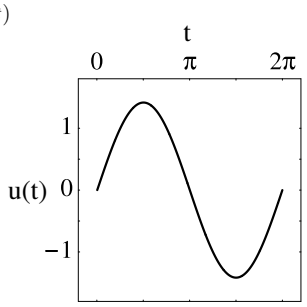
(b) Expanded signal $\mathbf{z}(t)$



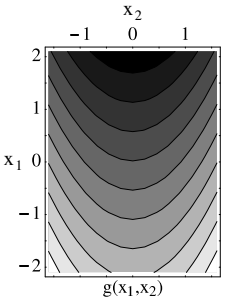
(c) Sphered expanded signal $\mathbf{y}(t)$



(d) Time derivative signal $\dot{\mathbf{y}}(t)$



(e) Output signal $\mathbf{u}(t)$



(f) Input-output function $g(\mathbf{x})$

Figure 5.2: Illustration of SFA by means of a simple example (5.13). **(a)** Input signal $\mathbf{x}$ is given by Equation (5.13), $\sin(t)$ denotes the slowly varying signal. **(b)** Expanded signal is defined as $\mathbf{z}(t) := \left[x_1(t), x_2(t), x_1(t)x_2(t), x_1^2(t), x_2^2(t)\right]^T$ only three of which are shown. **(c)** Sphered signal $\mathbf{y}(t)$ with zero mean and unit covariance matrix. **(d)** Time derivative signal $\dot{\mathbf{y}}(t)$. The direction of minimal variance is defined by $\mathbf{q}_1$. In this direction the sphered signal $\mathbf{y}(t)$ varies most slowly. The axes of next higher variances define $\mathbf{q}_2$ and $\mathbf{q}_3$. They are shown as dashed lines. **(e)** The Projection of $\mathbf{z}(t)$ onto the $\mathbf{q}_1$-axis gives the first output signal component u_1, which is $\sin(t)$. **(f)** The first component of the input-output function $g_1(x_1, x_2)$ is shown. Adapted from Wiskott and Sejnowski [2002].

Part II

Linear Blind Source Separation

6

Linear ICA based on Third- and Fourth-Order Cumulants

In this chapter we extend the cumulant based methods for ICA and present an improved algorithm that takes third- and fourth-order cumulants into account simultaneously. At the same time it is simpler and faster than Comon's algorithm [1994a], which our algorithm is based upon. Beside Comon's algorithm there exist other methods based on higher-order Cumulants, e.g. the earliest method proposed for BSS resp. ICA is based on the cancellation of high-order cross moments [Comon et al., 1991; Jutten and Herault, 1991]. Also Yellin and Weinstein [1996] and Girolami and Fyfe [1996] use higher-order statistics in order to recover the source signal. The well known FastICA algorithm [Hyvärinen, 1999] can be formulated based on kurtosis, too.

The new algorithm is described in Section 6.1. An approximation of the contrast function, building the basis of the algorithm, is introduced in Section 6.2. Section 6.3 provides with a new visualization of the contrast function for a three-dimensional ICA problem. A performance comparison with other algorithms is given in Section 6.4. We conclude with a brief discussion in Section 6.5. All simulations were done with Matlab (Version 6.0); analytical calculations were supported by Mathematica (Version 5.0), both working on Linux. The results presented in this chapter have partly been published in [Blaschke and Wiskott, 2002] and [Blaschke and Wiskott, 2004].

6.1 Improved ICA Algorithm

6.1.1 Cumulants and Independence

We start with a rather intuitive approach. Consider the standard linear ICA model as described in Section 4.2

$$\mathbf{u} = \mathbf{R}\mathbf{x} = \mathbf{R}\mathbf{A}\mathbf{s}, \tag{6.1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ is a linear mixture of the source signal $\mathbf{s} = [s_1, s_2, \ldots, s_N]$, and $\mathbf{u} = [u_1, u_2, \ldots, u_N]$ is the estimated source signal with statistically independent components u_i.

We can calculate higher-order cumulants $C^{(\mathbf{u})}_{\ldots}$ of the estimated source signal components u_i as explained in Section 2.2.2. Cumulants of a given order form a tensor (see Section 2.2.2). The off-diagonal elements or cross-cumulants (e.g. all cumulants with $ijkl \neq iiii$ in the case of fourth-order cumulants) characterize the statistical dependencies between components. If and only if all components u_i are statistically independent, the off-diagonal elements vanish and the cumulant tensors (of all orders) are diagonal (see Prop. 2.2.7).

Thus, a possible ICA algorithm finds an unmixing matrix that diagonalizes the cumulant tensors $C^{(\mathbf{u})}_{\ldots}$ of all orders of the output data u_i, at least approximately. The first order cumulant tensor is a vector and does

not have off-diagonal elements. The second order cumulant tensor can be diagonalized easily by whitening the input data as defined in Section 4.3.1. Referring to the second step in the approach described in the same section we now need to diagonalize higher-order cumulants, where we will only include cumulants up to fourth order. In general there is no orthogonal matrix that diagonalizes the third- or fourth-order cumulant tensor, thus the diagonalization of these tensors can only be done approximately. Therefore we need to define an optimization criterion for this approximate diagonalization which is done in the next section. The criterion is a simple extension of the diagonalization scheme developed in Section 4.6. The difference is that we now want to diagonalize a cumulant tensor and not a matrix.

6.1.2 Contrast Function

In order to formalize the approximate diagonalization of the cumulant tensors of order three and four we define the following criterion

$$\bar{\Psi}_{34}(\mathbf{u}) := \frac{1}{3!} \sum_{ijk \neq iii}^{N} \left(C_{ijk}^{(\mathbf{u})} \right)^2 + \frac{1}{4!} \sum_{ijkl \neq iiii}^{N} \left(C_{ijkl}^{(\mathbf{u})} \right)^2 , \tag{6.2}$$

which is simply the sum over the squared third- and fourth-order off-diagonal elements and needs to be minimized. The factors $\frac{1}{3!}$ and $\frac{1}{4!}$ arise from the expansion of the Kullback Leibler divergence as developed in Section 2.3. The Kullback Leibler divergence provides an information theoretic approach to the ICA problem different to the more intuitive interpretation of cumulant-tensor diagonalization (see also Eq. (6.4)).

Since the square sum over all elements of a cumulant tensor is preserved under any orthogonal transformation $\mathbf{Q}$ of the underlying data $\mathbf{y}$ [Deco and Obradovic, 1996], one can equally well maximize the sum over the diagonal elements,

$$\Psi_{34}(\mathbf{u}) := \frac{1}{3!} \sum_{i=1}^{N} \left(C_{iii}^{(\mathbf{u})} \right)^2 + \frac{1}{4!} \sum_{i=1}^{N} \left(C_{iiii}^{(\mathbf{u})} \right)^2 , \tag{6.3}$$

instead of minimizing the sum over the off-diagonal elements (6.2). $\Psi_{34}(\mathbf{u})$ is obviously much simpler than $\bar{\Psi}_{34}(\mathbf{u})$. Notice that this is a contrast as defined in Section 4.3.2 because all functionals $\sum_i \left(C_{ii\ldots i}^{(\mathbf{u})} \right)^2$ of cumulants of order ≥ 2 are contrasts and their sum $\Psi_{34}(\mathbf{u})$ is a contrast, too [Comon, 2002]. For a more general approach to contrast functions see [Moreau and Thirion-Moreau, 1999].

We can interpret these findings in the sense of information theory by referring to the expression for the approximated mutual information (3.27) derived in Section 3.4

$$I(\mathbf{u}) \approx J(\mathbf{u}) - \frac{1}{12} \sum_{i=1}^{N} \left(C_{iii}^{(\mathbf{x})} \right)^2 - \frac{1}{48} \sum_{i=1}^{N} \left(C_{iiii}^{(\mathbf{x})} \right)^2 . \tag{6.4}$$

$I(\mathbf{u})$ is minimal for a signal $\mathbf{u}$ with independent components. Since $J(\mathbf{u}) = J(\mathbf{y})$, because of the invariance of the negentropy under linear invertible transformations (cf. 3.3), we can neglect this term in an optimization procedure. And furthermore, instead of minimizing $I(\mathbf{u})$ we can equally well maximize $-I(\mathbf{u})$. Thus, we derive the same objective function as (6.3).

Due to the multilinearity of the cumulants $C_{\ldots}^{(\mathbf{u})}$ (cf. Property 2.2.3) in $C_{\ldots}^{(\mathbf{y})}$, (6.3) can be rewritten as

$$\Psi_{34}(\mathbf{Q},\mathbf{y}) = \frac{1}{3!} \sum_{i=1}^{N} \Big(\underbrace{\sum_{jkl=1}^{N} Q_{ij} Q_{ik} Q_{il} C_{jkl}^{(\mathbf{y})}}_{C_{iii}^{(\mathbf{u})}} \Big)^2 + \frac{1}{4!} \sum_{i=1}^{N} \Big(\underbrace{\sum_{jklm=1}^{N} Q_{ij} Q_{ik} Q_{il} Q_{im} C_{jklm}^{(\mathbf{y})}}_{C_{iiii}^{(\mathbf{u})}} \Big)^2 . \tag{6.5}$$

$C_{\ldots}^{(\mathbf{y})}$ are the cumulants of the whitened data set $\mathbf{y}$ and $Q_{..}$ are the elements of the rotation matrix $\mathbf{Q}$. With $\mathbf{u} = \mathbf{Q}\mathbf{y}$ Equations (6.3) and (6.5) are formally related by $\Psi_{34}(\mathbf{u}) = \Psi_{34}(\mathbf{I},\mathbf{u}) = \Psi_{34}(\mathbf{Q},\mathbf{y})$. $\Psi_{34}(\mathbf{Q},\mathbf{y})$ is now subject to an optimization procedure to find the orthogonal matrix $\mathbf{Q}$ that maximizes it.

6.1.3 Givens Rotations

As mentioned above maximizing the objective in (6.5) is equivalent to diagonalizing the third- and fourth-order cumulant tensor simultaneously. This simply arises from the fact that we want to minimize all off-diagonal elements or cross-cumulants of both tensors. Furthermore, since we are searching for an orthogonal matrix $\mathbf{Q}$, we can derive a diagonalization scheme for higher-order cumulants, analogously to that developed in Chapter 4.6 for matrix diagonalization.

For simplicity and without loss of generality we now consider only the subspace of two selected components, so that the Givens rotation matrix becomes

$$\mathbf{Q}^{\mu\nu} = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{6.6}$$

Contrast function (6.5) can then be rewritten as $\Psi_{34}(\phi,\mathbf{y}) = \Psi_3(\phi,\mathbf{y}) + \Psi_4(\phi,\mathbf{y})$ with

$$\Psi_n(\phi,\mathbf{y}) := \frac{1}{n!}\sum_{i=0}^{n} d_{ni}\left(\cos(\phi)^{(2n-i)}\sin(\phi)^{i}\right) + \frac{1}{n!}\sum_{i=0}^{n} d_{ni}\left(\cos(\phi)^{i}(-\sin(\phi))^{(2n-i)}\right) \tag{6.7}$$

with some constants d_{ni} that depend only on the cumulants $C^{(\mathbf{y})}_{\cdots}$ before rotation (see Appendix B.1.1). To simplify this equation Comon [1994a] defined some auxiliary variables $\theta := \tan(\phi)$ and $\xi := \theta - \frac{1}{\theta}$ and derived

$$\Psi_3(\theta,\mathbf{y}) = \frac{1}{3!}\left(\theta + \frac{1}{\theta}\right)^{-3}\sum_{i=1}^{3} a_i\left(\theta^i - (-\theta)^{-i}\right), \tag{6.8}$$

$$\Psi_4(\xi,\mathbf{y}) = \frac{1}{4!}\left(\xi^2+4\right)^{-2}\sum_{i=0}^{4} b_i\xi^i \tag{6.9}$$

for (6.7), with some constants a_i and b_i depending on the cumulants before rotation. To maximize (6.8) or (6.9) one has to take their derivative and find the root giving the largest value for Ψ_3 or Ψ_4, respectively. With this formulation only either the third-order or the fourth-order diagonal cumulants can be maximized but not both simultaneously.

In a more direct approach and after some quite involved calculations using various trigonometric theorems, we were able to derive a contrast function that (i) combines third- and fourth-order cumulants, (ii) is mathematically much simpler, (iii) has a more intuitive interpretation, and (iv) is therefore easier to optimize and approximate. We found

$$\Psi_{34}(\phi,\mathbf{y}) = A_0 + A_4\cos(4\phi+\phi_4) + A_8\cos(8\phi+\phi_8) \tag{6.10}$$

with some constants A_0, A_4, A_8 and ϕ_4, ϕ_8 that depend only on the cumulants $C^{(\mathbf{y})}_{\cdots}$ before rotation (see Appendix B.1.2). The third term comes from the fourth order cumulants only while the first two terms incorporate information from the third- and the fourth-order cumulants. Contrast functions for third- or fourth-order cumulants only, i.e. Ψ_3 or Ψ_4, can be easily obtained by setting all fourth- or third-order cumulants to zero, respectively.

It is actually relatively easy to see that it is possible to write the contrast in such a simple form. Firstly, rotation by multiples of $\frac{\pi}{2}$ corresponds to a permutation of the two components possibly plus sign changes, which does not affect the value of the contrast. Therefore, Ψ_{34} has a periodicity of $\frac{\pi}{2}$ and can be written as a sum of cosine-functions with frequencies 0, 4, 8, 12, 16, etc. Secondly, the terms in (6.7) are products of at most eight $\sin(\phi)$ and $\cos(\phi)$ functions, which can lead at most to a frequency of 8. Taking together these two arguments it is clear that only the frequencies 0, 4, and 8 are present and the contrast can be written in the form of (6.10). Because of the $\frac{\pi}{2}$ periodicity it suffices to evaluate the contrast in the interval $[\phi_4 - \frac{\pi}{4}, \phi_4 + \frac{\pi}{4}]$.

De Lathauwer et al. [1996] derived a related formula for third-order cumulants only that is quadratic in $\sin(2\phi)$ and $\cos(2\phi)$ and can be transformed to an expression similar to (6.10).

6.1.4 Unmixing Algorithm

Unmixing for a whitened signal $\mathbf{y}$ with $N = 2$ components can now be achieved in four steps:

(i) Compute the constants in (6.10),

(ii) Find the angle ϕ_{max} that maximizes $\Psi_{34}(\phi, \mathbf{y})$ in (6.10),

(iii) Calculate the Givens rotation-matrix $\mathbf{Q}^{\mu\nu}$ according to (6.6), and

(iv) Apply it to the whitened signal $\mathbf{y}$ to obtain estimated source signal $\mathbf{u} = \mathbf{Q}^{\mu\nu}\mathbf{y}$.

Since Ψ_{34} is maximal, the cumulant tensors $C^{(\mathbf{u})}_{ijk}$ and $C^{(\mathbf{u})}_{ijkl}$ are as diagonal as possible according to contrast (6.5) and the estimated signal components u_i are maximally statistically independent.

There are different ways to find the angle ϕ_{max} that maximizes $\Psi_{34}(\phi, \mathbf{y})$. Since all constants in Equation (6.10) are known and ϕ_{max} has to lie in the interval $\left[\phi_4 - \frac{\pi}{4},\ \phi_4 + \frac{\pi}{4}\right]$ we simply calculate $\Psi_{34}(\phi, \mathbf{y})$ for 1000 equidistant values of ϕ covering this interval and took the angle with largest value. We also tested the Matlab built-in function *fminbnd* (Matlab 6.0) based on Golden Section search and parabolic interpolation, which was significantly slower, but found no difference in the unmixing performance.

For $N > 2$ the contrast maximization follows directly from the $N = 2$ case. We denote the contrast function for a selected pair μ, ν of components by $\Psi^{\mu\nu}_{34}(\phi^{\mu\nu}, \mathbf{y})$. Note that pairwise statistical independence of the signal components implies mutual independence of all signal components [Comon, 1994b]. Therefore it is sufficient to iteratively maximize all $\Psi^{\mu\nu}_{34}$ like in the case of $N = 2$ until $\phi^{\mu\nu}_{max}$ is smaller then a given threshold ε for every pair μ, ν. In practice this can take several sweeps through all pairs. Every sweep consists of $N(N-1)/2$ rotations.

After centering and whitening, a maximization schedule for $N > 2$ can be as follows:

(a) Initialize auxiliary variables $\mathbf{Q}' = \mathbf{I_n}$ and $\mathbf{y}' = \mathbf{y}$

(b) Choose a pair of components μ and ν (randomly or in any given order)

(c) Calculate the Cumulants that are needed for $\Psi^{\mu\nu}_{34}(\phi^{\mu\nu}, \mathbf{y}')$

(d) Find the angle $\phi^{\mu\nu}_{max}$ such that $\Psi^{\mu\nu}_{34}\left(\phi^{\mu\nu}_{max}, \mathbf{y}'\right)$ is maximal

(e) If $\phi^{\mu\nu}_{max} > \varepsilon$ update $\mathbf{Q}'$ according to $\mathbf{Q}' \rightarrow \mathbf{Q}^{\mu\nu}\mathbf{Q}'$

(f) Rotate the signal components: $\mathbf{y}' \rightarrow \mathbf{Q}^{\mu\nu}\mathbf{y}'$

(g) Go to step (b) unless all possible $\phi^{\mu\nu}_{max} \leq \varepsilon$ with $\varepsilon \ll 1$

(h) Set $\mathbf{Q} = \mathbf{Q}'$ and $\mathbf{u} = \mathbf{Q}\mathbf{y}$.

In the simulations presented below we will not use the ε criterion but simply set $\varepsilon = 0$ and go through all possible pairs a fixed number of times in order to have a common criterion for all cumulant based methods (see below).

We refer to this algorithm as CuBICA (**Cu**mulant **B**ased **I**ndependent **C**omponent **A**nalysis) and indicate the different variants by appending the order of cumulants used in the contrast. For example a variant with a contrast function based on 3rd and 4th order information is called CuBICA34. Approximate contrast functions (see below) are indicated by an additional 'a', e.g. CuBICA34a.

6.1.5 Convergence of CuBICA

Since Ψ_{34} is a contrast it has the property

$$\Psi_{34}(\mathbf{Eu}) \leq \Psi_{34}(\mathbf{u}) \;\; \forall \mathbf{E} \text{ orthogonal}, \tag{6.11}$$

if $\mathbf{u}$ has maximally independent components (cf. second property in Definition 4.3.1). In the algorithm one can divide $\Psi_{34}(\mathbf{Q}^{\mu\nu}, \mathbf{y}')$ in (6.5) for every new Givens rotation $\mathbf{Q}^{\mu\nu}$ into two parts. One part is not affected by the rotation and the other is $\Psi_{34}^{\mu\nu}(\phi, \mathbf{y}')$. Since $\Psi_{34}^{\mu\nu}(\phi, \mathbf{y}')$ is maximized, $\Psi_{34}(\mathbf{Q}^{\mu\nu}, \mathbf{y}')$ and therefore also $\Psi_{34}(\mathbf{Q}', \mathbf{y})$ have to increase monotonically with every rotation. But $\Psi_{34}(\mathbf{Q}', \mathbf{y})$ has an upper bound, and thus will converge to a maximum. Of course we cannot rule out that there might be local maxima although they have not been observed.

6.2 Approximation of Ψ_{34}

6.2.1 Empirical Approach

Empirically we have found that the third term, A_8, in (6.10) is small compared to the second one. In fact the amplitude of the third term is about one magnitude smaller than that of the second term, A_4, independently of the chosen data sets (see Fig. 6.1). This suggests to neglect the third term and write as an approximate criterion

$$\tilde{\Psi}_{34}(\phi, \mathbf{y}) = A_0 + A_4 \cos(4\phi + \phi_4) \,. \tag{6.12}$$

Note that $\tilde{\Psi}_{34}$ still takes third- *and* fourth-order cumulants into account. As in the exact case, unmixing criteria restricted to fourth-order cumulants, i.e. $\tilde{\Psi}_4$, can be easily obtained by setting all third-order cumulants to zero. The contrast function for third-order cumulants order only, $\tilde{\Psi}_3$, remains the same in the approximate form (6.12) since A_8 in (6.10) contains no information of third-order cumulants. Finding the maxima ϕ_{max} of (6.12) is trivial. They are the angles satisfying the condition

$$\phi_{max} = n\frac{\pi}{2} - \frac{\phi_4}{4}, \quad n \in \{0, \pm 1, \pm 2, \pm 3, \ldots\} \,. \tag{6.13}$$

The maximum we chose is simply $\phi_{max} = -\frac{\phi_4}{4}$.

6.2.2 Analytical Simplifications

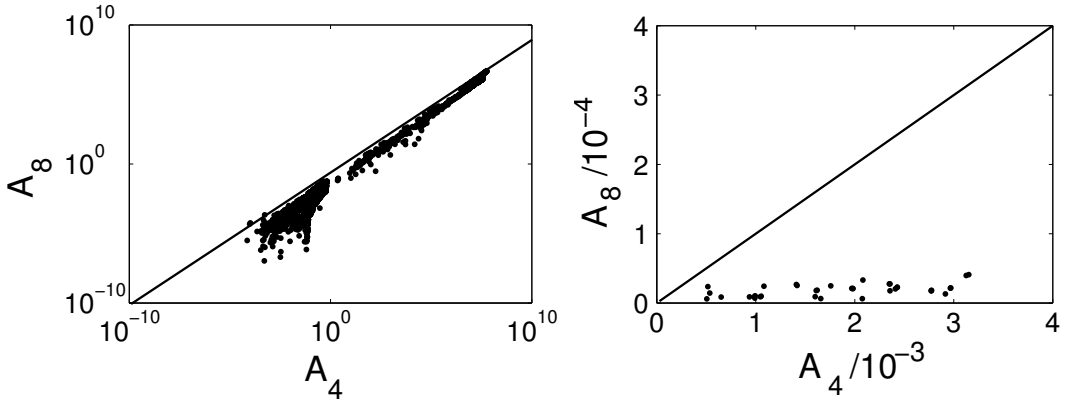
Since after whitening via $\mathbf{y} = \mathbf{Wx}$ we get uncorrelated signal components y_i with unit variance and zero mean, we know that finding the unknown source signal components can now be achieved by an orthogonal transformation $\mathbf{u} = \mathbf{Qy}$. This also means, that the whitened signal $\mathbf{y}$ can be expressed as $\mathbf{y} = \mathbf{Ms}$ where $\mathbf{M}$ defines an orthogonal transformation with $\mathbf{M} = \mathbf{\Lambda P Q}^T$. The diagonal matrix $\mathbf{\Lambda}$ defines possible scaling and $\mathbf{P}$ defines a permutation matrix.

Now, consider the case with two source signal components ($N = 2$). Furthermore, we neglect possible reflections and scaling. $\mathbf{M}$ then defines a rotation around the angle θ given by

$$\mathbf{M} \;=\; \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} . \tag{6.14}$$

All cumulants in $\mathbf{y}$ can therefore be written as a function of the rotation angle θ and the cumulants in $\mathbf{s}$. Thus, we can transform the cumulants in $\mathbf{y}$ involved in the objective function $\Psi_{34}^{\mu\nu}(\mathbf{y})$ according to 2.2.3 as

$$C_{ijkl}^{(\mathbf{y})} \;=\; \sum_{\alpha,\beta,\gamma,\delta=1,2} M_{i,\alpha} M_{j,\beta} M_{k,\gamma} M_{l,\delta} C_{\alpha\beta\gamma\delta}^{(\mathbf{s})} \,. \tag{6.15}$$



(a) (b)

Figure 6.1: Plot of amplitude A_8 versus A_4 for all rotations in two simulations with different data sets. The diagonal lines indicate $A_8 = 0.1 * A_4$. **(a)** Simulation with data set (v) (symmetrically distributed sources) and contrast function Ψ_4. Similar results were obtained by using Ψ_{34} as a contrast. Note the logarithmic axes. A_8 is about one magnitude smaller than A_4. Less than 1% of all values for $\frac{A_8}{A_4}$ exceed the 0.1-line. **(b)** Simulation with data set (ii) (non-symmetrically distributed sources) and contrast function Ψ_{34}. In this case the difference is even greater since for non-symmetrically distributed sources A_4 has additional terms from third order cumulants which do not appear in A_8.

Since $\mathbf{s}$ has only independent components, all cumulants in $\mathbf{s}$ except the auto-cumulants are zero and we can simplify to

$$C^{(\mathbf{y})}_{ijkl} = \sum_{\alpha=1,2} M_{i,\alpha} M_{j,\alpha} M_{k,\alpha} M_{l,\alpha} C^{(\mathbf{s})}_{\alpha\alpha\alpha\alpha} . \tag{6.16}$$

After some calculations (see Appendix B.1.3) we can derive the phases ϕ_4 and ϕ_8 of the objective (6.10) as functions of the rotation angle θ. They are given by

$$\phi_4 = 4\theta , \tag{6.17}$$

$$\phi_8 = 8\theta . \tag{6.18}$$

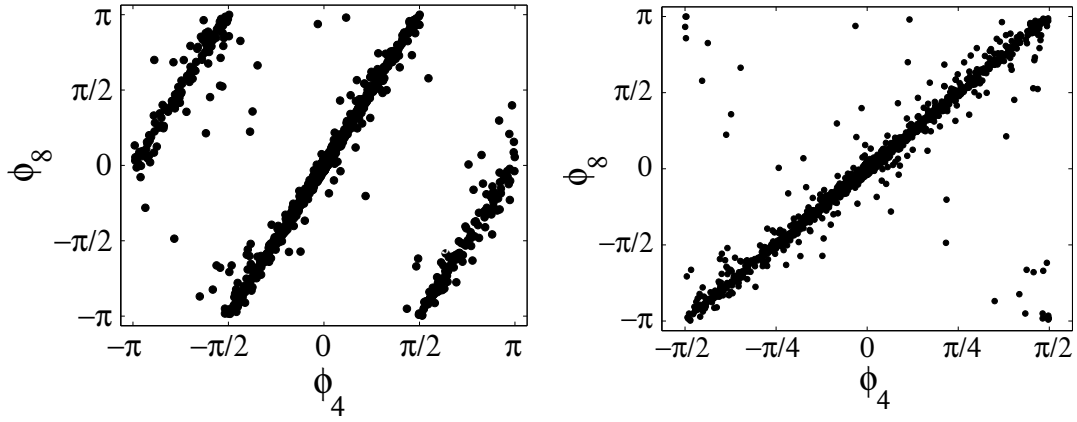
Thus the contrast function (6.10) can be written as

$$\Psi_{34}(\phi) = A_0 + A_4 \cos(4\phi + 4\theta) + A_8 \cos(8\phi + 8\theta) , \tag{6.19}$$

where the constants are combinations of cumulants in $\mathbf{s}$ (see Appendix B.1.3) and not of cumulants in $\mathbf{y}$ like in (6.10). However, the two contrast functions (6.10) and (6.19) are identical if $\mathbf{y} = \mathbf{Ms}$ and the source signal $\mathbf{s}$ has independent components. If not all source signal components are mutually independent this simplification will not work, since the approximations are due to the symmetry of the source signal (see (6.16)). The equivalence of ϕ_4 and ϕ_8 ((6.17)-(6.18)) implies that the second and third term on the right hand side of (6.19) reach their minima at the same rotation angle ϕ, namely at $\phi = -\theta$. Figure 6.2 shows the relation between ϕ_4 and ϕ_8 as calculated during an ICA simulation. However, the relation is not exact. This has at least two reasons: (i) the source signal components used in the simulation are not exactly independent (this would only be possible for signal components with infinite length); (ii) the relation (6.16) holds for exact cumulants, while in the simulation we are using sample cumulants. Due to frequency doubling, the third term has additional minima. But these minima always correspond to the maxima of the second term and can therefore be neglected. Thus, finding the absolute minimum of the objective function (6.19) is equivalent to finding the absolute minimum of

$$\bar{\Psi}_{34}(\phi) = A_4 \cos(4(\phi + \theta)) , \tag{6.20}$$

where we also omitted A_0, which is a constant term.



(a) (b)

Figure 6.2: (**a**) Scatterplot of phases ϕ_4 versus ϕ_8 of an ICA simulation. Phases where calculated for each Givens rotation step during the unmixing of 50 arbitrary mixtures of dataset (i). For each unmixing process 45 Givens rotations have been evaluated. The linear correlation between ϕ_4 and ϕ_8 can be seen easily. Since $\cos(4\phi - \phi_4) = \cos(4(\phi - \theta))$ and $\cos(8\phi - \phi_8) = \cos(8(\phi - \theta))$ the relation between the two is simply $\phi_8 = 2\phi_4$. The sidebands result from a shift of ϕ_4 by $\pm\pi$. This is equivalent to a $\pm 2\pi$ shift of ϕ_8. Since the contrast function (6.19) has a $\pi/4$ symmetry, the values in the sidebands correspond to the same minima as those in the mainband. The analytical result requires an exact correspondence between ϕ_4 and ϕ_8. (**b**) Same scatterplot but all $|\phi_4| > \pi/2$ are shifted by $\pm\pi$.

6.3 Visualization of the Contrast

Assuming whitened data $\mathbf{y}$, the contrast function based on cumulants in $\mathbf{u}$ can be written as a function of a rotation $\mathbf{Q}$ of the contrast based on cumulants in $\mathbf{s}$. For a specific rotation the contrast reaches its maximum, which implies $\mathbf{u} = \mathbf{P\Lambda s}$.

If $\mathbf{y}$ is a two-dimensional whitened mixture the ICA problem can be solved directly by computing the two-dimensional rotation $\mathbf{Q}^{1,2}$ (cf. (6.6)) that maximizes the contrast function $\Psi_{34}^{(2)} := \Psi_{34}(\mathbf{u}, N = 2)$, defined in Equation (6.3). The 2 stands for *two*-dimensional $\mathbf{u}$. The rotation $\mathbf{Q}^{1,2}$ in the u_1-u_2-plane can be parametrized by the angle ϕ. Thus, $\Psi_{34}^{(2)} = \Psi_{34}^{(2)}(\phi)$ is a function of the single rotation angle ϕ and can be easily visualized. Figure 6.3 shows $\Psi_{34}^{(2)}(\phi)$ as a function of ϕ. The $\pi/2$-periodicity of the contrast, as pointed out in Section 6.1.3, can be easily verified. Thus, there exist four equally good solutions to the ICA problem. The contrast in Figure 6.3 is virtually identical to a single cosine-function. This is in good accordance with the equivalent formulation of the contrast (6.3) defined in (6.10) if we neglect the last term, which can be done if the amplitude A_8 is small compared to A_4.

If $\mathbf{x}$ has three components there exist three possible rotation planes. The optimization of the corresponding contrast function $\Psi_{34}^{(3)} := \Psi_{34}(\mathbf{u}, N = 3)$ can be achieved by finding a 3×3 rotation matrix $\mathbf{Q}$ that maximizes $\Psi_{34}^{(3)}$. In three dimensions such a rotation can be parametrized by Euler angles and therefore a four-dimensional plot is necessary to visualize the shape of the contrast function $\Psi_{34}^{(3)}$. In general, given an N-dimensional whitened signal $\mathbf{y}$, a $N(N-1)/2$-dimensional plot is needed to visualize the respective shape of the contrast function.

Since it is still an open question whether the contrast function (6.3) has local optima for $N > 2$ such plots may be a good way to give some insights about the shape of the performance surface. Flockton

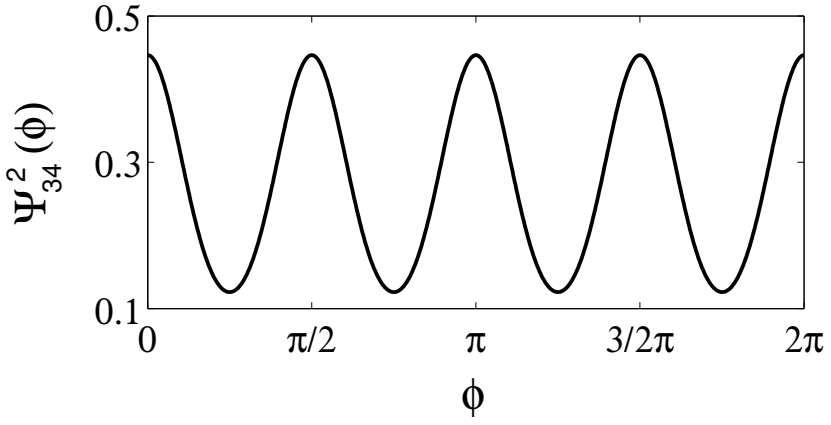


Figure 6.3: Plot of the contrast function $\Psi_{34}^{(2)}(\phi)$ as a function of ϕ, with $\mathbf{y}$ being a whitened mixture of two source signal components. The $\pi/2$-periodicity of $\Psi_{34}^{(2)}$ can be easily verified.

et al. [1996] showed the contrast surface in the case of $N = 3$. To circumvent the problem of visualizing a four-dimensional plot they considered the case where one of the three rotation angles (e.g. θ) was kept fixed. To get an impression of the whole contrast function several of such plots with different values of θ where used.

Here we use a method that allows us to show the main features of a contrast function of a three-dimensional ICA problem in a single three-dimensional plot. To simplify matters we first address the two-dimensional case ($N = 2$). For $N = 2$ The contrast is the sum of the two squared cumulants $\left(C_{1111}^{(\mathbf{u})}\right)^2$ and $\left(C_{2222}^{(\mathbf{u})}\right)^2$ plus a constant, that we neglect. We rotate the coordinate axes u_1 and u_2 by an angle ϕ. For each rotation angle ϕ the intersection of the rotated coordinate axes and the contrast is defined by the values of the respective squared cumulant (see Fig. 6.4). If we apply this method for every possible rotation angle we get a closed surface, we call the cumulant-energy surface. To read out the value of the contrast for a given rotation angle one has to add up the values determined by the intersection of the rotated coordinate axes with the cumulant-energy surface. The maximum of the contrast is reached at the point where the surface has maximal total distance from the origin. Obviously the contrast that can be read out from Figure 6.4 shows no local optima as we already know from (6.10). The maximum is reached for $\phi = 0, \pi/2, \pi, 3/2\pi$. Thus, as seen in Figure 6.3, there are four equally good solutions to this ICA problem. Figure 6.5 shows the cumulant-energy surface if the underlying source signal consists of three super-Gaussian components. It is highly symmetrical and shows no local minima. Since there exist six possible permutations of u_1, u_2 and u_3 and eight possible sign reversals there are 48 equivalent maxima of $\Psi_{34}^{(3)}$ and thus solutions to the ICA problem. As can be easily seen a successive maximization of the contrast function $\Psi_{34}^{(3)}$ by Givens rotations will always lead to a global maximum, independently of the starting point. Now, assume u_1 has a sub- and u_2 has a super-Gaussian distribution. A rotation in the u_1-u_2-plane by $\pi/2$ simply exchanges u_1 and u_2. Since u_1 (u_2) goes from negative (positive) kurtosis to positive (negative) kurtosis during such a rotation the cumulant energy-surface has to cross zero in the u_1-u_2-plane. This effect can be seen in Figure 6.6 where the cumulant energy-surface of two super-Gaussian and one sub-Gaussian components is shown. Note that the analytical approximation of the contrast function (6.20) postulates line symmetry of the two-dimensional contrast for exactly independent source signal components. Therefore, the two-dimensional energy-surface lying on each possible Givens rotation plane should also show line symmetry. However, the energy-surface in Figure 6.5 shows point symmetry. Thus, the underlying source signal components are not exactly independent. But, since this effect is small the contrast (6.20) is still a good approximation. In principle for large differences of the two phases ϕ_4 and ϕ_8 local maxima can emerge, but have not been observed.

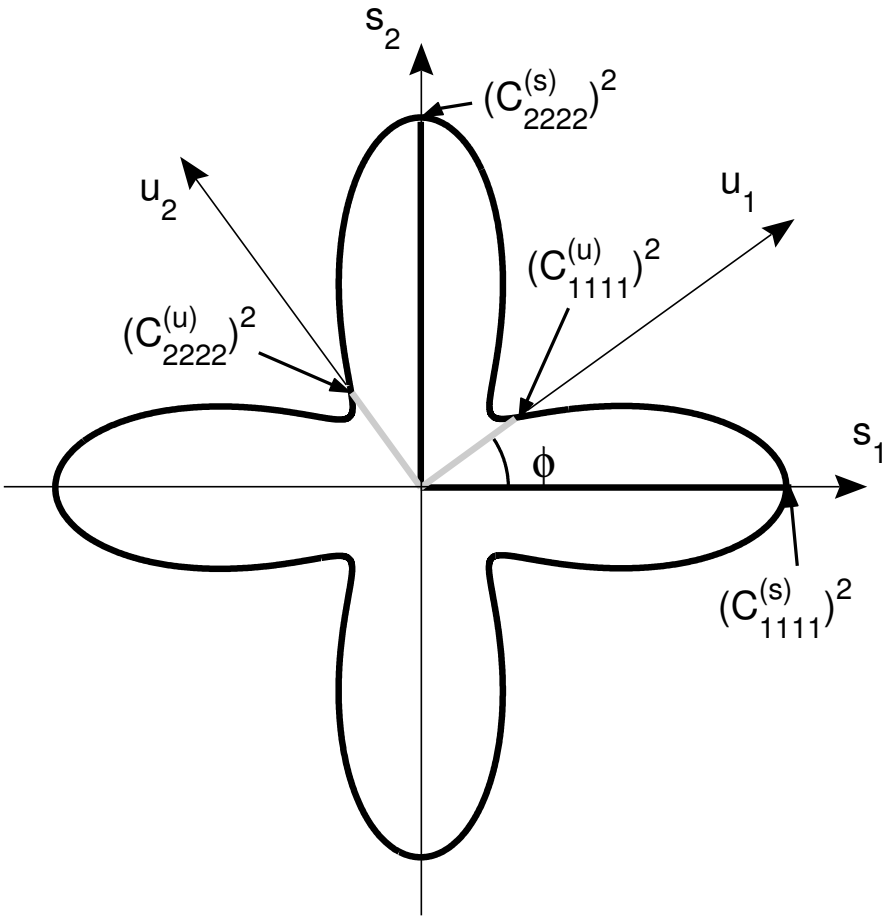
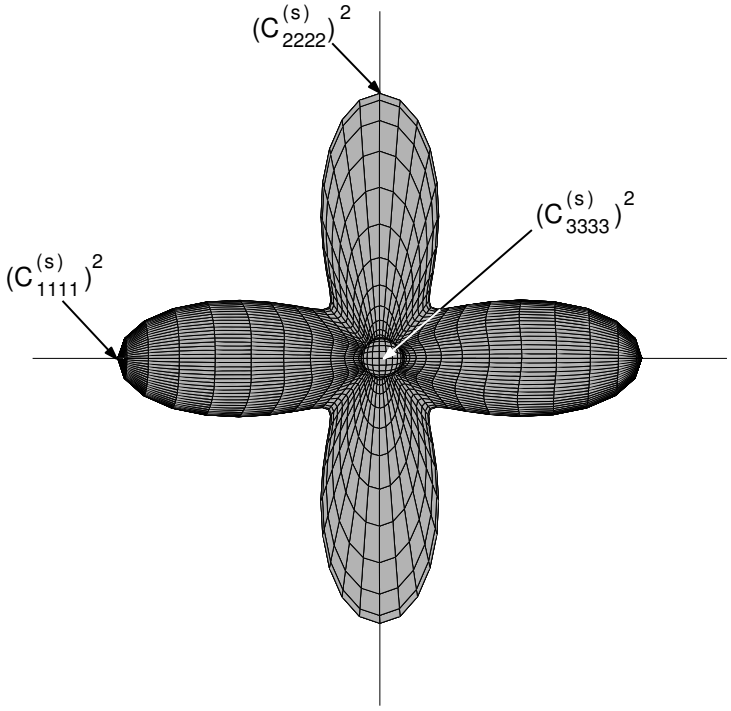
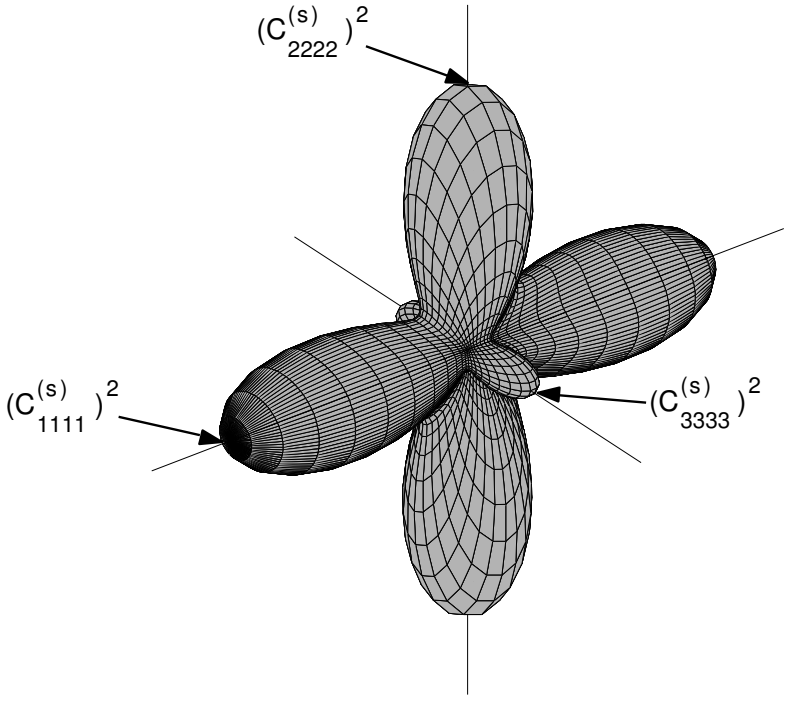


Figure 6.4: Cumulant-energy surface of a two-dimensional signal as a function of the rotation angle. Assuming a whitened mixture, the contrast $\left(C^{(\mathbf{u})}_{1111}\right)^2 + \left(C^{(\mathbf{u})}_{2222}\right)^2$ depends only on the rotation angle ϕ. The values of $\left(C^{(\mathbf{u})}_{1111}\right)^2$ resp. $\left(C^{(\mathbf{u})}_{2222}\right)^2$ at each possible angle forms the shape of the surface. To read out the value of the contrast at a given angle one needs to add up the values determined by the intersection of the rotated coordinate axes with the figure. For $\phi = 0, \pi/2, \pi, 3/2\pi$ the contrast reaches its maximum. In this case $\mathbf{u} = \mathbf{P\Lambda s}$, where $\mathbf{P}$ is a permutation matrix and $\mathbf{\Lambda}$ is a diagonal matrix with entries $|\Lambda_{ii}| = 1$.



(a)



(b)

See next page for figure caption

Figure 6.5: Cumulant-energy surface of three super-Gaussian distributed signal components. **a** Front view. **b** Side view. The coordinate axes show the reference coordinate system defined by unit vectors in the directions of s_1, s_2 and s_3. The maximal total distance of the surface from the origin is reached if the coordinate axes lie along these three unit vectors. Considering all possible permutations and reflections of s_1, s_2 and s_3 it can be seen that there exist 48 equally good maxima and therefore also equally good solutions to the ICA problem.

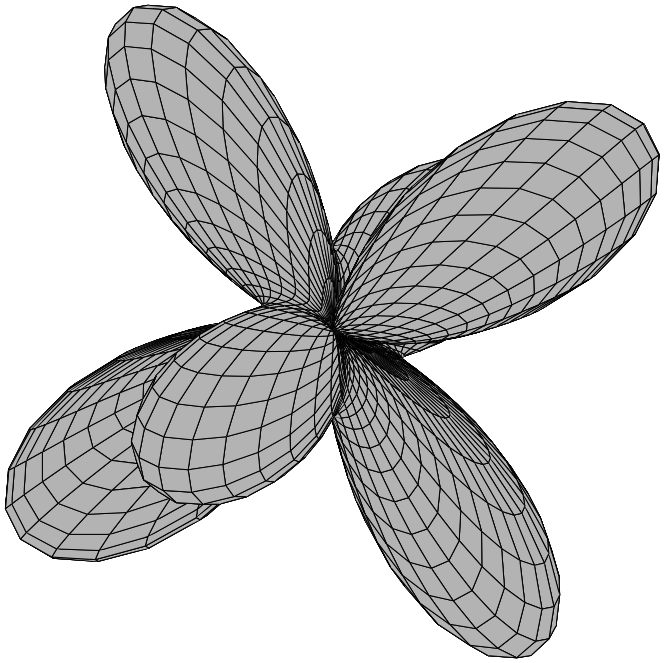


Figure 6.6: Cumulant energy-surface of two signal components with super-Gaussian and one component with sub-Gaussian distribution.

This highly symmetrical form of the cumulant energy-surface in three dimensions may be a hint that also for higher dimensional surfaces symmetries exist that prevent the cumulant energy-surface from having local optima.

6.4 Comparison with Other Algorithms

We compared four variants of CuBICA [1] with Comon's original algorithm based on $\Psi_4(\xi, \mathbf{y})$ [1994a], with the JADE algorithm [Cardoso and Souloumiac, 1993], which diagonalizes 4th order cumulant matrices, with the Infomax Algorithm [Lee et al., 1999], and with the FastICA package [Hyvärinen, 1999] using a fixed-point algorithm with different nonlinearities. In all cases we have used original software provided by the authors [2].

[1] CuBICA34 (based on $\Psi_{34}(\phi, \mathbf{y})$), CuBICA34a (based on $\tilde{\Psi}_{34}(\phi, \mathbf{y})$), CuBICA4 (based on $\Psi_4(\phi, \mathbf{y})$), and CuBICA4a (based on $\tilde{\Psi}_4(\phi, \mathbf{y})$)

[2] Comon's algorithm: `http://www.i3s.unice.fr/~comon/codesICA.txt` (Version 6 of March 1992, downloaded December 12th, 2001); JADE: `ftp://tsi.enst.fr/pub/jfc/Algo/Jade/jadeR.m` (Version 1.5 of December 1997, downloaded March 6th, 2001); FastICA: `http://www.cis.hut.fi/projects/ica/fastica/loadcode.shtml` (Version 2.1 of January 15th, 2001, downloaded January 15th, 2001); Infomax: `http://www.cnl.salk.edu/~tewon/ICA/Code/ext_ica_download.html` (Version 2.0 of August 23rd, 1998, downloaded March 6th, 2001); CuBICA: `http://itb.biologie.hu-berlin.de/~blaschke` (Version 1.6 of February 22th, 2002)

It is interesting to note that the different algorithms make different assumptions about the distributions of the sources. Infomax uses a one-parametric symmetrical model for the distributions of the sources and thus makes the assumptions very explicit. It is not clear to us which deviations of the true distribution from the model can degrade the unmixing performance and to what extent. Cumulant based methods, on the other hand, make no explicit assumptions about the source distributions. However, by focusing only on cumulants of low order and since cumulants of different order do not mix under a linear transformation, these methods are completely blind to higher order cumulants. Thus there is an implicit assumption that the distributions are such that low order cumulants contain enough information for the unmixing. Therefore considering fourth-order cumulants only is equivalent to a one-parametric model of the source distribution whereas considering third- and fourth-order cumulants results in a two-parametric family of functions. FastICA is similar in this respect using a one-parametric approach, although it is not restricted to cumulants but can also be derived using non-polynomial functions.

6.4.1 Simulations

We assembled five different data sets of length 44218. Data set (i) contained real acoustic sources from [John Fitzgerald Kennedy Library, Boston, 1996] and [Pearlmutter, 1996]. Data set (ii) contained non-symmetrically distributed sources and (v) was composed of different symmetrically distributed sub- and super-Gaussian sources, both sets were generated synthetically. Set (iii) and (iv) were mixtures of real acoustic and synthetic sources. For further details see Table 6.1. Each data set was mixed by a randomly chosen mixing matrix with entries chosen uniformly from $[-1,1]$.

To simplify the comparison between the algorithms we used the same stopping criterion for all four cumulant based methods, namely we stopped after M sweeps through all possible pairs of signal components, where M is the nearest integer to $1+\sqrt{N}$ and N is the number of source components. Unmixing performance did not depend significantly on the stopping criterion, but comparison of the time performances is clearer with a common stopping criterion. Since it is not easy to define a similar criterion for FastICA and Infomax we did not change these algorithms.

To quantify the performances we use the measure E defined in Section 4.7. An example of the development of E during a simulation is shown in Figure 6.7.

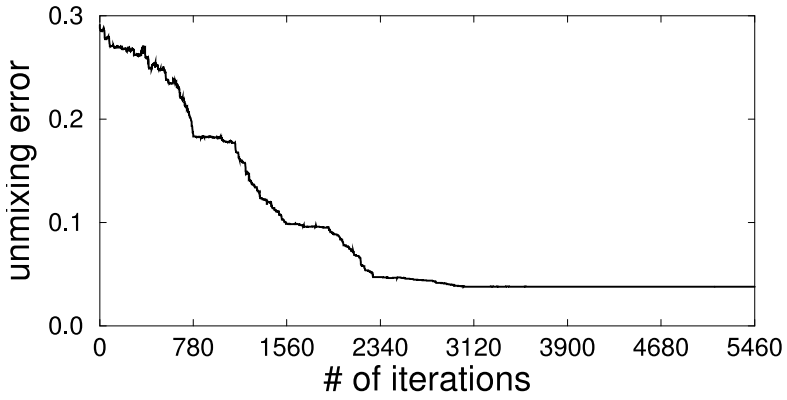


Figure 6.7: Development of the unmixing error for data set (v) from Table 6.1 with $N = 40$ using Ψ_{34} as a contrast. The algorithm was stopped after 7 sweeps through all $N*(N-1)/2 = 780$ possible pairs of signal components.

Infomax and FastICA required some manual assistance, while the other algorithms could be applied blindly. For Infomax we usually applied the algorithm to the data twice with different parameter settings. The first run did a rough unmixing which was then refined in the second run on the already roughly unmixed data. Several test runs were necessary to find appropriate parameter settings, which was quite time consuming. For FastICA we had to do test runs to determine the nonlinearity yielding best performance.

In both cases the error criterion (4.42) guided the parameter selection, so that the results were not obtained completely unsupervised but with some supervision. Since the data sets were sufficiently long to rule out over-fitting, we did not use separate training and test sets in these experiments.

To investigate the dependency of the algorithms on the length of the data set, we also did simulations on data set (v) with different numbers of data points and compared the unmixing errors (see Fig. 6.8). Data set (v) was split into 11 subsets of length T, with $T \in \{40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480\}$ (for $T = 5120$, 10240, and 20480, there were only 8, 4, and 2 pieces, respectively). The first subset was used to optimize the parameters of FastICA and Infomax for one given mixing matrix. Then all algorithms were tested with ten different mixing matrices on each of the remaining subsets.

6.4.2 Results

We measured unmixing errors and the elapsed time for all 8 different algorithms and 5 data sets of full length (see Table 6.1). Since additional simulations with different mixing matrices showed no significant variations in the results, we only give here the mean values for unmixing error and time consumption.

For symmetrically distributed sources, all algorithms performed similarly well (data sets (i), (iv), (v)). If the sources were skew-symmetric, the additional third order information was crucial and CuBICA34, CuBICA34a, and FastICA using a corresponding nonlinearity clearly gave better results (data set (ii)). In case where the sources were both symmetric and skew-symmetric, only CuBICA34 and CuBICA34a with contrast function Ψ_{34} and $\tilde{\Psi}_{34}$, respectively, could discriminate between the different distributions (data set (iii)). Thus, the comparison in Table 6.1 suggests that different ICA-algorithms perform similarly well as long as their contrast functions are sensitive to the properties of the source distributions. Methods blind to skew-symmetric distributions fail on data set (ii) and only the methods that take third- *and* fourth- order cumulants into account can deal with mixtures of symmetric and skew-symmetric distributions (data set (iii)). The CuBICA-algorithms with approximate unmixing criterion gave similar unmixing errors as the algorithms using the exact contrast. This suggests that A_8 in Equation (6.12) is indeed negligible. However, since there is no advantage over the algorithms using the exact contrast in terms of CPU-time, this is mainly of theoretical interest.

The complexity of Comon's algorithm and CuBICA is of the same order. A significant difference is the way how the optimal rotation angle $\phi^{\mu\nu}$ is found. Comon's algorithm uses a Matlab function to numerically find the root of a polynomial of degree four in each step whereas CuBICA generates an array of function values and searches for the maximal value. In Matlab implementation CuBICA performs faster and gives in general slightly smaller unmixing errors. JADE is an algorithm based on kurtosis-maximization. It uses a matrix-approximation for cumulant tensors of 4th order. This may explain the less accurate performance on data set (v) but is also responsible for the relatively high speed, since matrices can be processed efficiently in the Matlab implementation. This speed advantage might be less significant in a C implementation, for instance, and vanishes for larger N (see data set (v)) because JADE needs to compute all N^4 possible cumulants of 4th order at least once. Comon's algorithm and CuBICA, on the other hand, only need to compute cumulants with at most two different indices. FastICA is significantly faster and Infomax is much slower than the cumulant based methods. Both algorithms have in common that in our experiments they needed some manual assistance. In FastICA we had to decide which nonlinearity should be used. This decision was guided by the unmixing error, a measure that is usually not available in more realistic applications. Infomax required some parameter tuning and repeated application to the data with different parameter sets which made the algorithm inconvenient to use. It also seems questionable whether the speed advantages of JADE and FastICA are worth the worse performance and the required manual assistance, respectively.

By comparing unmixing errors of the different algorithms depending on the number of data points N one can see from Figure 6.8 that all methods degrade similarly with shortened length of the data sets. One marked difference however was that Infomax had a large variance, while all other algorithms gave virtually identical results over different simulation runs. Thus although Infomax yielded best performance in some runs it performed worst in others and we found it to be unreliable, particularly on the short data sets. On the long data sets used for Table 6.1 Infomax was nearly as reliable as the other algorithms.

Unmixing error (E)					
contrast function/ algorithm	data sets, # of components (N)				
	(i) N=6	(ii) N=6	(iii) N=7	(iv) N=12	(v) N=40
CuBICA34	**0.017**	**0.039**	**0.041**	**0.038**	**0.039**
CuBICA34a	**0.018**	**0.040**	**0.042**	**0.038**	**0.038**
CuBICA4	**0.017**	0.31	0.11	**0.035**	**0.039**
CuBICA4a	**0.017**	0.32	0.12	**0.037**	**0.038**
Comon	**0.017**	0.25	0.14	**0.049**	**0.061**
JADE	**0.016**	0.30	0.11	**0.035**	0.10
Infomax	**0.018**	0.47	0.17	**0.043**	**0.035**
FastICA	**0.016**	**0.040**	0.11	**0.042**	**0.037**

Elapsed time / sec					
contrast function/ algorithm	data sets, # of components (N)				
	(i) N=6	(ii) N=6	(iii) N=7	(iv) N=12	(v) N=40
CuBICA34	**1.5**	**1.4**	**2.8**	**10.1**	230.3
CuBICA34a	**1.4**	**1.4**	**2.7**	**9.8**	227.6
CuBICA4	**1.3**	**1.3**	**2.6**	**9.5**	223.8
CuBICA4a	**1.4**	**1.4**	**2.7**	**9.8**	237.1
Comon	**2.4**	**2.3**	**4.3**	**14.1**	300.2
JADE	**0.7**	**0.7**	**1.1**	**5.4**	404.6
Infomax	48.1	49.3	57.8	112.1	512.3
FastICA	**1.7**	**0.5**	**0.5**	**6.2**	**16.8**

Table 6.1: Unmixing error (E) and CPU-time in seconds for different algorithms and data sets. Data sets: (i) 5 real acoustic sources from [John Fitzgerald Kennedy Library, Boston, 1996] + 1 normally distributed source ($N(0,1)$), (ii) 5 skew-normally distributed sources [Azzalini, 1985] + 1 normally distributed source, (iii) 3 music sources from [Pearlmutter, 1996] + 3 skew-normally distributed sources + 1 normally distributed source, (iv) 6 real acoustic sources (3 speech + 3 music sources) from [John Fitzgerald Kennedy Library, Boston, 1996] and [Pearlmutter, 1996] + 3 Laplace distributed sources + 1 normally distributed source + 1 skew-normally distributed source + 1 $\sin(0.05 * t)$, (v) 10 Beta distributed sources (super-Gaussian) + 10 Cauchy distributed sources (sub-Gaussian) + 10 Laplace distributed sources (super-Gaussian) + 10 Student-t distributed sources (sub-Gaussian). The number of data points for all data sets was T=44218. Additional 20 simulations with different mixing matrices showed no significant variations in the unmixing errors. Low values of E indicate good performance. Times have been measured on a 1.8 GHz Pentium IV PC using Matlab 6.0 implementation. Relatively small unmixing errors and short CPU-times are set bold face.

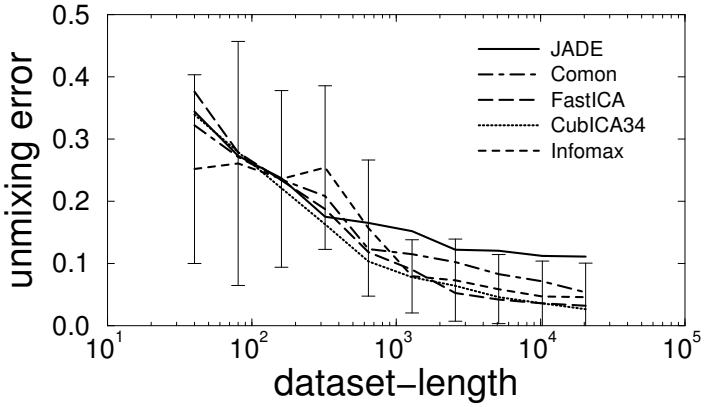


Figure 6.8: Mean unmixing errors for data set (v) from Table 6.1 with $N = 40$ components and different numbers of data points T with $T \in \{40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480\}$. For each T we took 10 different samples and performed 10 simulations on each, every simulation with a different mixing matrix. For $T = 5120, 10240$, and 20480 we used 7, 3, and 1 different samples, respectively, due to the length of the whole data set (v). One additional sample was used to train Infomax and FastICA. The standard deviation of the unmixing errors was less then 0.01 for all algorithms except Infomax. For Infomax the parameter set with smallest unmixing error was found on a training set mixed with a single mixing matrix. We used only one mixing matrix because finding a good parameter set for several mixing matrices was too time consuming since the algorithm did often not converge on one of the matrices. Results shown here are for the same test data sets as for the other algorithms. The error bars denote twice the standard deviation of the unmixing error of Infomax.

6.5 Summary

We have proposed CuBICA, an improved cumulant based method for independent component analysis. In contrast to Comon's method [1994a] and other algorithms (FastICA, Infomax, and JADE) it takes third- and fourth-order statistics into account simultaneously (CuBICA34) and is thus able to handle linear mixtures of symmetrically and skew-symmetrically distributed source signal components. Due to its mathematically simple formulation and since A_8 in Equation (6.12) is small compared to A_4, approximate algorithms, CuBICA34a and CuBICA4a, can be derived easily, which show equal performances. Although, this is mainly of theoretical interest, since the approximate algorithms are not significantly faster. Furthermore, in contrast to FastICA and Infomax, CuBICA can be used without any parameter adjustments.

Since CuBICA can handle symmetric and asymmetric distributed sources, is easy to use, and shows good performance, it may be a good general algorithm for performing ICA.

7

Linear ICA Based on Cumulants of Order Two

In this chapter we introduce an algorithm for linear ICA that, in contrast to the algorithms derived in the previous chapter, is based on second-order statistics. The measure of independence used for CuBICA34 is based on higher order statistics (see Section 6.1.2) and is thus not applicable in the case of second-order ICA. Therefore we use a different measure of statistical independence as described in Section 4.4. Nevertheless the method is also based on successive Givens rotations in order to obtain statistically independent output components. Starting with a brief introduction to the basic method in Section 7.1 the algorithm is described in Section 7.2 for a single time delay and in Section 7.3 for several time delays. A comparison with TDSEP, an algorithm that also uses second-order statistics, is given in Section 7.4.

7.1 Time Delayed Correlations

Molgedey and Schuster [1994] have been the first to propose an algorithm for linear ICA solely based on second-order cumulants. Analogously to the previous chapter one could consider to diagonalize the correlation matrix (cumulant tensor of second-order) $\left\langle \mathbf{x}(t)\mathbf{x}(t)^T \right\rangle$ of the mixture $\mathbf{x}(t)$. This is equivalent to a whitening (see Sec. 4.3.1). But, we know from [Comon, 1994b] that this is not sufficient to find the original source signal components. Furthermore, the unmixing matrix $\mathbf{R}$ solving the linear ICA problem $\mathbf{u}(t) = \mathbf{R}\mathbf{x}(t)$ is generally non-symmetric whereas the correlation matrix $\left\langle \mathbf{u}(t)\mathbf{u}(t)^T \right\rangle$ is symmetric. Thus for an ICA method based on second-order statistics it is not sufficient to consider only the symmetric correlation matrix. Jutten et al. [1988] therefore proposed to additionally measure nonlinear, non symmetric correlations like $\left\langle u_i(t)\left(u_j(t)\right)^3 \right\rangle$.

Instead Molgedey and Schuster [1994] suggested to simultaneously diagonalize the correlation matrix $\left\langle \mathbf{x}(t)\mathbf{x}(t)^T \right\rangle$ plus the time delayed correlation matrix $\left\langle \mathbf{x}(t)\mathbf{x}(t+\tau)^T \right\rangle$ with time delay τ. This is in contrast to other approaches where the required asymmetry is achieved by introducing nonlinearities in PCA like in [Karhunen and Joutsensalo, 1994]. Assuming that $\mathbf{x}(t)$ has zero mean, the ICA problem introduced by Molgedey and Schuster can be formulated as a generalized eigenvalue problem [Molgedey and Schuster, 1994]

$$\mathbf{C}_2^{(\mathbf{x})}(\tau)\mathbf{R} = \mathbf{C}_2^{(\mathbf{x})}\mathbf{R}\mathbf{\Lambda}, \tag{7.1}$$

where $\mathbf{R}$ is the unmixing matrix, subject to learning, $\mathbf{C}_2^{(\mathbf{x})}$ is the covariance matrix as defined in 2.19, and $\mathbf{C}_2^{(\mathbf{x})}(\tau)$ is the time delayed correlation matrix of the mixed signal $\mathbf{x}(t)$ defined similar to the covariance

matrix by

$$\mathbf{C}^{(\mathbf{x})}(\tau) := \left\langle \mathbf{x}(t)\mathbf{x}(t+\tau)^T \right\rangle, \tag{7.2}$$

$$C_{ij}^{(\mathbf{x})}(\tau) := \left\langle x_i(t)x_j(t+\tau) \right\rangle. \tag{7.3}$$

with τ being the time delay between two signal components. Equation (7.1) can be solved by standard numerical linear algebra. Molgedey and Schuster also proposed a simple neural network (see Fig.7.1) that is able to solve this kind of blind source separation.

If there exist degenerate eigenvalues in Equation (7.1) the method will fail to extract all source signal components. Nevertheless, Tong et al. [1991] have shown that under some assumptions there exists a τ such that the source signal components can be recovered.

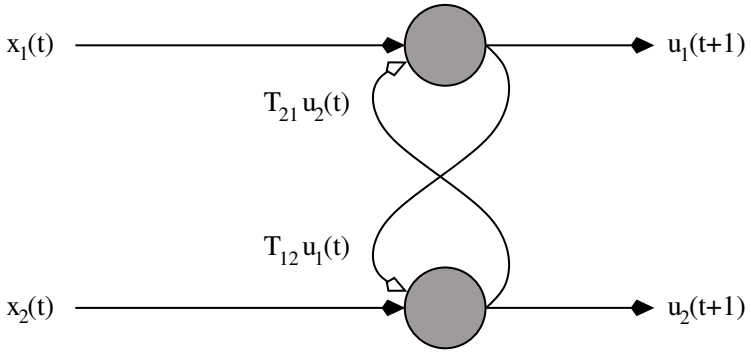


Figure 7.1: Neural network with lateral inhibition that solves the source separation problem for a linear mixture of two source signal components. Inhibitory synapses are shown as open arrows, excitatory synapses are shown as filled arrows. The output of the network can be described by $\mathbf{u}(t+1) = -\mathbf{T}\mathbf{u}(t) + \mathbf{x}(t)$. The weights T_{12} and T_{21} define the strength of the inhibitory synapses. The diagonal elements of the weight matrix are zero ($T_{11} = T_{22} = 0$). Matrix $\mathbf{T}$ is related to the unmixing matrix $\mathbf{R}$ via $\mathbf{R} = \mathbf{I} + \mathbf{T}$, with identity matrix $\mathbf{I}$.

7.2 CuBICA2 with a Single Time Delay

Molgedey and Schuster introduced time delayed correlations in order to achieve non symmetric unmixing matrices. If we consider the alternative measure of independence described in Section 4.4 signal components are considered statistically independent if they have vanishing time-delayed cross-correlations. That means, that in the ideal case all time-delayed correlation-matrices are diagonal. Here the symmetrized version of a cross correlation matrix is used because the non symmetric matrices can have complex eigenvalues and eigenvectors, which can cause problems during diagonalization (see e.g. [Ziehe and Müller, 1998] where the objective function is not shown in the paper but the Matlab implementation made available by the authors makes use of this symmetric form). The symmetrized form of (7.1) reads

$$\mathbf{C}^{(\mathbf{x})}(\tau) = \frac{1}{2}\left\langle \mathbf{x}(t)\mathbf{x}(t+\tau)^T + \mathbf{x}(t+\tau)\mathbf{x}(t)^T \right\rangle \tag{7.4}$$

$$C_{ij}^{(\mathbf{x})}(\tau) = \frac{1}{2}\left\langle x_i(t)x_j(t+\tau) + x_i(t+\tau)x_j(t) \right\rangle. \tag{7.5}$$

Eigenvalue problem (7.1) can be reformulated similar to that in the previous chapter which also allows a straightforward extension to the use of several time lags. First divide the unmixing matrix according to Section 4.3.1 and as done in the previous chapter into two parts, namely $\mathbf{R} = \mathbf{Q}\mathbf{W}$. $\mathbf{W}$ denotes a whitening transformation and $\mathbf{Q}$ is an orthogonal transformation.

Now, assume that the input signal $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ is whitened. Then the generalized eigenvalue problem (7.1) reduces to a normal one

$$\mathbf{C}_2^{(\mathbf{y})}(\tau)\,\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}, \tag{7.6}$$

with remaining orthogonal transformation $\mathbf{Q}$. From Equation (7.6) follows

$$\mathbf{Q}^T\mathbf{C}_2^{(\mathbf{y})}(\tau)\,\mathbf{Q} = \mathbf{C}_2^{(\mathbf{u})}(\tau) = \mathbf{\Lambda}. \tag{7.7}$$

Thus $\mathbf{Q}$ diagonalizes $\mathbf{C}_2^{(\mathbf{u})}(\tau)$, Therefore the eigenvalue problem can be described as an diagonalization problem. Using the definition for matrix-diagonalization from Section 4.6.2 we can define a simple objective function

$$\Psi_2 = \sum_{\substack{i,j=1\\ i\neq j}}^{N} \left(C_{ij}^{(\mathbf{u})}(\tau)\right)^2 \tag{7.8}$$

$$= \sum_{\substack{i,i=1\\ i\neq j}}^{N} \left(\mathbf{q}_i^T\mathbf{C}^{(\mathbf{y})}(\tau)\,\mathbf{q}_j\right)^2, \tag{7.9}$$

where $\mathbf{q}_i := [Q_{i1}, Q_{i2}, \ldots, Q_{iN}]^T$ is the i-th row of $\mathbf{Q}$. Subscript 2 stands for second-order ICA. This objective function can be maximized by applying successive Givens rotations similar to the optimization procedure for CuBICA34. In Section 4.6.2 we have derived a closed expression for the rotation angle ϕ of a single Givens rotation in the case of $N = 2$. We can use (4.37) and solve the optimization problem by minimizing

$$\Psi_2 = A_{0\tau} + A_{4\tau}\cos(4\phi + \phi_{4\tau}), \tag{7.10}$$

with the constants as defined in Appendix B.2.1. A τ in the subscripts of $A_{0\tau}$, $A_{4\tau}$ and $\phi_{4\tau}$ show their dependency on the time delay τ. Using the same argument as in Section 6.1.2 the minimization of the sum of the squared off-diagonal elements of the time-delayed correlation-matrix as in (7.9) is equivalent to the maximization of the sum of the squared time-delayed auto-correlations. This leads to an objective function equivalent to (7.10) but with different constants. The minimum of (7.10) is given by $\phi_{\min} = -\phi 4/4$. Thus, reformulating the eigenvalue problem (7.6) as a diagonalization problem we obtain an objective function that is easy to evaluate and the optimal rotation angle can be calculated in a very simple way. If the number of mixed source signal components N is $N > 2$ the unmixing can be achieved resorting to the method proposed in 6.1.4.

7.3 CuBICA2 with Several Time Delays

From [Tong et al., 1991] we know that second-order ICA can always be solved with a single time delay. However, the delay τ has to be chosen properly so that all eigenvalues of $\mathbf{C}^{(\mathbf{y})}(\tau)$ are distinct. To obtain a more robust method one can consider a certain number T of time-delayed correlation-matrices with respective time delays $\tau = 1, 2, \ldots, T$ and diagonalize them jointly. Belouchrani et al. [1997] derived an algorithm for blind source separation based on simultaneous diagonalization of several time delayed correlation matrices. Ziehe and Müller [1998] proposed a similar algorithm as an extension to the original approach by Molgedey and Schuster. Extending the objective (7.9) to several time delays is straightforward

$$\Psi_{2\mathrm{j}} := \sum_{\tau=1}^{T} \kappa_\tau \Psi_2(\tau) \tag{7.11}$$

$$= \sum_{\tau=1}^{T} \kappa_\tau \sum_{\substack{i,j=1\\ i\neq j}}^{N} \left(\mathbf{C}_{ij}^{(\mathbf{u})}(\tau)\right)^2 \tag{7.12}$$

$$= \sum_{\tau=1}^{T} \kappa_\tau \sum_{\substack{i,j=1\\ i\neq j}}^{N} \left(\mathbf{q}_i^T\mathbf{C}^{(\mathbf{y})}(\tau)\,\mathbf{q}_i\right)^2, \tag{7.13}$$

where we additionally introduced factors κ_τ, which allows us to weight correlation matrices with different time delays differently. Again time dependencies are denoted by a τ as a subscript. The j in 2j stands for *joint* diagonalization. Note that κ_τ should all be positive if Ψ_{2j} is minimized.

Extending the objective function of ICA in this way yields a joint diagonalization of several correlation matrices with different time delays, thus decorrelation is achieved over a small time window of length T. It is intuitively clear that by enlarging the window length the unmixing performance should improve.

Equation (7.10) described the objective function for second-order ICA for a single time delay τ and $N = 2$. For each arbitrary time delay there will be a similar objective function, distinct only by its constants. If we put (7.10) into the first line of Equation (7.13) and denote the explicit τ-dependencies of the constants we arrive at

$$\Psi_{2j} = \sum_{\tau=1}^{T} [A_{0\tau} + A_{4\tau}\cos(4\phi + \phi_{4\tau})] \tag{7.14}$$

$$= \sum_{\tau=1}^{T} A_{0\tau} + \sum_{\tau=1}^{T} A_{4\tau}\cos(4\phi + \phi_{4\tau}) . \tag{7.15}$$

We can simplify objective function (7.15) using some trigonometric identities and derive a very simple objective function, subject to minimization

$$\Psi_{2j}(\phi) = \bar{A}_{0\tau} + \bar{A}_{4\tau}\cos\left(4\phi + \bar{\phi}_{4\tau}\right) , \tag{7.16}$$

with constant as defined in Appendix B.2.2. The minimum of Ψ_{2j} is reached for angles ϕ_{min} satisfying the condition

$$\phi_{min} = \pi/4 - \bar{\phi}_{4\tau}/4 . \tag{7.17}$$

In contrast, the calculation of the rotation angle in TDSEP [Ziehe and Müller, 1998] and SOBI [Belouchrani et al., 1997] involves a two-dimensional eigenvalue problem plus taking a square-root [Cardoso and Souloumiac, 1996].

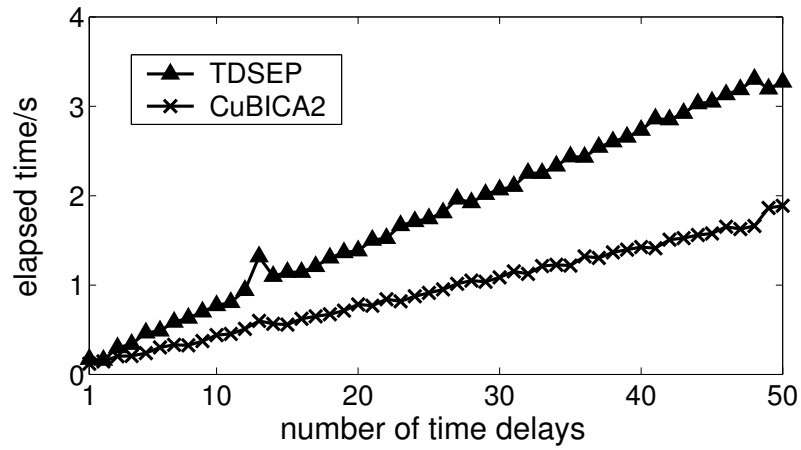
7.4 Simulation

We carried out a simulation to compare CuBICA2 with TDSEP [Ziehe and Müller, 1998], a different ICA method based on the same principle. As a source signal we used dataset (i) from Chapter 6. For TDSEP we have used original software provided by the authors [1]. The data set was mixed by a randomly chosen mixing matrix with entries chosen uniformly from $[-1, 1]$. We used different numbers of time delays T, namely all $1 \leq T \leq 50$. The unmixing performance was measured by the measure introduced in Section 4.7. The results are shown in Figure 7.2. There is virtually no difference between the two algorithms with respect to the unmixing error. Comparing the two by means of the elapsed CPU-time the simulation showed that CuBICA2 is faster than TDSEP by a factor of two. We also tested an implementation of TDSEP with similar routines to that of CuBICA2 which showed no difference in time consumption.

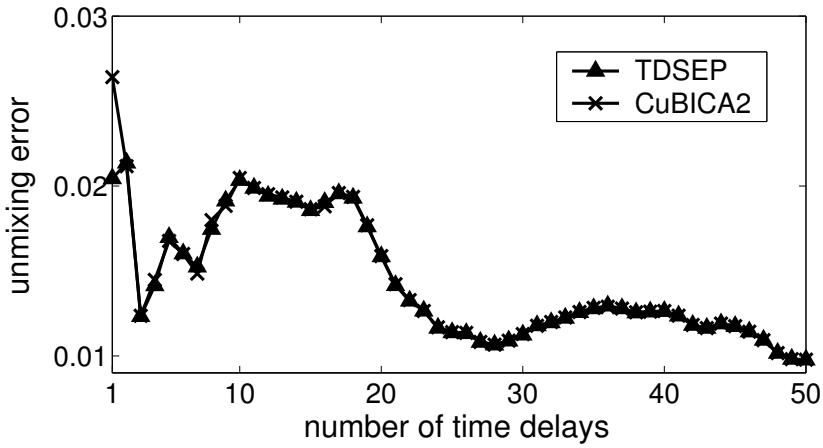
7.5 Summary

CuBICA2, a method for independent component analysis based solely on second-order statistics, has been proposed. In contrast to TDSEP [Ziehe and Müller, 1998], an algorithm based on the same principle, the calculation of the rotation angle for each Givens rotation can be computed in a linear fashion. Comparing CuBICA2 and TDSEP shows that, while both algorithms show identical unmixing performance, CuBICA2 is a factor of two faster than TDSEP. The elegant way of computing the rotation angle gives the possibility to easily develop ICA algorithms that combine second-order and higher-order statistics. Such combined

[1] TDSEP: `http://wwwold.first.fhg.de/~ziehe/download.html` (Version 2.01 of February 14th, 1999, downloaded August 6th, 2004)



(a)



(b)

Figure 7.2: Comparison of TDSEP and CuBICA in terms of **(a)** elapsed time and **(b)** unmixing performance of the respective algorithm. Different numbers of time delays T have been tested ($1 \leq T \leq 50$). TDSEP and CuBICA2 perform virtually identically with respect to the unmixing error, however CuBICA2 is faster by a factor of two. The difference in the performance for $T = 1$ is due to the fact that in this case TDSEP solves the ICA problem via the generalized eigenvalue problem. This method shows better performance than the Jacobi method.

algorithms can be helpful in cases, where some components of the source signal have no auto-correlation and others lack higher-order statistics. A work in this direction is [Müller et al., 1999].

Furthermore CuBICA2 allows us to derive an algorithm for nonlinear blind source separation in combination with SFA, which we will describe in the next part of this thesis.

Part III

Nonlinear Blind Source Separation and Slow Feature Analysis

8

Relations between ICA and SFA in the linear Case

In data analysis when dealing with vectorial signals it is often useful to find a suitable representation to gain as much information as possible about the underlying processes. For example, consider two people speaking simultaneously while being recorded with two microphones. The observed signal is a mixture of their voices and a useful representation would be one where each signal component contains only information of a single speaker. For visual input data, for instance, one might be interested in a representation that is invariant to typical transformations, such as translation or zoom. A variety of linear and nonlinear methods are known, depending on the task, to extract the interesting features from an observed signal.

In the first and second part of this thesis we have introduced the concepts of BSS/ICA and slow feature analysis (SFA), two possible methods that are able to extract some of the interesting features from an observed signal. ICA finds a representation of the data such that signal components are mutually statistically independent, which can be used to separate the two speakers in the example above. SFA extracts slowly varying features, which can be used in the second example for learning visual invariances. At first glance these two perspectives to analyze multivariate signals are very different and actually seem to be conflicting, since two slowly varying signals of finite length are more likely to have statistical dependencies than quickly varying ones. But as we will see ICA, and SFA do have common properties which we want to point out by comparing the two algorithms mathematically.

SFA is constrained to signals with temporal structure like speech signals and it is based on second-order statistics. For a comparison with ICA we therefore consider the ICA algorithm introduced in Chapter 7 that only uses second order information and also needs a temporally structured signal.

SFA is generally a nonlinear method. It uses a nonlinear expansion to map the input signal into a feature space and solves the linear problem in the feature space, whereas ICA is typically a linear method (although there exist some nonlinear approaches). To make a comparison between the two methods possible, we will restrict SFA to the linear case. Nevertheless all calculations in the following are essentially the same for linear or nonlinear SFA.

Throughout this chapter we will follow the two-stage approach (cf. Sec. 4.3.1). After the whitening-stage, the first stage, (see Sec. 4.3.1) an orthogonal transformation $\mathbf{Q}$ on $\mathbf{y}$, which mainly corresponds to a rotation, is sufficient to yield independent components [Comon, 1994b] or slowly varying features (cf. Sec. 5.1.2). Thus the output signal $\mathbf{u}(t)$ can be obtained by combining the whitening matrix $\mathbf{W}$ and a rotation matrix $\mathbf{Q}$

$$\mathbf{u}(t) = \mathbf{Q}\mathbf{y}(t) = \mathbf{Q}\mathbf{W}\mathbf{x}(t) \,. \tag{8.1}$$

In the following we will always assume whitened data $\mathbf{y}(t)$ and focus on finding $\mathbf{Q}$ for unmixing. Notice, that for whitened data properties zero mean (4.10), unit variance (4.11), and decorrelation (4.12) constraints

hold. Since these properties are preserved under any orthogonal transformation it is obvious that the components of $\mathbf{u}(t)$ also satisfy these conditions (5.2-5.4) imposed by SFA. For ICA, these properties are essential too, since they force the $y_i(t)$ to be statistically independent in first, and second order (cf. Sec. 4.3.1).

In order to compare SFA and BSS/ICA we recapitulate the objective function of CuBICA2 with a single time delay τ, subject to minimization (7.9)

$$\Psi_2 = \sum_{\substack{i,j=1 \\ i\neq j}}^{N} \left(C_{ij}^{(\mathbf{u})}(\tau)\right)^2 \tag{8.2}$$

$$= \sum_{\substack{i,j=1 \\ i\neq j}}^{N} \left(\mathbf{q}_i^T \mathbf{C}^{(\mathbf{y})}(\tau)\,\mathbf{q}_j\right)^2, \tag{8.3}$$

where $\mathbf{q}_i = [Q_{i1}, Q_{i2}, \ldots, Q_{iN}]^T$ is the i-th row of $\mathbf{Q}$. Ψ_2 is a function of the $\mathbf{q}_i$ which are subject to learning and the whitened signal components $\mathbf{y}$ are given. Keeping the ICA objective (8.3) in mind we will now derive an SFA objective function in the following, that allows a comparison between SFA and second-order ICA.

In Section 8.1 we derive similarities between ICA and SFA first in the case of a single time delay. Possible extensions to several time delays are derived in Section 8.2. The chapter concludes with a comparison in Section 8.3 and a summary in Section 8.4.

8.1 Linear Slow Feature Analysis

Assume a whitened input signal $\mathbf{y}(t) = [y_1(t), \ldots, y_N(t)]^T$ is given. Linear SFA finds a rotation matrix $\mathbf{Q}$ such that the components u_i of the output signal $\mathbf{u}(t) = \mathbf{Q}\mathbf{y}(t)$ are varying as slowly as possible, the first one being the slowest possible, the second one the next slowest uncorrelated to the first, etc. As a measure of slowness we define (small values indicating slowly varying signals) (cf. 5.1)

$$\Delta(u_i) := \left\langle (\dot{u}_i(t))^2 \right\rangle. \tag{8.4}$$

Searching for the slowest components we want to minimize $\Delta(u_i)$. Because of the prewhitening and since $\mathbf{Q}$ is orthogonal, $u_i(t)$ has zero mean and unit variance. This ensures that the solution will not be the trivial solution $u_i(t) = \text{const}$. Decorrelation of the signal components, the third property due to the prewhitening, guarantees that different components carry different information (see also Sec. 5.1).

We will first show how to solve the optimization problem of SFA in a way similar to that described in Chapter 5 and then establish a mathematical link between SFA and ICA in a way inspired by second-order ICA.

Since we have discrete time series, we need an approximation of $\dot{\mathbf{u}}(t)$. As a first order approximation of the first derivative of $\mathbf{u}(t)$ we define

$$\mathbf{u}(t) \approx \mathbf{u}(t+1) - \mathbf{u}(t). \tag{8.5}$$

With this approximation we can compute $\left\langle \dot{u}_i^2 \right\rangle$ as

$$\left\langle \dot{u}_i^2 \right\rangle \approx \left\langle [u_i(t+1) - u_i(t)]\,[u_i(t+1) - u_i(t)] \right\rangle \tag{8.6}$$

$$= [\langle u_i(t+1)\,u_i(t+1)\rangle + \langle u_i(t)\,u_i(t)\rangle] - [\langle u_i(t)\,u_i(t+1)\rangle + \langle u_i(t+1)\,u_i(t)\rangle] \tag{8.7}$$

$$= 2\langle u_i(t)\,u_i(t)\rangle - \langle u_i(t)\,u_i(t+1) + u_i(t+1)\,u_i(t)\rangle \tag{8.8}$$

$$(\langle u_i(t+1)\,u_i(t+1)\rangle = \langle u_i(t)\,u_i(t)\rangle \text{ because averaging is over all t})$$

$$= 2 - \langle u_i(t)\,u_i(t+1) + u_i(t+1)\,u_i(t)\rangle \tag{8.9}$$

$$(\langle u_i(t)\,u_i(t)\rangle = 1 \quad \text{because } \mathbf{u}(t) \text{ is white}).$$

Since the constant factor does not matter during optimization we can make a simplification. Instead of minimizing $\Delta(u_i)$, we can maximize

$$\begin{aligned}\widetilde{\Delta}(u_i) &:= 1-\frac{1}{2}\Delta(u_i) && (8.10)\\ &= \frac{1}{2}\left\langle u_i(t)\,u_i(t+1)+u_i(t+1)\,u_i(t)\right\rangle && (8.11)\\ &= C_{ii}^{(\mathbf{u})}(1) && (8.12)\\ &= \mathbf{q}_i^T\,\mathbf{C}^{(\mathbf{y})}(1)\,\mathbf{q}_i\,, && (8.13)\end{aligned}$$

where we notice that $\widetilde{\Delta}(u_i)$ is a function of the orthogonal $\mathbf{Q}$ and thus we are searching for the normed weight vectors $\mathbf{q}_i$ that maximize $\widetilde{\Delta}(u_i)$ in (8.13). The solution for $i=1$ is the eigenvector of the time-delayed correlation-matrix $\mathbf{C}^{(\mathbf{y})}(1)$ that belongs to the largest eigenvalue. This eigenvector produces the slowest component $u_1(t)$. The eigenvectors of the next higher eigenvalues produce the next slowest components ($u_2(t), u_3(t), \ldots$ and so forth).

Thus, to extract all slow components the maximization problem (8.13) can be formulated as an eigenvalue problem (cf. (5.1.2))

$$\mathbf{C}^{(\mathbf{y})}(1)\,\mathbf{Q}=\mathbf{Q}\mathbf{\Lambda} \tag{8.14}$$

where $\mathbf{\Lambda}$ denotes a diagonal matrix with Λ_{ii} the ith eigenvalue belonging to the eigenvector $\mathbf{q}_i$. Solving the eigenvalue problem (8.14) yields the eigenvalues and eigenvectors of $\mathbf{C}^{(\mathbf{y})}(1)$ but without the preferred order. Therefore the eigenvectors must be sorted by their corresponding eigenvalues in decreasing order. In this way the extracted signal components $u_i(t)$ are arranged according to slowness with u_1 being the slowest component.

In order to allow a better comparison with second-order ICA, we now want to deduce an alternative formulation of SFA, i.e. we want to construct an objective function similar to that of second-order ICA. First we show the equivalence of solving the eigenvalue problem (8.14) and the diagonalization of $\mathbf{C}^{(\mathbf{u})}(1)$. If we multiply both sides of (8.14) with $\mathbf{Q}^T$ we obtain

$$\mathbf{Q}^T\mathbf{C}^{(\mathbf{y})}(1)\,\mathbf{Q}=\frac{1}{2}\left\langle \mathbf{u}(t)\,\mathbf{u}(t+1)^T+\mathbf{u}(t+1)\,\mathbf{u}(t)^T\right\rangle=\mathbf{C}^{(\mathbf{u})}(1)=\mathbf{\Lambda}. \tag{8.15}$$

Since $\mathbf{\Lambda}$ is diagonal, $\mathbf{C}^{(\mathbf{u})}(1)$ is diagonal, too. Therefore solving the eigenvalue problem is equivalent to finding a matrix $\mathbf{Q}$ with $\mathbf{u}(t)=\mathbf{Q}\mathbf{y}(t)$ such that the time-delayed correlation-matrix $\mathbf{C}^{(\mathbf{u})}(1)$ is diagonal (cf. [Wiskott, 2003b]). To perform the diagonalization we use the same Jacobi scheme as for second-order ICA, namely we minimize all off-diagonal entries of $\mathbf{C}^{(\mathbf{u})}(1)$ (see Sec. 7.2) and derive the following objective function for SFA to be minimized

$$\widetilde{\Psi}_{SFA} := \sum_{\substack{i,j=1\\ i\neq j}}^{N}\left(\mathbf{C}_{ij}^{(\mathbf{u})}(1)\right)^2. \tag{8.16}$$

Interestingly this objective function is identical to the one for second-order ICA (8.3). Thus we arrive at the important result that in the linear case, second-order ICA and SFA are equivalent in the case of one time delay.

To bring (8.16) into a form that can be understood more intuitively in the sense of SFA we can use the fact that the sum of all squared entries of correlation matrices with a given time delay τ is invariant under orthogonal transformations

$$\sum_{i,j=1}^{N}\left(\mathbf{C}_{ij}^{(\mathbf{u})}(\tau)\right)^2=\sum_{i,j=1}^{N}\left(\mathbf{C}_{ij}^{(\mathbf{y})}(\tau)\right)^2=\text{const.} \tag{8.17}$$

We can split this sum into two terms

$$\sum_{i,j=1}^{N}\left(\mathbf{C}_{ij}^{(\mathbf{u})}(\tau)\right)^2 = \sum_{i=1}^{N}\left(\mathbf{C}_{ii}^{(\mathbf{u})}(\tau)\right)^2+\sum_{\substack{i,j=1\\ i\neq j}}^{N}\left(\mathbf{C}_{ij}^{(\mathbf{u})}(\tau)\right)^2=\text{const.}, \tag{8.18}$$

so that it is easy to see that minimization of $\widetilde{\Psi}_{SFA}$ is equivalent to maximization of

$$\Psi_{SFA} := \sum_{i=1}^{N} \left(\mathbf{C}_{ii}^{(\mathbf{u})}(1) \right)^2 \tag{8.19}$$

$$= \sum_{i=1}^{N} \left(\mathbf{q}_i^T \mathbf{C}^{(\mathbf{y})}(1) \mathbf{q}_i \right)^2 . \tag{8.20}$$

Thus, having started from minimizing temporal variations (8.4) as an objective for SFA we now conclude with an objective for maximizing auto-correlations (8.20). This relation can be interpreted intuitively: a signal component with large auto-correlation has a high temporal predictability. Predictability on the other hand means that the signal component has to vary slowly.

Maximizing (8.20) produces the same slow components $u_1(t), \ldots, u_N(t)$ as obtained by the eigenvalue problem (8.14), again not ordered by slowness. Thus, if an order of the slow components is needed, a further sorting step has to be applied to the components when using the Jacobi method (cf. 4.6).

What if $C_{ii}^{(\mathbf{u})}(1) < 0$? This could happen if for example $u_i(t)$ has alternating signs for successive data points, e.g. define a signal component by

$$u_i(t) := \begin{cases} -1 & \text{for } t \text{ odd} \\ 1 & \text{for } t \text{ even} \end{cases} , \tag{8.21}$$

with $1 \leq t \leq P$. This signal component has zero mean and unit variance and thus fulfills Constraints (5.2) and (5.3). Furthermore it is favourable in terms of the objective (8.20), since $C_{ii}^{(\mathbf{u})}(1)$ has a large absolute value. But it is a very fast varying component which seems paradoxical, since maximizing (8.20) should result in slowly varying components. This apparent contradiction can be resolved by studying the constraints imposed on the optimization of (8.20). Since $\mathbf{Q}$ is an orthogonal matrix, the trace of $\mathbf{C}^{(\mathbf{u})}(1)$ is invariant under the transformation $\mathbf{u}(t) = \mathbf{Q}\mathbf{y}(t)$ [Zurmühl and Falk, 1997]. Thus, if we consider all N possible components in the optimization procedure, the decrease of one correlation $C_{ii}^{(\mathbf{u})}(1)$ implies that at least one other correlation $C_{jj}^{(\mathbf{u})}(1)$ with $j \neq i$ is increased. Therefore extracting quickly varying components implies that other extracted components are slowly varying signals. Hence, it is reasonable to further minimize negative correlations since this in turn implies that other correlations will be maximized.

8.2 More than one Time Delay

8.2.1 Second-Order ICA

A straightforward extension of objective (8.3) to several time delays subject to minimization is given by (7.13). We give the definition

$$\Psi_{2\mathrm{j}} := \sum_{\tau=1}^{T} \kappa_\tau \Psi_2(\tau) \tag{8.22}$$

$$= \sum_{\tau=1}^{T} \kappa_\tau \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \left(\mathbf{C}_{ij}^{(\mathbf{u})}(\tau) \right)^2 \tag{8.23}$$

$$= \sum_{\tau=1}^{T} \kappa_\tau \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \left(\mathbf{q}_i^T \mathbf{C}^{(\mathbf{y})}(\tau) \mathbf{q}_j \right)^2 , \tag{8.24}$$

with κ_τ that allow to weight correlation matrices with different time delays differently. Note that κ_τ should all be positive if $\Psi_{2\mathrm{j}}$ is minimized.

8.2.2 SFA

Joint Diagonalization

To extend SFA to more than a single time delay, we can use a similar argument as for second-order ICA. Adding more time-delayed auto-correlations increases the temporal predictability of the signal, that is, knowing the amplitude of a signal at a given time point, we can give a good prediction of the next T time points since they are strongly correlated. Signals with large temporal predictability in turn are likely to be slowly varying. Thus, an intuitive extension of the normal SFA objective (8.20), subject to maximization, is

$$\Psi_{\text{SFAj}} := \sum_{\tau=1}^{T} \kappa_\tau \Psi_{\text{SFA}}(\tau) \tag{8.25}$$

$$= \sum_{\tau=1}^{T} \kappa_\tau \sum_{i=1}^{N} \left(\mathbf{C}_{ii}^{(\mathbf{u})}(\tau) \right)^2 . \tag{8.26}$$

$$= \sum_{\tau=1}^{T} \kappa_\tau \sum_{i=1}^{N} \left(\mathbf{q}_i^T \mathbf{C}^{(\mathbf{y})}(\tau) \mathbf{q}_i \right)^2 . \tag{8.27}$$

Again, the j in SFAj stands for *joint* diagonalization. Like in (8.24) we introduced weighting factors κ_τ for each time-delayed correlation-matrix with time delay τ. Note that this is again equivalent to the ICA objective (8.24) due to the constancy of the sum of all squared entries of each time-delayed correlation-matrix (8.18).

We have to be careful with this definition, though, for two reasons.

Firstly, while the definition of slowness based on $C_{ii}^{(\mathbf{u})}(1)$ corresponds to our intuition of what a slow signal is, $C_{ii}^{(\mathbf{u})}(2)$ can have a large value for signal components that we would not consider to be slow at all. In fact, the alternating signal defined at the end of Section 8.1 would yield a maximal value of $C_{ii}^{(\mathbf{u})}(2)$.

Secondly, consider the case where two time-delayed auto-correlations have different sign, e.g. $C_{ii}^{(\mathbf{u})}(1) < 0$ and $C_{ii}^{(\mathbf{u})}(2) > 0$. Maximizing objective function (8.26) would favor a decreasing value of $C_{ii}^{(\mathbf{u})}(1)$ (since it is negative) and an increasing value of $C_{ii}^{(\mathbf{u})}(2)$. The former would intuitively tend to make the signal faster while the latter would make it slower. Thus, if the auto-correlations of a component have different signs for different time-delays, the objective function seems to be inconsistent, at least for this component. This conflict cannot be solved as easily as the conflict discussed at the end of Section 8.1. However, one can at least monitor the signs of the auto-correlations and diagnose the inconsistent cases.

It is not clear to us to what extent these two problems matter in practice. We believe that by weighting the first auto-correlation stronger than the other ones, e.g. with an exponential decay of the weights, the inconsistencies can be largely avoided.

Linear Filtering

An alternative to the joint diagonalization of several correlation matrices with different time delays, as is done also for second-order ICA, is to average over a range of time delays within one correlation matrix and

diagonalize just this one matrix. To do so, we introduce the following new measure of slowness:

$$\begin{aligned}
\Sigma(u_i) &:= \frac{1}{2}\left\langle u_i(t)\left(\sum_{\tau=1}^{T}\kappa_\tau u_i(t+\tau)\right)+\left(\sum_{\tau=1}^{T}\kappa_\tau u_i(t+\tau)\right)u_i(t)\right\rangle && (8.28)\\
&= \sum_{\tau=1}^{T}\kappa_\tau\frac{1}{2}\left\langle u_i(t)\,u_i(t+\tau)+u_i(t+\tau)\,u_i(t)\right\rangle && (8.29)\\
&= \sum_{\tau=1}^{T}\kappa_\tau C_{ii}^{(\mathbf{u})}(\tau) && (8.30)\\
&= \mathbf{q}_i^T\left(\sum_{\tau=1}^{T}\kappa_\tau\mathbf{C}^{(\mathbf{y})}(\tau)\right)\mathbf{q}_i && (8.31)\\
&=: \mathbf{q}_i^T\widetilde{\mathbf{C}}^{(\mathbf{y})}\mathbf{q}_i\,, && (8.32)
\end{aligned}$$

with constants κ_τ that weight different time delays differently. This definition (8.28) differs from (8.11) in that $u_i(t)$ should not only be well correlated to the next data point but to a weighted average over the next T data points. This is a straightforward way of taking several time scales into account. Note that the weighted averaging is a linear-filter operation. Like in the joint-diagonalization extension exponentially decaying weights $\kappa_\tau := \exp(-\gamma\tau)$ for different time delays seems to be a suitable choice. With such weights this measure of slowness is similar to the objective of temporal smoothness used by Stone [1995] and somewhat related also to the trace learning rules first introduced by Földiák [1991].

Because of the formal similarity of (8.32) with (8.13) we can apply the analogous steps that led from (8.13) to (8.20) and derive the following objective function to be maximized

$$\begin{aligned}
\Psi_{\mathrm{SFAl}} &:= \sum_{i=1}^{N}\left(\widetilde{C}_{ii}^{(\mathbf{u})}\right)^2 && (8.33)\\
&= \sum_{i=1}^{N}\left(\mathbf{q}_i^T\widetilde{\mathbf{C}}^{(\mathbf{y})}\mathbf{q}_i\right)^2, && (8.34)
\end{aligned}$$

where SFAl stands for linear-filtering SFA. Since this objective function is based on just one correlation matrix, it does not have the problems mentioned above for the joint-diagonalization extension.

Higher Derivatives

We have also considered extending SFA by simultaneously minimizing not only the variance of the first but also of higher-order derivatives. This also leads to equation (8.34), because approximations of higher-order derivatives involve multiple time delays.

First, define an objective function using the nth derivative instead of the first derivative used in (8.4) as

$$\Delta_n(u_i) := \left\langle\left(u_i^{(n)}\right)^2\right\rangle, \tag{8.35}$$

where $u_i^{(n)}$ denotes the nth derivative of u_i. We can now define an extended objective function for SFA including several derivatives, which is subject to minimization

$$\Sigma(u_i) := \sum_{n=0}^{T}\alpha_n\Delta_n(u_i) = \sum_{n=0}^{T}\alpha_n\left\langle\left(u_i^{(n)}\right)^2\right\rangle, \tag{8.36}$$

where T is the highest derivative taken into account, and α_n are factors that weight the different derivatives. Corresponding to the approximated first derivative of u_i (8.5) we can also define appropriate approximations

for all other derivatives $u_i^{(n)}$ (see Appendix C.1). Using these we can write the approximated objective function, subject to minimization

$$\Sigma(u_i) \approx \sum_{n=0}^{T} \alpha_n \sum_{\tau=0}^{n} \beta_{n\tau} C_{ii}^{(\mathbf{u})}(\tau) \tag{8.37}$$

$$= \sum_{\tau=0}^{T} \delta_\tau C_{ii}^{(\mathbf{u})}(\tau)\,. \tag{8.38}$$

Constants $\beta_{n\tau}$ are due to the approximated derivatives (see Appendix C.2) and listed in Table 8.0(a) up to order four. From Table 8.0(a) we can identify two interesting properties of the $\beta_{n\tau}$:

(i) $\beta_{n\tau}$ are larger for larger n.

(ii) $\beta_{n\tau}$ have alternating signs for successive time delays.

Due to property (i) higher derivatives would dominate the objective function (8.38). Therefore we introduced the additional constants α_n which can compensate for this imbalance. Combining the $\beta_{n\tau}$ and α_n results in constants δ_τ (see Appendix C.3).

Since the correlation matrix with zero time delay is constant under orthogonal transformations we can neglect it and derive, similar to (8.11), an objective function, subject to maximization

$$\tilde{\Sigma}(u_i) := -\sum_{\tau=1}^{T} \delta_\tau C_{ii}^{(\mathbf{u})}(\tau) \tag{8.39}$$

$$= \mathbf{q}_i^T \left(-\sum_{\tau=1}^{T} \delta_\tau \mathbf{C}^{(\mathbf{y})}(\tau) \right) \mathbf{q}_i \tag{8.40}$$

$$=: \mathbf{q}_i^T \overline{\mathbf{C}}^{(\mathbf{y})} \mathbf{q}_i\,. \tag{8.41}$$

Equation (8.41) is formally similar to (8.13). Thus, we can apply the analogous steps that led from (8.13) to (8.20) and derive the following objective function, subject to maximization

$$\Psi_{\text{SFAh}} := \sum_{i=1}^{N} \left(-\sum_{\tau=1}^{T} \delta_\tau C_{ii}^{(\mathbf{u})}(\tau) \right)^2 \tag{8.42}$$

$$= \sum_{i=1}^{N} \left(\overline{C}_{ii}^{(\mathbf{u})} \right)^2 \tag{8.43}$$

$$= \sum_{i=1}^{N} \left(\mathbf{q}_i^T \overline{\mathbf{C}}^{(\mathbf{y})} \mathbf{q}_i \right)^2, \tag{8.44}$$

where SFAh stands for ***h**igher derivatives*.

The constants δ_τ in (8.42) depend on the choice of α_n (cf. Appendix C.3). Setting

$$\alpha_n = \frac{1}{\sum_{\tau=0}^{n} |\beta_{n\tau}|} \quad \forall\, n \in \{0, 1, \ldots, T\} \tag{8.45}$$

results in weighting constants δ_τ shown in Table 8.0(b). It can be seen that all δ_τ used in the objective function (8.42) (those with $1 \leq \tau \leq T$) decrease almost exponentially in τ for a given T.

The weights α_n of different derivatives need to be positive, otherwise some of the $\Delta_n(u_i)$ will be minimized and some will be maximized. Therefore the constants δ_τ share property (ii) with $\beta_{n\tau}$. Thus, by maximizing Ψ_{SFAh} of Equation (8.44) a sum of time-delayed auto-correlations $C_{ii}^{(\mathbf{u})}(\tau)$ with positive δ_τ (τ odd) and negative δ_τ (τ even) will be optimized (one could only force the δ_τ to have the same sign for all τ by choosing α_n in a counterintuitive way, e.g. in the case of $T = 4$ using $[\alpha_1, \alpha_2, \alpha_3, \alpha_4] = [-10, 15, -7, 1]$

(a) Constants $\beta_{n\tau}$

n ↓ τ →	0	1	2	3	4
0	1/2	0	0	0	0
1	1	-1	0	0	0
2	3	-4	1	0	0
3	10	-15	6	-1	0
4	35	-56	28	-8	1

(b) Constants δ_τ

T ↓ τ →	0	1	2	3	4	factor
0	1					$\times 2$
1	3	-1				$\times 1/2$
2	15	-8	1			$\times 1/8$
3	70	-47	10	-1		$\times 1/32$
4	315	-244	68	-12	1	$\times 1/128$

Table 8.1: **(a)** Constants $\beta_{n\tau}$ arising from the linear approximation of the first four derivatives of u_i. Subscript n denotes the order of derivative. **(b)** Constants δ_τ arising from the linear approximation of the first four derivatives of u_i, if all derivatives are weighted corresponding to $\alpha_n = 1/\left(\sum_{\tau=0}^{n} |\beta_{n\tau}|\right) \forall\, n \in \{0, 1, \ldots, T\}$. Additionally we omitted common factors of all constants in a single row, since only the relative weighting of the time-delayed correlations is relevant, they are shown in the last column of each row. All derivatives up to order T are summed up. Note, that the number of time delays equals the order of the highest derivative taken into account. For detailed calculations see Appendix C.2.

all T time-delayed correlation-matrices in (8.44) will be weighted equally and have the same sign ($\delta_\tau = 1$)). This is counterintuitive and hints at a flaw of this approach. Higher derivatives, even though their approximation involves several time delays, do not focus on longer time scales but on the fine structure on shorter time scales, which are estimated based on several successive data points. Conceptually, this seems to be the wrong direction to go, even though unmixing performance was actually good in some simple examples.

8.3 Comparison of SFA and ICA

We have seen, that the objective functions for second-order ICA and linear SFA are identical for a single time delay. Even if SFA and ICA have the same objective function the two algorithms are designed to solve different problems. Therefore they are based on different assumptions of the source signal and the mixture model. It is worth discussing these assumptions needed to perform SFA resp. ICA.

For second-order ICA it can be shown that given a linear mixture $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ of source signal components source separation can be achieved if the source signal components $s_j(t)$ are (i) uncorrelated, and (ii) have different auto-correlations at some time delay $\tau > 0$ (Theorem 2 in [Tong et al., 1991]). SFA needs (i) uncorrelated source signal components and (ii) uncorrelated first derivatives. The decorrelation condition is always fulfilled for both methods since we always apply a whitening step first. To compare the second conditions we again use the linear approximation (8.5) to see that decorrelation of the derivatives of the source signal components is approximately equal to decorrelation of the time-delayed signal-components ($\left\langle s_i(t)\, s_j(t+\tau)\right\rangle = 0$) with the given time delay $\tau = 1$ (8.9). The difference between the two methods is that for SFA the time delay is given and the source signal components must have different auto-correlations at this point, whereas for second-order ICA the right time delay can be chosen properly.

The case of several time delays e.g. TDSEP [Belouchrani et al., 1997], and SOBI [Ziehe and Müller,

1998], which is a way to circumvent the problem in second-order ICA of finding the right time delay, has a correspondence in the joint-diagonalization approach of SFA, called SFAj (8.27).

However since ICA is a linear method the equivalence of SFA and second-order ICA only holds for linear SFA. SFA will produce accurate results, namely the slowest components that can be achieved, no matter if linear or the nonlinear SFA is used whereas ICA can recover only linearly mixed source signal components.

8.4 Summary

To summarize differences and similarities between the second-order ICA and SFA one can state that (i) since ICA is a linear method whereas SFA can be linear and nonlinear a reasonable comparison can only be done with linear SFA, (ii) in the case of a single time delay $\tau = 1$ the two algorithms are equivalent, cf. (8.2) and (8.16); (iii) in case of several time delays a formulation of SFA (8.27) similar to ICA (8.24) can be found, but is known not to work in all cases. Instead a different extension (8.34) is introduced, that provides a proper extension of the slowness criterion (8.11) to several time scales.

9

Independent Slow Feature Analysis

In this chapter we will concentrate on nonlinear blind source separation (BSS) (cf. Sec. 4.5). While the linear BSS problem can be solved by resorting to independent component analysis (ICA) (cf. Sec. 4.2) this is not possible in the nonlinear case.

The objective of this chapter is to show that the nonlinear BSS problem can be solved by combining ICA and SFA. After a short introduction in Section 9.1 we will introduce independent slow feature analysis (ISFA) in Section 9.2, a combination of ICA and SFA, that solves the nonlinear BSS problem. After some simple simulations in Section 9.3, demonstrating the performance of ISFA, we conclude with a brief discussion in Section 9.4.

9.1 A New Approach to Nonlinear BSS

Starting from the nonlinear mixing model (4.27)

$$\mathbf{x}(t) = F(\mathbf{s}(t)), \tag{9.1}$$

with a nonlinear function $F(\cdot)\,\mathbb{R}^N \to \mathbb{R}^M$, we want to extract the original source signal $\mathbf{s}(t)$ when only $\mathbf{x}(t)$ is observed. There are several known methods that try to solve this nonlinear BSS problem (cf. Sec. 4.5). They can roughly be divided into algorithms with a parametric approach (Fig. 9.1 **(a)**) and algorithms with a nonlinear-expansion approach (Fig. 9.1 **(b)**). With the parametric model the nonlinearity of the mixture is estimated by parametrized nonlinearities (see e.g. [Almeida, 2004]). In the nonlinear-expansion approach the observed mixture is mapped into a high dimensional feature space and afterwards a linear method is applied to the expanded data, a common technique to turn a non-linear problem into a linear one.

In our approach we do not follow the parametric approach but adopt a nonlinear-expansion approach. With respect to the source signal components we make the assumption that they have significant autocorrelation, which makes them vary on a relatively slow time scale. Nonlinear mixtures of such signal components are typically more quickly varying than the original components. Assume for example a sinusoidal signal component $x_i(t) = \sin(t)$ and a second component that is the square of the first $x_j(t) = x_i(t)^2 = 0.5\,(1-\cos(2t))$ are given. The second component is more quickly varying due to the frequency doubling induced by the squaring. To extract the right source signal components one should therefore prefer the slowly varying ones. Considering this we propose, in order to perform nonlinear BSS, to complement the independence objective of pure ICA with a slowness objective. The basis of this slowness objective builds SFA as defined in Chapter 5.

In Chapter 8 we developed an alternative objective for SFA showing the direct relation between ICA and SFA. This has been done for the linear case. In the nonlinear case this relation can not be established, but the alternative formulation at least permits a simple integration of the SFA and ICA objectives. Here

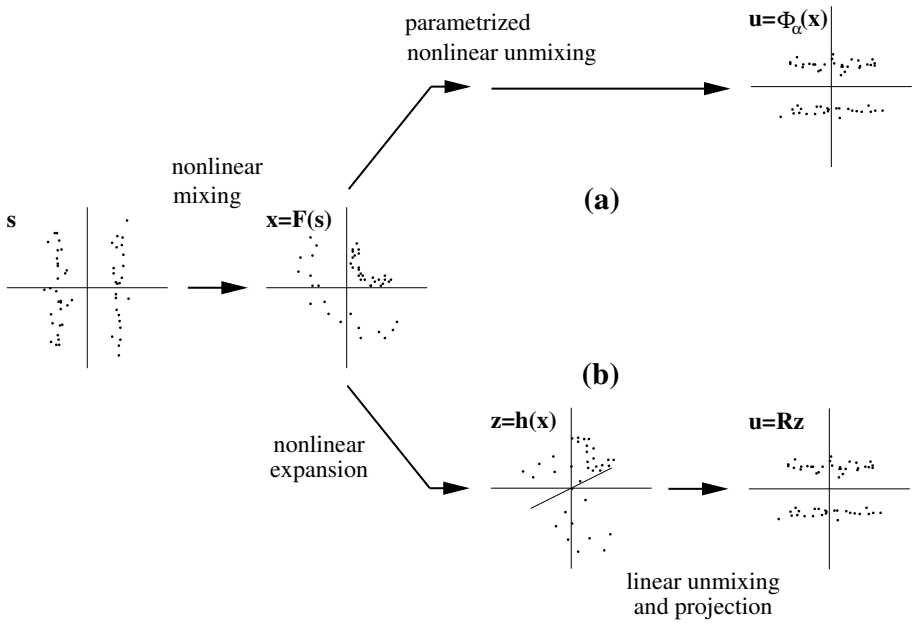


Figure 9.1: Sketch of two different approaches to nonlinear ICA. **(a)** Parametric approach: Learn a nonlinear function Φ_α with parameter set α to find the inverse of $F(\cdot)$. **(b)** Nonlinear expansion approach: The mixture $\mathbf{x}$ is mapped into a high dimensional feature space to obtain $\mathbf{z}$. Linear ICA is applied to $\mathbf{z}$. Results are projected back into input space to obtain the estimated source signal $\mathbf{u}$.

again the reformulation (8.19)of the SFA objective, subject to maximization

$$\Psi_{\text{SFA}}(\mathbf{Q}) \quad = \quad \sum_{i=1}^{M} \left(\mathbf{C}_{ii}^{(\mathbf{u})}(\tau) \right)^2 = \sum_{i=1}^{M} \left(\sum_{k,l=1}^{M} Q_{ik} Q_{il} C_{kl}^{(\mathbf{y})}(\tau) \right)^2 . \tag{9.2}$$

Note that with $\tau = 1$ this objective yields exactly the same slowly varying signals as with the original objective (5.1).

9.2 Independent Slow Feature Analysis

The nonlinear BSS method proposed in this section combines the principle of independence known from linear second-order BSS methods with the principle of slowness as described above. Because of the combination of ICA and SFA we refer to this method as Independent Slow Feature Analysis (ISFA). As already explained, second-order ICA tends to make the output components independent and SFA tends to make them slow. Since we are dealing with a nonlinear mixture we first compute a nonlinearly expanded signal $\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t))$ with $\mathbf{h}(\cdot)\ \mathbb{R}^M \to \mathbb{R}^L$ being some nonlinear function chosen such that $\mathbf{z}(t)$ has zero mean. In a second step $\mathbf{z}(t)$ is whitened to obtain $\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t)$. Finally we apply linear ICA combined with linear SFA on $\mathbf{y}(t)$ in order to find the estimated source signal $\mathbf{u}(t)$. Because of the whitening we know that ISFA, like ICA and SFA, is solved by finding an orthogonal $L \times L$ matrix $\mathbf{Q}$. We write the estimated source signal $\mathbf{u}(t)$ as

$$\mathbf{v}(t) := \begin{bmatrix} \mathbf{u}(t) \\ \tilde{\mathbf{u}}(t) \end{bmatrix} = \mathbf{Q}\mathbf{y}(t) = \mathbf{Q}\mathbf{W}\mathbf{z}(t) = \mathbf{Q}\mathbf{W}\mathbf{h}(\mathbf{x}(t)) , \tag{9.3}$$

where we introduced auxiliary variables $\mathbf{v}(t)$ and $\tilde{\mathbf{u}}(t)$ since R, the dimension of the estimated source signal $\mathbf{u}(t)$, is usually much smaller than L, the dimension of the expanded signal. While the $u_i(t)$ are statistically

independent and slowly varying the components $\tilde{u}_i(t)$ are more quickly varying and may be statistically dependent on each other as well as on the selected components $u_i(t)$. The $\tilde{u}_i(t)$ are irrelevant for the final result but important during the optimization procedure, see below.

To summarize, we have an M-dimensional input $\mathbf{x}(t)$, an L-dimensional nonlinearly expanded and whitened $\mathbf{y}(t)$, and an R-dimensional estimated source signal $\mathbf{u}(t)$. ISFA finds an R dimensional subspace such that the $u_i(t)$ are independent and slowly varying. This is achieved at the expense of all $\tilde{u}_i(t)$.

9.2.1 Objective Function

To recover R source signal components u_i, $i = 1,\ldots,R$ out of an L-dimensional expanded and whitened signal $\mathbf{y}$ the objective reads

$$\Psi_{\mathrm{ISFA}}(u_1,\ldots,u_R;\tau) \quad := \quad b_{\mathrm{ICA}} \sum_{\substack{i,j=1,\\ i\neq j}}^{R} \left(\mathbf{C}_{ij}^{(\mathbf{u})}(\tau)\right)^2 - b_{\mathrm{SFA}} \sum_{i=1}^{R} \left(\mathbf{C}_{ii}^{(\mathbf{u})}(\tau)\right)^2, \tag{9.4}$$

where we simply combine the ICA objective as defined in (7.9) and SFA objective (9.2) weighted by the factors b_{ICA} and b_{SFA}, respectively. Note that the ICA objective is usually applied to the linear case to unmix the linear whitened mixture $\mathbf{y} = \mathbf{W}\mathbf{x}$ whereas here it is used on the nonlinearly expanded whitened signal $\mathbf{y} = \mathbf{W}\mathbf{z}$. ISFA minimizes Ψ_{ISFA}, which is the reason why the SFA part has a negative sign.

9.2.2 Optimization Procedure

From (9.3) we know that $\mathbf{C}^{(\mathbf{u})}(\tau)$ in (9.4) depends on the orthogonal matrix $\mathbf{Q}$. There are several ways to find the orthogonal matrix that minimizes the objective function. Here we apply successive Givens rotations, as defined in Section 4.6.1, to obtain $\mathbf{Q}$. The objective (9.4) as a function of a Givens rotation $\mathbf{Q}^{\mu\nu}$ within the plane of two selected components μ and ν reads

$$\Psi_{ISFA}^{\mu\nu}(\mathbf{Q}^{\mu\nu}) = b_{\mathrm{ICA}} \sum_{\substack{i,j=1,\\ i\neq j}}^{R} \left(\sum_{k,l=1}^{L} Q_{ik}^{\mu\nu} Q_{jl}^{\mu\nu} C_{kl}^{(\mathbf{y})}(\tau)\right)^2 - b_{\mathrm{SFA}} \sum_{i=1}^{R} \left(\sum_{k,l=1}^{L} Q_{ik}^{\mu\nu} Q_{il}^{\mu\nu} C_{kl}^{(\mathbf{y})}(\tau)\right)^2. \tag{9.5}$$

Applying a Givens rotation $\mathbf{Q}^{\mu\nu}$ in the $\mu\nu$-plane changes all covariances $C_{\cdot\cdot}^{(\mathbf{y})}(\tau)$ with at least one of the indices equal to μ or ν. For each Givens rotation there exists an angle $\phi_{\min}$ with minimal $\Psi_{\mathrm{ISFA}}^{\mu\nu}$. Successive application of Givens rotations $\mathbf{Q}^{\mu\nu}$ with rotation angle $\phi_{\min}$ leads to the final rotation matrix $\mathbf{Q}$ yielding

$$\mathbf{Q}^T \mathbf{C}^{(\mathbf{y})}(\tau)\mathbf{Q} \quad = \quad \begin{bmatrix} \mathbf{C}^{(\mathbf{u})}(\tau) & \mathbf{C}^{(\mathbf{u},\tilde{\mathbf{u}})}(\tau) \\ \mathbf{C}^{(\tilde{\mathbf{u}},\mathbf{u})}(\tau)^T & \mathbf{C}^{(\tilde{\mathbf{u}})}(\tau) \end{bmatrix}, \tag{9.6}$$

with $r \times (L-r)$ dimensional $\mathbf{C}^{(\mathbf{u},\tilde{\mathbf{u}})}(\tau)$ with entries

$$C_{ij}^{(\mathbf{u},\tilde{\mathbf{u}})}(\tau) = \left\langle u_i(t)\,\tilde{u}_j(t+\tau) + \tilde{u}_i(t)\,u_j(t+\tau)\right\rangle. \tag{9.7}$$

In the ideal case $\mathbf{C}^{(\mathbf{u})}(\tau)$ is diagonal with a large trace $\sum_i C_{ii}^{(\mathbf{u})}(\tau)$.

Assume we want to minimize Ψ_{ISFA} for a given R, where R denotes the number of signal components we want to unmix. Applying a Givens rotation $\mathbf{Q}^{\mu\nu}$ we have to distinguish three cases

- **Case 1** Both axes, μ and ν, lie inside the subspace spanned by the first R axes ($\mu,\nu \leq R$) (see Fig. 9.2 **(a)**):

  The sum over all squared cross correlations of all signal components that lie outside the subspace is constant as well as those of all signal components inside the subspace. The former holds because of the first invariance (4.39) and the latter because of the first (4.39) and second invariance (4.40). There is no interaction between inside and outside, in fact the objective function is exactly the objective for

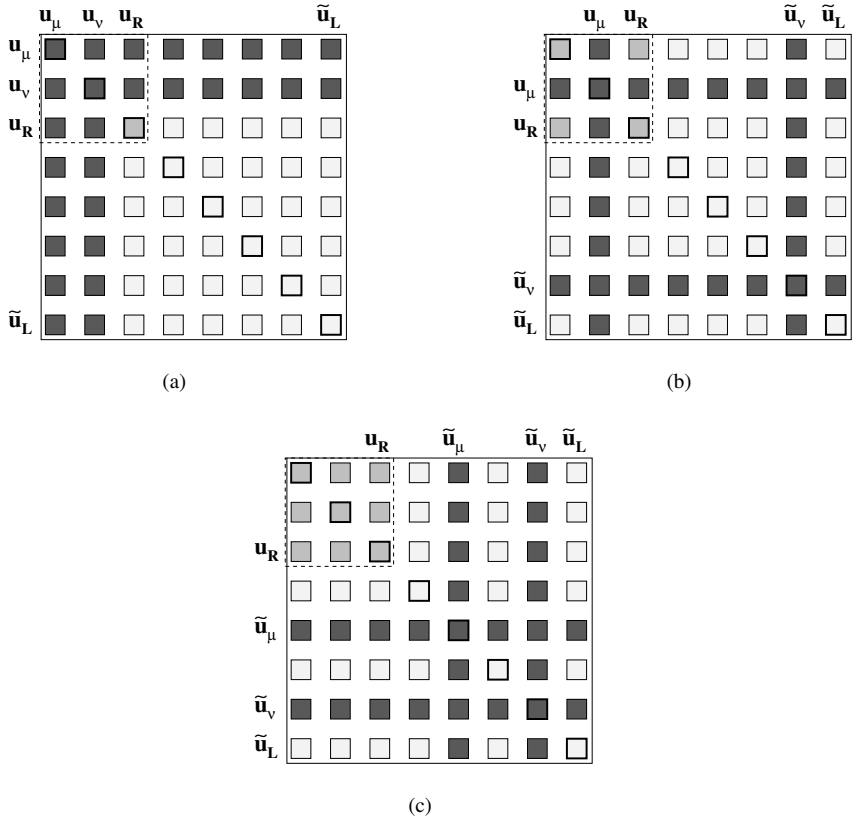


Figure 9.2: The three possible cases during successive plane rotations. Each square represents a squared cross or auto-correlation $\left(C_{ij}^{(\mathbf{v})}\right)^2$ ($\mathbf{v}(t) = [\mathbf{u}(t), \tilde{\mathbf{u}}(t)]^T$) where index $i\,(j)$ denotes the row (column) of the square. Dark squares indicate all entries that are changed by a rotation in the μ-ν-plane. L is the dimensionality of the expanded signal $\mathbf{v}$ and R the number of signal components $u_i(t)$ subject to optimization. The entries corresponding to the correlations that are incorporated in the objective function are located in the upper left corner. The dashed line separates these entries from all others. **(a)** The rotation plane spanned by the μ- and ν-axis lies inside the subspace covered by the objective function. As we have seen in Equations (4.39) and (4.40) the correlations outside the subspace are changing such that their squares are pairwise constant. Therefore the sum over all entries outside the dashed square (and therefore also inside) stays constant. There is no interaction between inside and outside and optimization corresponds to the classical ICA- or SFA-case. **(b)** One of the axes spanning the rotation plane is outside (ν) and the other (μ) inside the subspace covered by the objective function. This is the only case where the entries within the dashed square can be optimized at the expense of those outside. For instance, according to (4.39) $\left(C_{\mu i}^{(\mathbf{u})}\right)^2$ can be optimized at the expense of $\left(C_{\nu i}^{(\tilde{\mathbf{u}})}\right)^2$ with $i \in \{1,\ldots,R\}$; according to (4.40) $\left(C_{\mu\mu}^{(\mathbf{u})}\right)^2$ can be optimized at the expense of $\left(C_{\mu\nu}^{(\tilde{\mathbf{u}})}\right)^2$, $\left(C_{\nu\mu}^{(\tilde{\mathbf{u}})}\right)^2$, and $\left(C_{\nu\nu}^{(\tilde{\mathbf{u}})}\right)^2$. **(c)** The objective function stays constant, since all correlations affected by the rotation are outside the subspace.

an ICA algorithm based on second-order statistics, e.g. TDSEP or SOBI [Belouchrani et al., 1997; Ziehe and Müller, 1998]. In Section 8.1 it has been shown that this is equivalent to SFA in the case of a single time delay.

- **Case 2** Only one axis, w.l.o.g. μ, lies inside the subspace; the other, ν, lies outside ($\mu \leq R < \nu$) (see Fig. 9.2 **(b)**):

  Since one axis of the rotation plane lies outside the subspace, u_μ in the objective function can be optimized at the expense of the $\tilde{u}_\nu$ outside the subspace. A rotation of $\pi/2$, for instance, would simply exchange components u_μ and $\tilde{u}_\nu$. This gives the possibility to find the slowest and most independent components in the whole space spanned by all L axes in contrast to Case 1 where the minimum is searched within the subspace spanned by the first R axes considered in the objective function.

- **Case 3** Both axes lie outside the subspace ($R < \mu, \nu$) (see Fig. 9.2 **(c)**):

  A Givens rotation with the two rotation axes outside the relevant subspace does not affect the objective function and can therefore be disregarded.

To optimize the objective function of ISFA (9.4) we need to calculate the explicit form of the objective function $\Psi^{\mu\nu}_{\text{ISFA}}$ in (9.5) for Cases 1 and 2. By inserting the Givens rotation matrix (4.29) into the objective function (9.5) we can write the latter as a function of the rotation angle ϕ

$$\begin{aligned}
\text{Case 1:} \quad \Psi^{\mu\nu}_{\text{ISFA}}(\phi) &= \left(\sum_{\alpha=0}^{2} d_\alpha \left(\cos(\phi)^{(4-\alpha)} \sin(\phi)^\alpha + \right.\right. \\
&\quad \left.\left. \cos(\phi)^{(\alpha)} (-\sin(\phi))^{(4-\alpha)}\right) + d_c\right), \qquad (9.8) \\
\text{Case 2:} \quad \Psi^{\mu\nu}_{\text{ISFA}}(\phi) &= \left(\sum_{\alpha=0}^{4} d_\alpha \left(\cos(\phi)^{(4-\alpha)} \sin(\phi)^\alpha\right) + d_c\right) + \\
&\quad \left(\sum_{\beta=0}^{2} e_\beta \left(\cos(\phi)^{(2-\beta)} \sin(\phi)^\beta\right) + e_c\right), \qquad (9.9)
\end{aligned}$$

with constants that depend only on the $C^{(\mathbf{y})}_{kl}$ before rotation (see the appendix). It can be shown that like in Section 6.1.3 these objective functions can always be written in the form

$$\begin{aligned}
\text{Case 1:} \quad \Psi^{\mu\nu}_{\text{ISFA}}(\phi) &= A_0 + A_4 \cos(4\phi + \phi_4), \qquad (9.10) \\
\text{Case 2:} \quad \Psi^{\mu\nu}_{\text{ISFA}}(\phi) &= A_0 + A_2 \cos(2\phi + \phi_2) + A_4 \cos(4\phi + \phi_4), \qquad (9.11)
\end{aligned}$$

with a single minimum (if w.l.o.g. $\phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$) that can easily be calculated (see e.g. [Blaschke and Wiskott, 2004]). The derivation of (9.10) and (9.11) involves various trigonometric identities and, because of its length, is documented in Appendix D.1. Interestingly Equations (9.10) and (9.11) do not change (of course the constants do) if we consider more than one time delay (cf. Appendix D.1).

It is easy to see why it is possible to write both objective functions (9.10) and (9.11) in such a simple form. Firstly, the terms in (9.8) and (9.9) are products of at most four $\sin(\phi)$ and $\cos(\phi)$ functions which allows, at most, a frequency of 4. Secondly, in Case 1 $\Psi^{\mu\nu}_{\text{ISFA}}$ has a periodicity of $\pi/2$ because rotations by multiples of $\pi/2$ correspond to a permutation (possibly plus sign change) of the two components. Since both components are inside the subspace, permutations do not change the objective function and the objective function has a $\pi/2$ periodicity. Thus we conclude that only frequencies of 0 and 4 can be present in (9.10). In Case 2, since one component lies outside the subspace, an exchange of components will change the objective function (9.11). A rotation by multiples of π however, which results only in a possible sign change, will leave the objective function unchanged, resulting in an objective function with π-periodicity and therefore frequencies of 0, 2 and 4.

The iterative approach with successive Givens rotations can be described by

(a) Initialize $\mathbf{Q}' = \mathbf{I}$ and $\mathbf{y}(t)' = \mathbf{y}(t)$ and compute $\mathbf{C}^{(\mathbf{y}')}$.

(b) Choose two axes μ and ν with $\mu \leq R$ (either randomly or in a pseudo-random order).

(c) Determine the optimal rotation angle $\phi_{\min}^{\mu\nu}$ for the selected axes from (9.10) or (9.11).

(d) Compute the Givens rotation matrix $\mathbf{Q}^{\mu\nu}\left(\phi_{\min}^{\mu\nu}\right)$ defined by (4.29).

(e) Update $\mathbf{C}^{(\mathbf{y}')}$ using $\mathbf{C}^{(\mathbf{y}')} \rightarrow (\mathbf{Q}^{\mu\nu})^T \mathbf{C}^{(\mathbf{y}')} \mathbf{Q}^{\mu\nu}$.

(f) Update $\mathbf{Q}'$ according to $\mathbf{Q}' \rightarrow \mathbf{Q}^{\mu\nu}\mathbf{Q}'$.

(g) Go to b until all rotation angles $\phi_{\min}^{\mu\nu}$ of a sweep through all possible rotation planes are below ε.

(h) Set $\mathbf{Q} = \mathbf{Q}'$ and $\mathbf{v}(t) = \mathbf{Q}\mathbf{y}(t)$.

In Step (b) it is important to notice that the rotation planes of the Givens rotations are selected from the whole L-dimensional space (although we avoid the irrelevant Case 3 by requiring $\mu \leq R$; see Fig. 9.2(c)) whereas the objective function only uses information of correlations among the first R signal components u_i. Steps (e) and (f) do not require a full matrix-multiplication but can be efficiently computed since $\mathbf{Q}_{\mu\nu}$ is very sparse. After convergence the additional components $\tilde{u}_j$ $(j = R+1,\ldots,L)$ can be discarded. There is no proof that the final minimum is also the global one. However, local minima can be avoided by the method used in the next section.

9.2.3 Incremental Extracting of Independent Components

It is possible to find the number of independent source signal components R by successively increasing the number of components to be extracted. In each step the objective function (9.4) is optimized for a fixed R. First a single signal component is extracted ($R = 1$), which can be achieved by plain SFA. Then an additional component is considered in the objective function ($R = 2$). The following scheme gives a sketch of the incremental procedure.

(a) Optimize Ψ_{ISFA} on $\mathbf{y}(t)$ with $R = 1$ yielding an initial $\mathbf{v}(t)$.

(b) Set $R \rightarrow R+1$, $k = R$, and $\mathbf{y} = \mathbf{v} = [\mathbf{u}, \tilde{\mathbf{u}}]^T$.

(c) Exchange components $y_k(t)$ and $y_R(t)$ (this has no effect in the first iteration of k).

(d) Optimize objective function Ψ_{ISFA} on $\mathbf{y}(t)$ yielding a new $\mathbf{v}(t)$.

(e) If optimization has yielded an additional independent signal $u_R(t)$ (according to some measure of independence; see below) go to step b.

(f) Set k $\rightarrow$ k + 1.

(g) If k > L stop algorithm; the number of extracted source signal components is R.

(h) Go to Step c

The permutation of components (Step (c)) is necessary to circumvent the problem of getting stuck in local optima of the objective function (9.4). Instead of going through the components (index k) systematically one can choose a pseudo-random sequence. The algorithm is stopped when no additional signal component can be extracted. In Step (e) any suitable measure of independence can be applied; we used the sum over squared cross-cumulants of fourth order

$$\Phi \quad = \quad \sum_i \sum_{j>i} \sum_{k>j} \sum_{l>k} C_{ijkl}^{(\mathbf{u})}. \tag{9.12}$$

In our artificial examples this value is typically small for independent components and increases by two orders of magnitudes if the number of components to be extracted is greater than the number of original source signal components. In real world applications the decision will probably be less clear and an appropriate heuristics for the threshold must be adopted.

9.3 Simulations

9.3.1 Simple Example

Here we show a simple example with two nonlinearly mixed signal components as shown in Figure 9.3. The mixture reads

$$x_1(t) = s_1 + 2 * s_2^2 \tag{9.13}$$
$$x_2(t) = s_2 . \tag{9.14}$$

As nonlinearities we used monomials up to second degree. To weight the SFA and the ICA part in Equation (9.4) we used $[b_{\mathrm{SFA}}, b_{\mathrm{ICA}}] = [1, 100]$ throughout the simulation. The number of time delays was $T = 20$. The results are shown in Table 9.1. CuBICA34 was not able to extract the source signal components, whereas ISFA obtained good results. For an illustration see Figure 9.3.

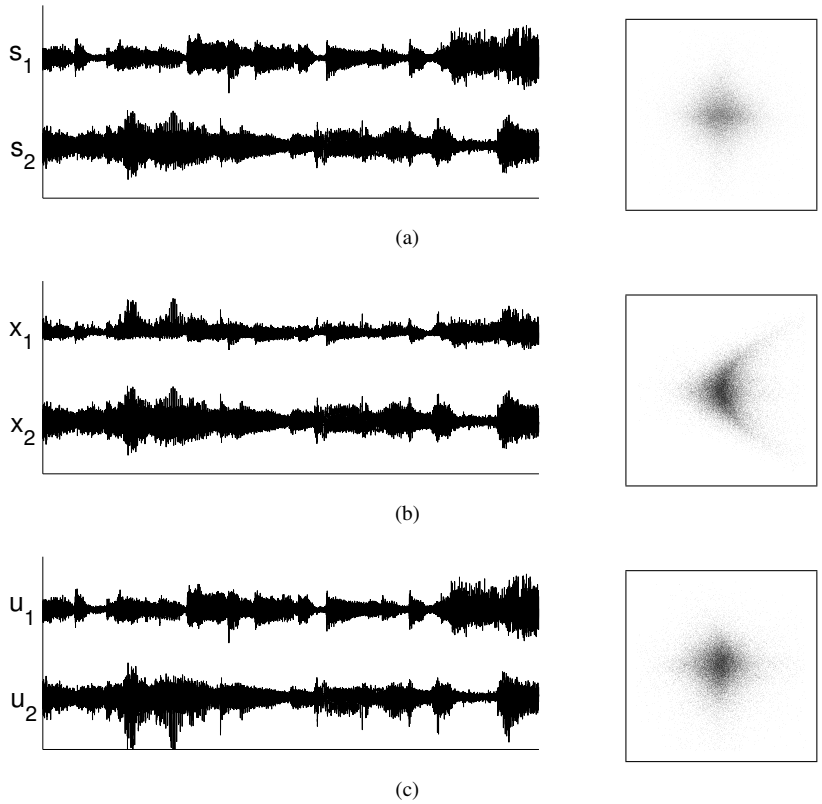


Figure 9.3: Waveforms of **(a)** the original source signal components s_i, **(b)** the nonlinear mixture (9.14) and **(c)** recovered components with nonlinear ISFA (u_i). As an unmixing nonlinearity we used all monomials up to degree 2.

9.3.2 Twisted Speech Data

This is a more complex example introduced by Harmeling et al. [2003]. We chose it to allow a direct comparison with kTDSEP. The mixture is defined by

$$x_1(t) = (s_2(t) + 3s_1(t) + 6)\cos(1.5\pi s_1(t)) \tag{9.15}$$
$$x_2(t) = (s_2(t) + 3s_1(t) + 6)\sin(1.5\pi s_1(t)) . \tag{9.16}$$

	linear		degree 2	
	u_1	u_2	u_1	u_2
s_1	-0.558	-0.581	0.996	-0.004
s_2	0.726	-0.687	-0.007	-0.989

Table 9.1: Correlation coefficients of extracted (u_1 and u_2) and original (s_1 and s_2) source signal components of the mixture described in Equation (9.14) for linear CuBICA34 (first column) and ISFA using monomials up to second degree. Weighting constants are $[b_{\mathrm{SFA}}, b_{\mathrm{ICA}}] = [1, 100]$ and the number of time delays is $T = 20$.

We used the ISFA algorithm with different nonlinearities (see Tab. 9.2). Again, we weighted the SFA and ICA parts like $[b_{\mathrm{SFA}}, b_{\mathrm{ICA}}] = [1, 100]$ and set the number of time delays to $T = 20$. Already a nonlinear expansion with monomials up to degree three was sufficient to give good results in extracting the original source signal. In all cases ISFA did find exactly two independent signal components. Using all monomials up to degree five led to results that showed virtually no difference between estimated and true source signal (see Fig. 9.4). A linear BSS method failed completely to find a good unmixing matrix.

For comparison we also give the results of kTDSEP, a nonlinear BSS approach by Harmeling et al. [2003] with a similar design as ISFA. With kTDSEP the nonlinear BSS problem is solved using a two step approach, too. First the observed signal is mapped to a high-dimensional kernel-feature-space and then linear second-order ICA is applied to this signal. To select the right components to be extracted this method is applied twice in succession and components that have strong correlations to those of the first sweep are assumed to be the true source signal components. The assumption made here is that the true source signal components are more reliable, that is they appear again after a second pass of the algorithm. For linear ICA this seems to be a reasonable assumption [Harmeling et al., 2004] and it works well in the examples shown in [Harmeling et al., 2003].

	linear		degree 2		degree 3	
	u_1	u_2	u_1	u_2	u_1	u_2
s_1	-0.890	0.215	0.936	0.013	0.001	0.988
s_2	-0.011	-0.065	-0.027	0.149	-0.977	0.006

	degree 4		degree 5		kTDSEP	
	u_1	u_2	u_1	u_2	u_1	u_2
s_1	0.002	-0.996	0.998	-0.000	0.990	-
s_2	0.983	-0.000	-0.000	0.994	-	0.947

Table 9.2: Correlation coefficients of extracted (u_1 and u_2) and original (s_1 and s_2) source signal components for CuBICA34 (first column) and ISFA with different nonlinearities. Weighting constants are $[b_{\mathrm{SFA}}, b_{\mathrm{ICA}}] = [1, 100]$ and the number of time delays is $T = 20$. Shown are results with monomials up to degree 2, 3, 4 and 5. Note, that the source signal can only be estimated up to permutation and scaling, resulting in different signs and permutations of u_1 and u_2. The correlation coefficients for kTDSEP were taken from Harmeling et al. [2003] with same mixture but different source signal. The kernel used in kTDSEP was a Gaussian RBF.

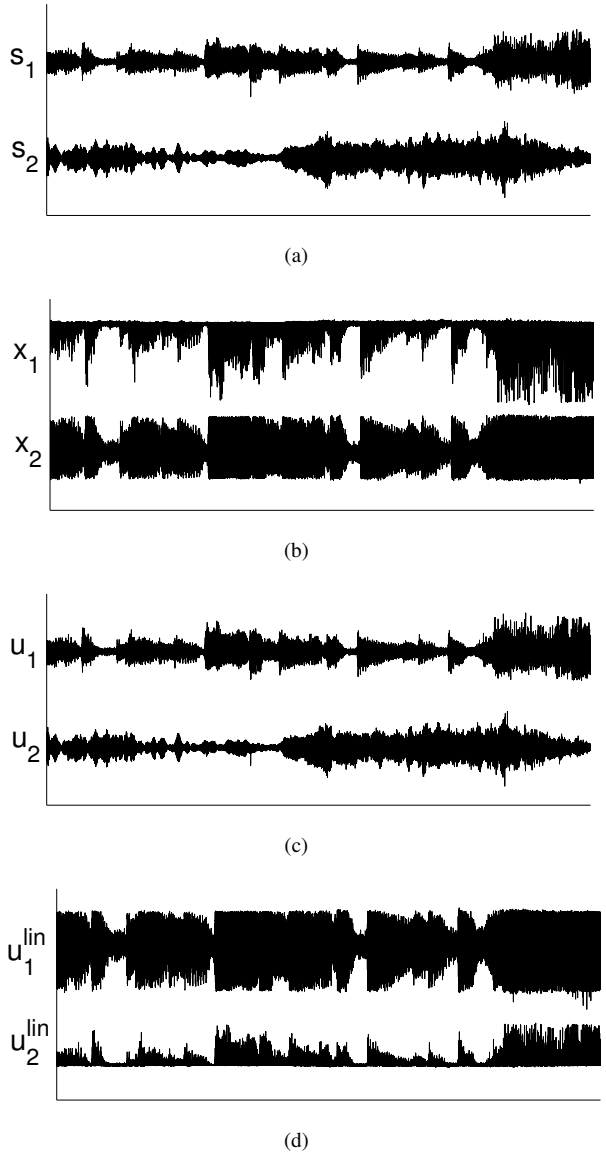


Figure 9.4: Waveforms of **(a)** the original source signal components $s_i(t)$, **(b)** the nonlinear mixture (9.16) **(c)** recovered components with nonlinear ISFA ($u_i(t)$), and **(d)** with CuBICA34 ($u_i(t)^{lin}$). As an unmixing nonlinearity we used all monomials up to degree 5.

9.4 Conclusion

We have shown that combining the ideas of independent component analysis and slow feature analysis into ISFA is a possible way to solve the nonlinear blind source separation problem for signals with autocorrelations. SFA favors independent components that are slowly varying, which seems to be a good way to discriminate between the original and nonlinearly distorted source signal components. A simple simulation showed that ISFA is able to extract the original source signal out of a nonlinear mixture. Furthermore ISFA can predict the number of source signal components via an incremental optimization scheme. Note that from the SFA point of view ISFA is a natural extension to standard SFA. While in standard SFA all extracted slowly varying signal components are uncorrelated or statistically independent up to second order,

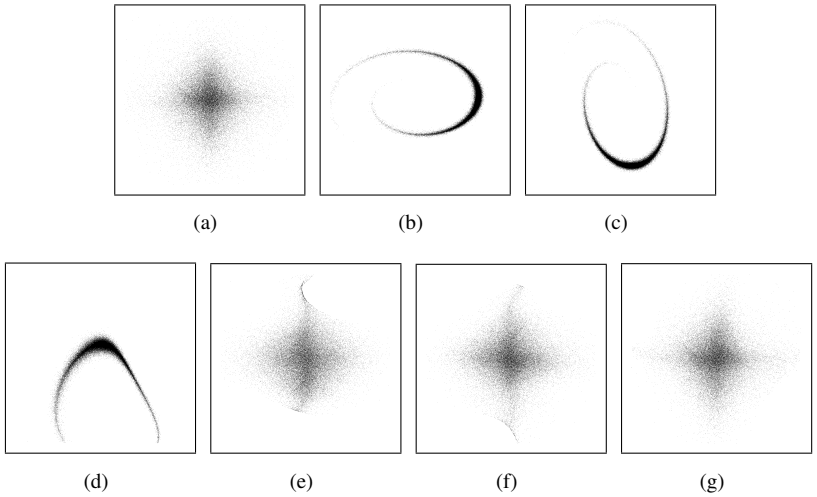


Figure 9.5: Scatter plot of source signal components s_1 and s_2, nonlinear mixture components x_1 and x_2 of mixture (9.16), and extracted components u_1 and u_2 with different nonlinearities. **(a)** source signal components; **(b)** mixture components; **(c)** results of linear ICA; **(d - g)** results of ISFA using **(d)** monomials up to degree 2, **(e)** monomials up to degree 3, **(f)** monomials up to degree 4, **(g)** monomials up to degree 5.

the extracted signal components in ISFA are statistically independent also for higher orders.

10

Conclusions

Within this thesis, we focused on the relation between independent component analysis (ICA) and slow feature analysis (SFA). For that purpose, we first introduced the two methods and provided the reader with the necessary tools to make a comparison possible. For example, we have derived a suitable ICA algorithm and an alternative formulation of SFA. In the following we list the results of the main Chapters 6-9.

In Chapter 6, we introduced an improved ICA algorithm, called CuBICA (**Cu**mulant **B**ased **I**ndependent **C**omponent **A**nalysis), based on higher-order cumulants. It is similar to the original algorithm by Comon [1994a] but implements some important improvements. First, the contrast function of CuBICA integrates cumulants of third- and of fourth-order. Thus, it is able to separate linear mixtures of symmetrically and skew-symmetrically distributed source signal components. Second, the contrast function can be formulated in a simple mathematical way. For example the contrast for an ICA task with only two signal components can be expressed as a simple cosine-function with the angle being the only optimization parameter. This allows to compute the optimal solution in a single step. CuBICA is not a suitable algorithm if one wants to compare ICA and SFA. Most of all because it is insensitive to any time structure of the input signal while SFA needs signals with non-vanishing auto-correlation. However, CuBICA is a nice ICA algorithm with a wide range of possible applications, in any case.

In Chapter 7, we focused on a different ICA algorithm, based on second-order statistics only, i.e. cross-correlations. In contrast to algorithms based on higher-order statistics like CuBICA not only instantaneous cross-correlations but also time-delayed cross correlations are considered for minimization. Some implementations of this method exist in the literature, some using a single time delay (e.g. [Molgedey and Schuster, 1994]) or several time delays (e.g. [Belouchrani et al., 1997; Nuzillard and Nuzillard, 2003; Zibulevsky and Pearlmutter, 2000; Ziehe and Müller, 1998]). Their common property is their requirement of signal components with auto-correlation like in SFA, and the ability to separate source signal components that have a Gaussian distribution. In this chapter we could derive an improved version called CuBICA2. The mathematical derivation of its cost function is similar to that of CuBICA that has been introduced in the previous chapter, and again the resulting formula has a very simple form allowing for easy optimization. A comparison with a similar algorithm like TDSEP [Ziehe and Müller, 1998] showed significant improvements in terms of time consumption.

In Chapter 8, we investigated the relation between ICA based on second-order statistics and SFA. For this purpose we derived an alternative formulation of the SFA objective function and compared it with that of CuBICA2. We used CuBICA2 because of its similarities with SFA, namely both algorithms are exclusively based on second-order statistics, and both rquire signals that have non-vanishing auto-correlation. It turned out, that in the case of a linear mixture the two methods are equivalent if a single time delay is taken into account. The comparison could not be extended to the case of several time delays. For ICA a straightforward extension could be derived in the previous section. A similar extension to SFA yields an objective function that can not be interpreted in the sense of SFA, which is defined as extracting the slowly varying signal from

a given input signal. However, a useful extension in the sense of SFA to more than one time delay could be derived. This extended SFA is nearly equivalent to a different ICA algorithm introduced by Stone [2001] using the objective of temporal predictability. Thus, there exists a close connection between the slowness objective of SFA and temporal predictability, which can be interpreted intuitively: a signal component with large auto-correlation has a high temporal predictability. Predictability on the other hand means that the signal component has to vary slowly.

In Chapter 9, the aim was to combine ICA and SFA. Again, this has been done using the second-order ICA algorithm introduced in Chapter 7. The result can be interpreted from two perspectives. From the ICA point of view the combination leads to an algorithm that solves the nonlinear blind source separation problem. We know from linear blind source separation (BSS) that mutual statistical independence of the source signal components is a sufficient criterion to solve the BSS task. However, for nonlinear BSS this is no longer the case. Further assumptions about the source signal components have to be made. The integration of SFA adds the assumption that correct output-signal-components have non-vanishing auto-correlations. From the SFA point of view the combination of ICA and SFA is an extension to SFA in terms of statistical independence. Standard SFA extracts slowly varying signal components that are uncorrelated meaning they are statistically independent up to second-order. The integration of ICA leads to signal components that are more or less statistically independent.

We have shown that the two objectives independence and slowness are strongly connected, in fact they are equivalent under certain conditions. The combination of ICA and SFA leads to an algorithm for nonlinear blind source separation respectively to extended SFA, where extracted signal components are statistically independent. Furthermore, we have derived an easy to use improved ICA algorithm that can handle symmetric and asymmetric distributed sources. Additionally it shows good performance and thus, may be a good general algorithm for performing ICA.

A

Givens Rotations

A.1 Derivation of Equation (4.37)

Lets insert (4.33) into (4.35) to obtain

$$\begin{aligned}\Psi_{\text{diag}} &= \left[\cos(\phi)^2 M_{12} - \cos(\phi)\sin(\phi) M_{11} + \cos(\phi)\sin(\phi) M_{22} - \sin(\phi)^2 M_{21}\right]^2 + \\ &\quad \left[\sin(\phi)^2 M_{21} - \cos(\phi)\sin(\phi) M_{11} + \cos(\phi)\sin(\phi) M_{22} - \cos(\phi)^2 M_{12}\right]^2 . \end{aligned} \tag{A.1}$$

After carrying out all multiplications and rearranging the terms we derive

$$\begin{aligned}\Psi_{\text{diag}} &= \left(\cos(\phi)^4 + \sin(\phi)^4\right)\left(M_{12}^2 + M_{21}^2\right) + \\ &\quad \left(\cos(\phi)^3\sin(\phi) - \sin(\phi)^3\cos(\phi)\right)\left(-2M_{11}M_{12} - 2M_{11}M_{21} + 2M_{12}M_{22} + 2M_{21}M_{22}\right) + \\ &\quad \left((\cos(\phi)^2\sin(\phi)^2 + \cos(\phi)^2\sin(\phi)^2\right)\left(M_{11}^2 - 2M_{12}M_{21} - 2M_{11}M_{22} + M_{22}^2\right) . \end{aligned} \tag{A.2}$$

Defining constants

$$c_0 := \left(M_{12}^2 + M_{21}^2\right) , \tag{A.3}$$

$$c_1 := \left(-2M_{11}M_{12} - 2M_{11}M_{21} + 2M_{12}M_{22} + 2M_{21}M_{22}\right) , \tag{A.4}$$

$$c_2 := \left(M_{11}^2 - 2M_{12}M_{21} - 2M_{11}M_{22} + M_{22}^2\right) , \tag{A.5}$$

Equation (A.2) can be rewritten as

$$\begin{aligned}\Psi_{\text{diag}} &= \left(\cos(\phi)^4 + \sin(\phi)^4\right) c_0 + \left(\cos(\phi)^3\sin(\phi) - \sin(\phi)^3\cos(\phi)\right) c_1 + \\ &\quad \left((\cos(\phi)^2\sin(\phi)^2 + \cos(\phi)^2\sin(\phi)^2\right) c_2 \end{aligned} \tag{A.6}$$

$$= \sum_{i=0}^{2} c_i \left(\cos(\phi)^{4-i}\sin(\phi)^i + (-1)^i \sin(\phi)^{4-i}\cos(\phi)^i\right) . \tag{A.7}$$

Using some trigonometric relations

$$\left(\cos(\phi)^4 + \sin(\phi)^4\right) = \frac{1}{4}\left(\cos(4\phi) + 3\right) , \tag{A.8}$$

$$\left(\cos(\phi)^3\sin(\phi) - \sin(\phi)^3\cos(\phi)\right) = \frac{1}{4}\sin(4\phi) , \tag{A.9}$$

$$\left((\cos(\phi)^2\sin(\phi)^2 + \cos(\phi)^2\sin(\phi)^2\right) = \frac{1}{4}\left(1 - \cos(4\phi)\right) , \tag{A.10}$$

and some rearrangement of terms we arrive at

$$\Psi_{\text{diag}} = \frac{3}{4}c_0 + \frac{1}{4}c_2 + \frac{1}{4}c_1 \sin(4\phi) + \frac{1}{4}\cos(4\phi)(c_0 - c_2)\,, \tag{A.11}$$

Finally we can put together the sine- and cosine-term on the right hand side of (A.11) and derive

$$\Psi_{\text{diag}} = A_0 + A_4 \cos(4\phi + \phi_4)\,, \tag{A.12}$$

with constants defined by

$$A_0 := \frac{1}{4}(3c_0 + c_2)\,, \tag{A.13}$$

$$A_4 := \frac{1}{4}\sqrt{(c_0 - c_2)^2 + c_1^2}\,, \tag{A.14}$$

$$-\tan(\phi_4) := \frac{c_1}{c_0 - c_2}\,. \tag{A.15}$$

A.2 Derivation of Invariances (4.39) and (4.40)

Given a vectorial signal $\mathbf{y}$ we consider a Givens rotation $\mathbf{u} = \mathbf{Q}^{\mu\nu}\mathbf{y}$ within the μ,ν plane and a matrix representation as defined in (4.29). Cumulant tensors in $\mathbf{y}$ of all order show some invariances under such transformations. We show here only second-order cumulant matrices $\mathbf{C}^{(\mathbf{y})}$, since this is the simplest non-trivial case.

Constants under rotation in this plane with rotation angle ϕ are

$$\begin{aligned}
\left(C^{(\mathbf{u})}_{\mu\alpha}\right)^2 + \left(C^{(\mathbf{u})}_{\nu\alpha}\right)^2 &= \left[\sum_{\beta,\gamma=1}^{N} Q_{\mu\beta}Q_{\alpha\gamma}C^{(\mathbf{y})}_{\beta\gamma}\right]^2 + \left[\sum_{\beta,\gamma=1}^{N} Q_{\nu\beta}Q_{\alpha\gamma}C^{(\mathbf{y})}_{\beta\gamma}\right]^2 \\
&= \left(\cos(\phi)C^{(\mathbf{y})}_{\mu\alpha} + \sin(\phi)C^{(\mathbf{y})}_{\nu\alpha}\right)^2 + \left(\cos(\phi)C^{(\mathbf{y})}_{\nu\alpha} - \sin(\phi)C^{(\mathbf{y})}_{\mu\alpha}\right)^2 \\
&= \left[\cos(\phi)^2 + \sin(\phi)^2\right]\left(C^{(\mathbf{y})}_{\mu\alpha}\right)^2 + \left[\cos(\phi)^2 + \sin(\phi)^2\right]\left(C^{(\mathbf{y})}_{\nu\alpha}\right)^2 \\
&\quad +2\left[\cos(\phi)\sin(\phi) - \cos(\phi)\sin(\phi)\right]C^{(\mathbf{y})}_{\mu\alpha}C^{(\mathbf{y})}_{\nu\alpha} \\
&= \left(C^{(\mathbf{y})}_{\mu\alpha}\right)^2 + \left(C^{(\mathbf{y})}_{\nu\alpha}\right)^2\,,
\end{aligned} \tag{A.16}$$

where $\alpha \neq \mu,\nu$. This sum is therefore constant under rotation in the $\mu\nu$-plane. There is a second constant

under the same rotation

$$\begin{aligned}
\left(C_{\mu\mu}^{(\mathbf{u})}\right)^2+2\left(C_{\mu\nu}^{(\mathbf{u})}\right)^2+\left(C_{\nu\nu}^{(\mathbf{u})}\right)^2 &= \left[\sum_{\beta,\gamma=1}^{N} Q_{\mu\beta}Q_{\mu\gamma}C_{\beta\gamma}^{(\mathbf{y})}\right]^2+2\left[\sum_{\beta,\gamma=1}^{N} Q_{\mu\beta}Q_{\nu\gamma}C_{\beta\gamma}^{(\mathbf{y})}\right]^2 \\
&\quad +\left[\sum_{\beta,\gamma=1}^{N} Q_{\nu\beta}Q_{\nu\gamma}C_{\beta\gamma}^{(\mathbf{y})}\right]^2 \\
&= \left(\cos(\phi)^2 C_{\mu\mu}^{(\mathbf{y})}+2\cos(\phi)\sin(\phi)C_{\mu\nu}^{(\mathbf{y})}+\sin(\phi)^2 C_{\nu\nu}^{(\mathbf{y})}\right)^2 \\
&\quad +2\left(\cos(\phi)^2 C_{\mu\nu}^{(\mathbf{y})}-\cos(\phi)\sin(\phi)C_{\mu\mu}^{(\mathbf{y})}\right. \\
&\quad \left.+\cos(\phi)\sin(\phi)C_{\nu\nu}^{(\mathbf{y})}-\sin(\phi)^2 C_{\nu\mu}^{(\mathbf{y})}\right)^2 \\
&\quad +\left(\sin(\phi)^2 C_{\mu\mu}^{(\mathbf{y})}-2\sin(\phi)\cos(\phi)C_{\mu\nu}^{(\mathbf{y})}+\cos(\phi)^2 C_{\nu\nu}^{(\mathbf{y})}\right)^2 \\
&= \left(\cos(\phi)^4+\sin(\phi)^4+2\cos(\phi)^2\cos(\phi)^2\right)\left(\left(C_{\mu\mu}^{(\mathbf{y})}\right)^2+2\left(C_{\mu\nu}^{(\mathbf{y})}\right)^2+\left(C_{\nu\nu}^{(\mathbf{y})}\right)^2\right) \\
&= \left(\cos(\phi)^2+\sin(\phi)^2\right)^2\left(\left(C_{\mu\mu}^{(\mathbf{y})}\right)^2+2\left(C_{\mu\nu}^{(\mathbf{y})}\right)^2+\left(C_{\nu\nu}^{(\mathbf{y})}\right)^2\right) \\
&= \left(C_{\mu\mu}^{(\mathbf{y})}\right)^2+2\left(C_{\mu\nu}^{(\mathbf{y})}\right)^2+\left(C_{\nu\nu}^{(\mathbf{y})}\right)^2, \qquad \text{(A.17)}
\end{aligned}$$

where we used the fact, that $\mathbf{C}^{(\mathbf{y})}$ and $\mathbf{C}^{(\mathbf{u})}$ are symmetric matrices.

B

Constants in Linear ICA

B.1 Constants in CuBICA34, CuBICA4, CuBICA34a and CuBICA4a

B.1.1 Constants in Equation (6.7)

The definitions of d_{ni} follow directly from the multilinearity of $C_{\cdots}^{(\mathbf{u})}$:

$$d_{30} := \left({C_{111}^{(\mathbf{y})}}^2 + {C_{222}^{(\mathbf{y})}}^2 \right), \tag{B.1}$$

$$d_{31} := 6\left(C_{111}^{(\mathbf{y})} C_{112}^{(\mathbf{y})} - C_{122}^{(\mathbf{y})} C_{222}^{(\mathbf{y})} \right), \tag{B.2}$$

$$d_{32} := 9\left({C_{112}^{(\mathbf{y})}}^2 + {C_{122}^{(\mathbf{y})}}^2 \right) + 6\left(C_{111}^{(\mathbf{y})} C_{122}^{(\mathbf{y})} + C_{112}^{(\mathbf{y})} C_{222}^{(\mathbf{y})} \right), \tag{B.3}$$

$$d_{33} := 2C_{111}^{(\mathbf{y})} C_{222}^{(\mathbf{y})} + 18 C_{112}^{(\mathbf{y})} C_{122}^{(\mathbf{y})}, \tag{B.4}$$

$$d_{40} := \left({C_{1111}^{(\mathbf{y})}}^2 + {C_{2222}^{(\mathbf{y})}}^2 \right), \tag{B.5}$$

$$d_{41} := 8\left(C_{1111}^{(\mathbf{y})} C_{1112}^{(\mathbf{y})} - C_{1222}^{(\mathbf{y})} C_{2222}^{(\mathbf{y})} \right), \tag{B.6}$$

$$d_{42} := 16\left({C_{1112}^{(\mathbf{y})}}^2 + {C_{1222}^{(\mathbf{y})}}^2 \right) + 12\left(C_{1111}^{(\mathbf{y})} C_{1122}^{(\mathbf{y})} + C_{1122}^{(\mathbf{y})} C_{2222}^{(\mathbf{y})} \right), \tag{B.7}$$

$$d_{43} := 48\left(C_{1112}^{(\mathbf{y})} C_{1122}^{(\mathbf{y})} - C_{1122}^{(\mathbf{y})} C_{1222}^{(\mathbf{y})} \right) + 8\left(C_{1111}^{(\mathbf{y})} C_{1222}^{(\mathbf{y})} - C_{1112}^{(\mathbf{y})} C_{2222}^{(\mathbf{y})} \right), \tag{B.8}$$

$$d_{44} := 36{C_{1122}^{(\mathbf{y})}}^2 + 32 C_{1112}^{(\mathbf{y})} C_{1222}^{(\mathbf{y})} + 2 C_{1111}^{(\mathbf{y})} C_{2222}^{(\mathbf{y})}. \tag{B.9}$$

B.1.2 Constants in Equation (6.10)

From (6.7) one can derive

$$\Psi_n(\phi, \mathbf{y}) = a_{n0} + s_{n4}\sin(4\phi) + c_{n4}\cos(4\phi) + s_{n8}\sin(8\phi) + c_{n8}\cos(8\phi) \quad \text{for } n \in \{3,4\}, \tag{B.10}$$

with

$$
\begin{aligned}
a_{30} :=& \frac{1}{3!}\frac{1}{8}\left[5\left(C^{(\mathbf{y})}_{111}{}^2 + C^{(\mathbf{y})}_{222}{}^2\right)\right. \\
&\left. + 9\left(C^{(\mathbf{y})}_{112}{}^2 + C^{(\mathbf{y})}_{122}{}^2\right) + 6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{122} + C^{(\mathbf{y})}_{112}C^{(\mathbf{y})}_{222}\right)\right], \qquad \text{(B.11)} \\
a_{40} :=& \frac{1}{4!}\frac{1}{64}\left[35\left(C^{(\mathbf{y})}_{1111}{}^2 + C^{(\mathbf{y})}_{2222}{}^2\right) + 80\left(C^{(\mathbf{y})}_{1112}{}^2 + C^{(\mathbf{y})}_{1222}{}^2\right)\right. \\
&+60\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122} + C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) \\
&\left. + 108C^{(\mathbf{y})}_{1122}{}^2 + 96C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222} + 6C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222}\right], \qquad \text{(B.12)} \\
s_{34} :=& \frac{1}{3!}\frac{1}{4}\left[6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{112} - C^{(\mathbf{y})}_{122}C^{(\mathbf{y})}_{222}\right)\right], \qquad \text{(B.13)} \\
c_{34} :=& \frac{1}{3!}\frac{1}{8}\left[3\left(C^{(\mathbf{y})}_{111}{}^2 + C^{(\mathbf{y})}_{222}{}^2\right)\right. \\
&\left. - 9\left(C^{(\mathbf{y})}_{112}{}^2 + C^{(\mathbf{y})}_{122}{}^2\right) - 6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{122} + C^{(\mathbf{y})}_{112}C^{(\mathbf{y})}_{222}\right)\right], \qquad \text{(B.14)} \\
s_{44} :=& \frac{1}{4!}\frac{1}{32}\left[56\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1112} - C^{(\mathbf{y})}_{1222}C^{(\mathbf{y})}_{2222}\right)\right. \\
&+ 48\left(C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1122} - C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{1222}\right) \\
&\left. +8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1222} - C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{2222}\right)\right], \qquad \text{(B.15)} \\
c_{44} :=& \frac{1}{4!}\frac{1}{16}\left[7\left(C^{(\mathbf{y})}_{1111}{}^2 + C^{(\mathbf{y})}_{2222}{}^2\right) - 16\left(C^{(\mathbf{y})}_{1112}{}^2 + C^{(\mathbf{y})}_{1222}{}^2\right)\right. \\
&-12\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122} + C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) \\
&\left. - 36C^{(\mathbf{y})}_{1122}{}^2 - 32C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222} - 2C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222}\right], \qquad \text{(B.16)} \\
s_{38} :=& 0, \qquad \text{(B.17)} \\
c_{38} :=& 0, \qquad \text{(B.18)} \\
s_{48} :=& \frac{1}{4!}\frac{1}{64}\left[8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1112} - C^{(\mathbf{y})}_{1222}C^{(\mathbf{y})}_{2222}\right)\right. \\
&-48\left(C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1122} - C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{1222}\right) \\
&\left. -8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1222} - C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{2222}\right)\right], \qquad \text{(B.19)} \\
c_{48} :=& \frac{1}{4!}\frac{1}{64}\left[\left(C^{(\mathbf{y})}_{1111}{}^2 + C^{(\mathbf{y})}_{2222}{}^2\right) - 16\left(C^{(\mathbf{y})}_{1112}{}^2 + C^{(\mathbf{y})}_{1222}{}^2\right)\right. \\
&-12\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122} + C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) \\
&\left. + 36C^{(\mathbf{y})}_{1122}{}^2 + 32C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222} + 2C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222}\right]. \qquad \text{(B.20)}
\end{aligned}
$$

With this it is trivial to determine the constants for $\Psi_{34}(\phi,\mathbf{y}) = \Psi_3(\phi,\mathbf{y}) + \Psi_4(\phi,\mathbf{y})$ in the form given in (6.10). We find:

$$A_0 := a_{30} + a_{40}\,, \tag{B.21}$$

$$A_4 := \sqrt{(c_{34}+c_{44})^2 + (s_{34}+s_{44})^2}\,, \tag{B.22}$$

$$A_8 := \sqrt{c_{48}^2 + s_{48}^2}\,, \tag{B.23}$$

$$\tan(\phi_4) := -\frac{s_{34}+s_{44}}{c_{34}+c_{44}}\,, \tag{B.24}$$

$$\tan(\phi_8) := -\frac{s_{48}}{c_{48}}\,. \tag{B.25}$$

The coefficients A_0, A_4 and ϕ_4 are functions of the cumulants of 3rd and 4th order of the centered and whitened signal $\mathbf{y}$. A_8 and ϕ_8 depend only on the 4th order cumulants.

B.1.3 Analytical Simplification of $\psi_{34}^{\mu\nu}$

The cumulants in $\mathbf{y}$ can always be written as a combination of the cumulants in $\mathbf{s}$

$$C_{ijkl}^{(\mathbf{y})} = \sum_{\alpha,\beta,\gamma,\delta=1,2} M_{i,\alpha} M_{j,\beta} M_{k,\gamma} M_{l,\delta} C_{\alpha\beta\gamma\delta}^{(\mathbf{s})}\,, \tag{B.26}$$

where $M_{i,j}$ are the entries of the rotation matrix defined in Equation (6.14). Since $\mathbf{s}$ has independent components this simplifies to

$$C_{ijkl}^{(\mathbf{y})} = \sum_{\alpha=1,2} M_{i,\alpha} M_{j,\alpha} M_{k,\alpha} M_{l,\alpha} C_{\alpha\alpha\alpha\alpha}^{(\mathbf{s})}\,. \tag{B.27}$$

Substituting this expression into the previous equation we arrive at the following relations

$$C_{111}^{(\mathbf{y})} = \cos(\theta)^3 C_{111}^{(\mathbf{s})} + \sin(\theta)^3 C_{222}^{(\mathbf{s})}\,, \tag{B.28}$$

$$C_{112}^{(\mathbf{y})} = -\cos(\theta)^2 \sin(\theta) C_{111}^{(\mathbf{s})} + \sin(\theta)^2 \cos(\theta) C_{222}^{(\mathbf{s})}\,, \tag{B.29}$$

$$C_{122}^{(\mathbf{y})} = \cos(\theta) \sin(\theta)^2 C_{111}^{(\mathbf{s})} + \sin(\theta) \cos(\theta)^2 C_{222}^{(\mathbf{s})}\,, \tag{B.30}$$

$$C_{222}^{(\mathbf{y})} = \sin(\theta)^3 C_{111}^{(\mathbf{s})} + \cos(\theta)^3 C_{222}^{(\mathbf{s})}\,, \tag{B.31}$$

$$C_{1111}^{(\mathbf{y})} = \cos(\theta)^4 C_{1111}^{(\mathbf{s})} + \sin(\theta)^4 C_{2222}^{(\mathbf{s})}\,, \tag{B.32}$$

$$C_{1112}^{(\mathbf{y})} = -\cos(\theta)^3 \sin(\theta) C_{1111}^{(\mathbf{s})} + \sin(\theta)^3 \cos(\theta) C_{2222}^{(\mathbf{s})}\,, \tag{B.33}$$

$$C_{1122}^{(\mathbf{y})} = \cos(\theta)^2 \sin(\theta)^2 C_{1111}^{(\mathbf{s})} + \sin(\theta)^2 \cos(\theta)^2 C_{2222}^{(\mathbf{s})}\,, \tag{B.34}$$

$$C_{1222}^{(\mathbf{y})} = -\cos(\theta) \sin(\theta)^3 C_{1111}^{(\mathbf{s})} + \sin(\theta) \cos(\theta)^3 C_{2222}^{(\mathbf{s})}\,, \tag{B.35}$$

$$C_{2222}^{(\mathbf{y})} = \sin(\theta)^4 C_{1111}^{(\mathbf{s})} + \cos(\theta)^4 C_{2222}^{(\mathbf{s})}\,. \tag{B.36}$$

The constants $c_{34}, c_{44}, c_{48}, s_{34}, s_{44}$ and s_{48} in (B.10) now become dependent on the rotation angle θ and the cumulants of the source signal. They are

$$
\begin{aligned}
a_{30} &= \frac{1}{3!}\frac{5}{8}\left(\left(C^{(\mathbf{s})}_{111}\right)^2+\left(C^{(\mathbf{s})}_{222}\right)^2\right), && \text{(B.37)}\\
a_{40} &= \frac{1}{4!}\frac{1}{64}\left(3\left(C^{(\mathbf{s})}_{1111}+C^{(\mathbf{s})}_{2222}\right)^2+32\left(\left(C^{(\mathbf{s})}_{1111}\right)^2+\left(C^{(\mathbf{s})}_{2222}\right)^2\right)\right), && \text{(B.38)}\\
c_{34} &= \frac{1}{3!}\frac{3}{8}\left(\left(C^{(\mathbf{s})}_{111}\right)^2+\left(C^{(\mathbf{s})}_{222}\right)^2\right)\cos(4\theta) && \text{(B.39)}\\
&:= e_{34}\cos(4\theta), && \text{(B.40)}\\
s_{34} &= -\frac{1}{3!}\frac{3}{8}\left(\left(C^{(\mathbf{s})}_{111}\right)^2+\left(C^{(\mathbf{s})}_{222}\right)^2\right)\sin(4\theta) && \text{(B.41)}\\
&= -e_{34}\sin(4\theta), && \text{(B.42)}\\
c_{44} &= \frac{1}{4!}\frac{1}{16}\left(7\left(C^{(\mathbf{s})}_{1111}\right)^2-2C^{(\mathbf{s})}_{1111}C^{(\mathbf{s})}_{2222}+7\left(C^{(\mathbf{s})}_{2222}\right)^2\right)\cos(4\theta) && \text{(B.43)}\\
&:= e_{44}\cos(4\theta), && \text{(B.44)}\\
s_{44} &= -\frac{1}{4!}\frac{1}{16}\left(7\left(C^{(\mathbf{s})}_{1111}\right)^2-2C^{(\mathbf{s})}_{1111}C^{(\mathbf{s})}_{2222}+7\left(C^{(\mathbf{s})}_{2222}\right)^2\right)\sin(4\theta) && \text{(B.45)}\\
&= -e_{44}\sin(4\theta), && \text{(B.46)}\\
c_{48} &= \frac{1}{4!}\frac{1}{64}\left(C^{(\mathbf{s})}_{1111}+C^{(\mathbf{s})}_{2222}\right)^2\cos(8\theta) && \text{(B.47)}\\
&:= e_{48}\cos(8\theta), && \text{(B.48)}\\
s_{48} &= -\frac{1}{4!}\frac{1}{64}\left(C^{(\mathbf{s})}_{1111}+C^{(\mathbf{s})}_{2222}\right)^2\sin(8\theta) && \text{(B.49)}\\
&= -c_{48}\sin(8\theta). && \text{(B.50)}\\
& && \text{(B.51)}
\end{aligned}
$$

Due to these dependencies, the contrast function (6.10) is therefore additionally a function of θ. The amplitudes A_4 and A_8 in (6.10) are given by (cf. (B.21)-(B.23))

$$
\begin{aligned}
A_0 &= a_{30}+a_{40}, && \text{(B.52)}\\
A_4 &= \sqrt{(e_{34}\cos(4\theta)+e_{44}\cos(4\theta))^2+(-e_{34}\sin(4\theta)-e_{44}\sin(4\theta))^2}\\
&= \sqrt{2(e_{34}+e_{44})^2}, && \text{(B.53)}\\
A_8 &= \sqrt{(e_{48}\cos(8\theta))^2+(-e_{48}\sin(8\theta))^2}\\
&= \sqrt{e_{48}^2}. && \text{(B.54)}
\end{aligned}
$$

Thus, the amplitudes are independent of θ and remain constant. The phases ϕ_4 and ϕ_8 are (cf. (B.24)-(B.25))

$$
\begin{aligned}
-\tan(\phi_4) &= \frac{s_{34}+s_{44}}{c_{34}+c_{44}} = -\frac{e_{34}\sin(4\theta)+e_{44}\sin(4\theta)}{e_{34}\cos(4\theta)+e_{44}\cos(4\theta)} = -\tan(4\theta), && \text{(B.55)}\\
-\tan(\phi_8) &= \frac{s_{48}}{c_{48}} = -\frac{e_{48}\sin(8\theta)}{e_{48}\cos(8\theta)} = -\tan(8\theta), && \text{(B.56)}
\end{aligned}
$$

resulting in the phase-relations

$$
\begin{aligned}
\phi_4 &= 4\theta, && \text{(B.57)}\\
\phi_8 &= 8\theta. && \text{(B.58)}
\end{aligned}
$$

B.2 Constants in CuBICA2

B.2.1 Constants in Equation (7.10)

Starting from the definition of the objective function

$$\Psi_2 = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \left(C_{ij}^{(\mathbf{u})}(\tau) \right)^2 , \tag{B.59}$$

we can derive ($N = 2$)

$$\begin{aligned} \Psi_2 = & \left[\left(\frac{3}{4} d_{20}(\tau) + \frac{1}{4} d_{22}(\tau) \right) \right. \\ & \left. + \left(\frac{1}{4} d_{20}(\tau) - \frac{1}{4} d_{22}(\tau) \right) \cos(4\phi) + \frac{1}{4} d_{21}(\tau) \sin(4\phi) \right] , \end{aligned} \tag{B.60}$$

with constants defined by (we drop the reference to τ to make the equations easy to read)

$$d_{20} := \left(2 \left(C_{12}^{(\mathbf{y})} \right)^2 \right) , \tag{B.61}$$

$$d_{21} := 4 \left(C_{12}^{(\mathbf{y})} C_{22}^{(\mathbf{y})} - C_{12}^{(\mathbf{y})} C_{11}^{(\mathbf{y})} \right) , \tag{B.62}$$

$$d_{22} := 2 \left(\left(C_{11}^{(\mathbf{y})} \right)^2 - 2 C_{11}^{(\mathbf{y})} C_{22}^{(\mathbf{y})} + {C_{22}^{(\mathbf{y})}}^2 \right) . \tag{B.63}$$

Equation (B.60) can be further simplified to obtain

$$\Psi_2 = a_{20} + s_{24} \sin(4\phi) + c_{24} \cos(4\phi) , \tag{B.64}$$

with constants

$$a_{20} = \frac{1}{4} (3 d_{20} + d_{22}) , \tag{B.65}$$

$$s_{24} = \frac{1}{4} d_{21} , \tag{B.66}$$

$$c_{24} = \frac{1}{4} (d_{20} - d_{22}) . \tag{B.67}$$

Further simplification leads to

$$\Psi_2 = A_{0\tau} + A_{4\tau} \cos(4\phi + \phi_{4\tau}) , \tag{B.68}$$

with constants

$$A_{0\tau} := a_{20} , \tag{B.69}$$

$$A_{4\tau} := \sqrt{c_{24}^2 + s_{24}^2} , \tag{B.70}$$

$$\tan(\phi_{4\tau}) := -\frac{s_{24}}{c_{24}} . \tag{B.71}$$

Note that these constants are all functions of τ.

B.2.2 Constants in Equation (7.16)

Using Equation (7.15) and some trigonometrics we can easily derive

$$\bar{A}_{0\tau} = \sum_{\tau=1}^{T} A_{0\tau}\,, \tag{B.72}$$

$$\bar{A}_{4\tau} = \sqrt{\left(\sum_{\tau=1}^{T} A_{4\tau}\sin\left(\phi_{4\tau}\right)\right)^2 + \left(\sum_{\tau=1}^{T} A_{4\tau}\cos\left(\phi_{4\tau}\right)\right)^2}\,, \tag{B.73}$$

$$\tan\left(\bar{\phi}_{4\tau}\right) = \frac{\sum_{\tau=1}^{T} A_{4\tau}\sin\left(\phi_{4\tau}\right)}{\sum_{\tau=1}^{T} A_{4\tau}\cos\left(\phi_{4\tau}\right)}\,. \tag{B.74}$$

C

Constants in Linear SFA with Higher Derivatives

C.1 Approximating Higher Derivatives of $\mathbf{y}(t)$

The difference quotients belonging to the first four derivatives (we also denote here the original signal as the 'zeroth' derivative) can be written as

$$\mathbf{y}(t)^{(0)} = \mathbf{y}(t), \tag{C.1}$$

$$\mathbf{y}(t)^{(1)} \approx \mathbf{y}(t+1) - \mathbf{y}(t), \tag{C.2}$$

$$\mathbf{y}(t)^{(2)} \approx \mathbf{y}(t+2) - 2\mathbf{y}(t+1) + \mathbf{y}(t), \tag{C.3}$$

$$\mathbf{y}(t)^{(3)} \approx \mathbf{y}(t+3) - 3\mathbf{y}(t+2) + 3\mathbf{y}(t+1) - \mathbf{y}(t), \tag{C.4}$$

$$\mathbf{y}(t)^{(4)} \approx \mathbf{y}(t+4) - 4\mathbf{y}(t+3) + 6\mathbf{y}(t+2) - 4\mathbf{y}(t+1) + \mathbf{y}(t), \tag{C.5}$$

where the denominator is set to one.

C.2 Computing Constants $\beta_{n\tau}$

Using the difference quotients defined in (C.1 - C.5) we can calculate the average of the square of these quotients as

$$\left\langle \mathbf{y}(t)^{(0)} \left(\mathbf{y}(t)^{(0)}\right)^T \right\rangle = \frac{1}{2}\left\langle \mathbf{y}(t)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t)^T \right\rangle, \tag{C.6}$$

$$\left\langle \mathbf{y}(t)^{(1)} \left(\mathbf{y}(t)^{(1)}\right)^T \right\rangle \approx \left\langle \mathbf{y}(t)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t)^T \right\rangle - \left\langle \mathbf{y}(t+1)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+1)^T \right\rangle, \tag{C.7}$$

$$\begin{aligned}\left\langle \mathbf{y}(t)^{(2)} \left(\mathbf{y}(t)^{(2)}\right)^T \right\rangle \approx\ & 3\left\langle \mathbf{y}(t)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t)^T \right\rangle - 4\left\langle \mathbf{y}(t+1)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+1)^T \right\rangle \\ & + \left\langle \mathbf{y}(t+2)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+2)^T \right\rangle,\end{aligned} \tag{C.8}$$

$$\begin{aligned}\left\langle \mathbf{y}(t)^{(3)} \left(\mathbf{y}(t)^{(3)}\right)^T \right\rangle \approx\ & 10\left\langle \mathbf{y}(t)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t)^T \right\rangle - 15\left\langle \mathbf{y}(t+1)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+1)^T \right\rangle \\ & + 6\left\langle \mathbf{y}(t+2)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+2)^T \right\rangle \\ & - \left\langle \mathbf{y}(t+3)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+3)^T \right\rangle,\end{aligned} \tag{C.9}$$

$$\begin{aligned}\left\langle \mathbf{y}(t)^{(4)} \left(\mathbf{y}(t)^{(4)}\right)^T \right\rangle \approx\ & 35\left\langle \mathbf{y}(t)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t)^T \right\rangle - 56\left\langle \mathbf{y}(t+1)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+1)^T \right\rangle \\ & + 28\left\langle \mathbf{y}(t+2)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+2)^T \right\rangle - 8\left\langle \mathbf{y}(t+3)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+3)^T \right\rangle + \\ & \left\langle \mathbf{y}(t+4)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+4)^T \right\rangle,\end{aligned} \tag{C.10}$$

where terms like $\left\langle \mathbf{y}(t+v)\mathbf{y}(t+w)^T + \mathbf{y}(t+w)\mathbf{y}(t+v)^T \right\rangle$ with $w \geq v$ are shifted to $\left\langle \mathbf{y}(t+v-w)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+v-w)^T \right\rangle$ w.l.o.g. We are now able to rewrite $\Delta_n(u_i)$ as

$$\begin{aligned}\Delta_n(u_i) &:= \left\langle \left(u_i^{(n)}\right)^2 \right\rangle, \\ &= \left[\mathbf{q}_i^T \left\langle \mathbf{y}^{(n)} \left(\mathbf{y}^{(n)}\right)^T \right\rangle \mathbf{q}_i\right], \\ &\approx \sum_{\tau=0}^{n} \beta_{n\tau}\left[\mathbf{q}_i^T \left\langle \mathbf{y}(t+\tau)\mathbf{y}(t)^T + \mathbf{y}(t)\mathbf{y}(t+\tau)^T \right\rangle \mathbf{q}_i\right], \\ &= \sum_{\tau=0}^{n} \beta_{n\tau} C_{ii}^{(\mathbf{u})}(\tau),\end{aligned} \tag{C.11}$$

where the constants $\beta_{n\tau}$ can be taken directly from (C.6 - C.10). Putting all $\Delta_n(u_i)$ together results in

$$\Sigma(u_i) := \sum_{n=0}^{T} \alpha_n \Delta_n(u_i) \approx \sum_{n=0}^{T} \alpha_n \sum_{\tau=0}^{n} \beta_{n\tau} C_{ii}^{(\mathbf{u})}(\tau) = \sum_{\tau=0}^{T} \delta_\tau C_{ii}^{(\mathbf{u})}(\tau), \tag{C.12}$$

where T is the number of derivatives taken into account. The factors $\beta_{n\tau}$ are shown in Table 8.0(a).

C.3 Computing Constants δ_τ

The constants δ_τ are calculated according to

$$\delta_\tau = \sum_{n=\tau}^{T} \alpha_n \beta_{n\tau}. \tag{C.13}$$

The first δ_τ up to $T = 4$ are listed in Table 8.0(b) where all derivatives are weighted equally ($\alpha_n = 1 \, \forall \, n \in \{1, 2 \ldots, T\}$).

D

Constants in ISFA

D.1 Constants in Equations (9.10) and (9.11)

The definitions of the constants d_n and e_n of Case 1 (9.8) and 2 (9.9) follow directly from the multilinearity of $C_{\cdots}^{(\mathbf{u})}(\tau)$. They are given in Table D.1. Using trigonometrics we can derive simpler objective functions as in (9.8) and (9.9) of the form

$$\text{Case 1:} \quad \Psi_{\text{ISFA}}^{\mu\nu}(\phi,\tau=1) = a_{20}+c_{24}\cos(4\phi)+s_{24}\sin(4\phi)\,, \tag{D.1}$$

$$\text{Case 2:} \quad \Psi_{\text{ISFA}}^{\mu\nu}(\phi,\tau=1) = a_{20}+c_{22}\cos(2\phi)+s_{22}\sin(2\phi)+c_{24}\cos(4\phi)+s_{24}\sin(4\phi)\,, \tag{D.2}$$

with constants defined in Table D.2. In the next step these objective functions are further simplified by putting the sine term and cosine term together in a single cosine term. This results in

$$\text{Case 1:} \quad \Psi_{\text{ISFA}}^{\mu\nu}(\phi,\tau=1) = \tilde{A}_0+\tilde{A}_4\cos\left(4\phi+\tilde{\phi}_4\right)\,, \tag{D.3}$$

$$\text{Case 2:} \quad \Psi_{\text{ISFA}}^{\mu\nu}(\phi,\tau=1) = \tilde{A}_0+\tilde{A}_2\cos\left(2\phi+\tilde{\phi}_2\right)+\tilde{A}_4\cos\left(4\phi+\tilde{\phi}_4\right)\,, \tag{D.4}$$

with constants defined in Table D.3.

Up to now we only considered a single time delayed correlation matrix. We may also want to use more than one time delay and possibly also give different weight to correlation matrices with different time delays. Note, that all constants from the objective function (D.3) and (D.4) are all depending on τ. The weight for correlation matrix with time delay τ is given by κ_τ. This results in an objective

$$\text{Case 1:} \quad \Psi_{ISFA}^{\mu\nu}(\phi,K) = \sum_{\tau=1}^{K}\kappa_\tau\Psi(\phi,\tau)=\sum_{\tau=1}^{K}\kappa_\tau\left[\tilde{A}_0(\tau)+\tilde{A}_4(\tau)\cos\left(4\phi+\tilde{\phi}_4(\tau)\right)\right] \tag{D.5}$$

$$= A_0+A_4\cos(4\phi+\phi_4)\,, \tag{D.6}$$

$$\text{Case 2:} \quad \Psi_{ISFA}^{\mu\nu}(\phi,K) = \sum_{\tau=1}^{K}\kappa_\tau\Psi(\phi,\tau)=\sum_{\tau=1}^{K}\kappa_\tau\left[\tilde{A}_0(\tau)+\tilde{A}_2(\tau)\cos\left(2\phi+\tilde{\phi}_2(\tau)\right)\right]+$$
$$\sum_{\tau=1}^{K}\kappa_\tau\left(\tilde{A}_4(\tau)\cos\left(4\phi+\tilde{\phi}_4(\tau)\right)\right) \tag{D.7}$$

$$= A_0+A_2\cos(2\phi+\phi_2)+A_4\cos(4\phi+\phi_4)\,, \tag{D.8}$$

where K is the maximal time delay. The constants are defined in Table D.4.

	Case 1	Case 2
d_0	$2b_{\mathrm{ICA}}\left(C^{(\mathbf{y})}_{\mu\nu}\right)^2 - b_{\mathrm{SFA}}\left(\left(C^{(\mathbf{y})}_{\mu\mu}\right)^2 + \left(C^{(\mathbf{y})}_{\nu\nu}\right)^2\right)$	$-b_{\mathrm{SFA}}\left(C^{(\mathbf{y})}_{\mu\mu}\right)^2$
d_1	$b_{\mathrm{ICA}}\left(4\left(C^{(\mathbf{y})}_{\mu\nu}C^{(\mathbf{y})}_{\nu\nu} - C^{(\mathbf{y})}_{\mu\mu}C^{(\mathbf{y})}_{\mu\nu}\right)\right) - b_{\mathrm{SFA}}\left(4\left(C^{(\mathbf{y})}_{\mu\mu}C^{(\mathbf{y})}_{\mu\nu} - C^{(\mathbf{y})}_{\mu\nu}C^{(\mathbf{y})}_{\nu\nu}\right)\right)$	$-4b_{\mathrm{SFA}}C^{(\mathbf{y})}_{\mu\nu}C^{(\mathbf{y})}_{\mu\mu}$
d_2	$b_{\mathrm{ICA}}\left(\left(C^{(\mathbf{y})}_{\mu\mu} - C^{(\mathbf{y})}_{\nu\nu}\right)^2 - 2\left(C^{(\mathbf{y})}_{\mu\nu}\right)^2\right) - b_{\mathrm{SFA}}\left(2\left(2\left(C^{(\mathbf{y})}_{\mu\nu}\right)^2 + C^{(\mathbf{y})}_{\mu\mu}C^{(\mathbf{y})}_{\nu\nu}\right)\right)$	$-b_{\mathrm{SFA}}\left(2\left(2\left(C^{(\mathbf{y})}_{\mu\nu}\right)^2 + C^{(\mathbf{y})}_{\mu\mu}C^{(\mathbf{y})}_{\nu\nu}\right)\right)$
d_3		$-4b_{\mathrm{SFA}}C^{(\mathbf{y})}_{\mu\nu}C^{(\mathbf{y})}_{\nu\nu}$
d_4	-	$-b_{\mathrm{SFA}}\left(C^{(\mathbf{y})}_{\nu\nu}\right)^2$
d_c	$b_{\mathrm{ICA}}\left(2\left(\sum_{\alpha=1}^{R-1}\sum_{\beta>\alpha}^{R}\left(C^{(\mathbf{y})}_{\alpha\beta}\right)^2 - \left(C^{(\mathbf{y})}_{\mu\nu}\right)^2\right)\right) - b_{\mathrm{SFA}}\sum^{R}_{\substack{\alpha=1,\\ \alpha\notin\{\mu,\nu\}}}\left(C^{(\mathbf{y})}_{\alpha\alpha}\right)^2$	$-b_{\mathrm{SFA}}\sum_{\substack{\alpha=1\\ \alpha\neq\mu}}^{R}\left(C^{(\mathbf{y})}_{\alpha\alpha}\right)^2$
e_0	-	$2b_{\mathrm{ICA}}\sum_{\substack{\alpha=1\\ \alpha\neq\mu}}^{R}\left(C^{(\mathbf{y})}_{\mu\alpha}\right)^2$
e_1	-	$4b_{\mathrm{ICA}}\sum_{\substack{\alpha=1\\ \alpha\neq\mu}}^{R}C^{(\mathbf{y})}_{\mu\alpha}C^{(\mathbf{y})}_{\alpha\nu}$
e_2	-	$2b_{\mathrm{ICA}}\sum_{\substack{\alpha=1\\ \alpha\neq\mu}}^{R}\left(C^{(\mathbf{y})}_{\alpha\nu}\right)^2$
e_c	-	$2b_{\mathrm{ICA}}\sum_{\substack{\alpha=1,\\ \alpha\neq\mu}}^{R-1}\sum_{\substack{\beta=\alpha+1,\\ \beta\neq\mu}}^{R}\left(C^{(\mathbf{y})}_{\alpha\beta}\right)^2$

Table D.1: Constants in Equation (9.8) and (9.9).

	Case 1	Case 2
a_{20}	$\frac{1}{4}(4d_3+d_2+3d_0)$	$\frac{1}{8}(3d_0+d_2+3d_4+8(d_c+e_c)+4(e_0+e_2))$
c_{22}	-	$\frac{1}{2}(d_0-d_4+e_0-e_2)$
s_{22}	-	$\frac{1}{4}(d_1+d_3-2e_2)$
c_{24}	$\frac{1}{4}(d_0-d_2)$	$\frac{1}{8}(d_0+d_4-d_2)$
s_{24}	$\frac{1}{4}d_1$	$\frac{1}{8}(d_1-d_3)$

Table D.2: Constants in Equation (9.8) and (9.9) as a function of the d_i and e_i.

	Case 1	Case 2
$\tilde{A}_0$	a_{20}	a_{20}
$\tilde{A}_2$	-	$\sqrt{c_{22}^2+s_{22}^2}$
$\tilde{A}_4$	$\sqrt{c_{24}^2+s_{24}^2}$	$\sqrt{c_{24}^2+s_{24}^2}$
$\tan(\tilde{\phi}_2)$	-	$-\frac{s_{22}}{c_{22}}$
$\tan(\tilde{\phi}_4)$	$-\frac{s_{24}}{c_{24}}$	$-\frac{s_{24}}{c_{24}}$

Table D.3: Constants of the further simplified objectives $\Psi^{\mu\nu}$.

	Case 1	Case 2
A_0	$\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_0(\tau)$	
A_2	-	$\sqrt{\left(\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_2(\tau)\cos(\tilde{\phi}_2(\tau))\right)^2+\left(\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_2(\tau)\sin(\tilde{\phi}_2(\tau))\right)^2}$
A_4	$\sqrt{\left(\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_4(\tau)\cos(\tilde{\phi}_4(\tau))\right)^2+\left(\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_4(\tau)\sin(\tilde{\phi}_4(\tau))\right)^2}$	
$\tan(\phi_2)$	-	$\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_2(\tau)\sin(\tilde{\phi}_2(\tau))/\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_2(\tau)\cos(\tilde{\phi}_2(\tau))$
$\tan(\phi_4)$	$\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_4(\tau)\sin(\tilde{\phi}_4(\tau))/\sum_{\tau=1}^{K}\kappa_\tau\tilde{A}_4(\tau)\cos(\tilde{\phi}_4(\tau))$	

Table D.4: Constants for the objective $\Psi^{\mu\nu}$ with more than one time delay.

Bibliography

S. A. Abdallah and M. D. Plumbley. An independent component analysis approach to automatic music transcription. In *Proc. of the 114th AES Convention, Amserdam, Netherlands*, 2003.

T. Akuzawa. New fast factorization method for multivariate optimization and its realization as ICA algorithm. In *Proc. of the 3nd Int. Workshop on Independent Component Analysis and Blind Signal Separation, San Diego, (ICA 2001)*, pages 114–119, 2001.

L. Almeida. Linear and nonlinear ICA based on mutual information - the MISEP method. *Signal Processing*, 84(2):231–245, 2004. Special Issue on Independent Component Analysis and Beyond.

S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

S. Amari, A. Cichocki, and H. Yang. Recurrent neural networks for blind separation of sources. In *Proc. of the Int. Symposium on Nonlinear Theory and its Applications (NOLTA-95)*, pages 37–42, Las Vegas, USA, 1995.

A. Azzalini. A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12:171–178, 1985.

M. Babaie-Zadeh, C. Jutten, and K. Nayebi. A geometric approach for separating post-nonlinear mixtures. In *Proc. of the XI European Signal Processing Conference (EUSIPCO 2002)*, pages 11–14, 2002.

A.J. Bell and T.J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

Adel Belouchrani, Karim Abed Meraim, Jean-François Cardoso, and Éric Moulines. A blind source separation technique based on second order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–44, 1997.

P. Berkes. sfa-tk: Slow feature analysis toolkit for matlab (v.1.0.1). `http://itb.biologie.hu-berlin.de/~berkes/software/sfa-tk/sfa-tk.shtml`, 2003.

P. Berkes. Pattern recognition with slow feature analysis. manuscript in preparation, 2004.

Pietro Berkes and Laurenz Wiskott. Slow feature analysis yields a rich repertoire of complex-cell properties. Cognitive Sciences EPrint Archive (CogPrints) 2804, `http://cogprints.ecs.soton.ac.uk/archive/00002804/`, February 2003.

T. Blaschke and L. Wiskott. CuBICA: Independent component analysis by simultaneous third- and fourth-order cumulant diagonalization. *IEEE Transactions on Signal Processing*, 52(5):1250–1256, 2004.

T. Blaschke, L. Wiskott, and P. Berkes. What is the relation between independent component analysis and slow feature analysis? *(in preparation)*, 2004.

Tobias Blaschke and Laurenz Wiskott. An improved cumulant based method for independent component analysis. In José R. Dorronsoro, editor, *Proc. Int. Conference on Artificial Neural Networks - (ICANN'02)*, Lecture Notes in Computer Science, pages 1087–1093. Springer, 2002.

J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140: 362–370, 1993.

Jean-François Cardoso. The three easy routes to independent component analysis; contrasts and geometry. In *Proc. of the 3rd Int. Conference on Independent Component Analysis and Blind Source Separation, San Diego, (ICA 2001)*, 2001.

Jean-François Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1):161–164, 1996.

A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing*. John Wiley&Sons, New York, 2002.

P. Comon. Tensor diagonalization, a useful tool in signal processing. In M. Blanke and T. Soderstrom, editors, *IFAC-SYSID, 10th IFAC Symposium on System Identification*, volume 1, pages 77–82, Copenhagen, Denmark, 1994a.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994b. Special Issue on Higher-Order Statistics.

P. Comon. Independent component analysis, contrasts and convolutive mixtures. In *Second IMA Conference on Mathematics in Communications*, Lancaster, UK, 2002.

P. Comon. Contrasts, independent component analysis, and blind deconvolution. *Int. Journal Adapt. Control Sig. Proc*, 18(3):225–243, 2004.

P. Comon, C. Jutten, and J. Herault. Blind separation of sources, Part ii : Problems statement. *Signal Processing*, 24:11–20, 1991.

T. Cover and J. Thomas. *Elements of information theory*. Wiley, New York, 1991.

L. De Lathauwer. *Signal processing based on multilinear algebra*. PhD thesis, K.U. Leuven, 1997.

L. De Lathauwer, B. De Moor, and J. Vandewalle. Blind source separation by simultaneous third-order tensor diagonalization. In *Proc. of the 8th European Signal Processing Conference (EUSIPCO'96)*, pages 2089–2092, Trieste, Italy, 1996.

L. De Lathauwer, B. De Moor, and J. Vandewalle. Independent component analysis based on higher-order statistics only. In *In Proc. of the 8th IEEE SP Workshop on Statistical Signal and Array Processing (SSAP'96)*, pages 356–359, Corfu, Greece, 1995.

G. Deco and D. Obradovic. *An information-theoretic approach to neural computing*. Springer Series in Perspectives in Neural Computing. Springer, New York, 1996.

E. Doi, T. Inui, T.-W. Lee, T. Wachtler, and T. J. Sejnowski. Spatio-chromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural Computation*, 15:397–417, 2003.

M. Feng and K.D. Kammeyer. Suppression of Gaussian noise using cumulants: A quantitative analysis. In *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP-97)*, pages 3813–3816,, 1997.

Y. Feng, Y. Zhuang, and Y. Pan. Popular music retrieval by independent component analysis. In *Proc. of the Int. Conf. on Music Information Retrieval and Related Activities, Paris, France (ISMIR 2002)*, pages 281–282, 2002.

R.A. Fisher. Moments and product moments of sampling distributions. *Proc. of the London Mathematical Society*, 2:199–238, 1929.

S. Flockton, D. Yang, and G. Scruby. Performance surfaces of blind source separation algorithms. In *Proc. of the Int. Conference on Neural Information Processing '96, Hong Kong, (ICONIP'96)*, pages 1229–1234, 1996.

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

E. Gassiat. *Déconvolution Aveugle*. PhD thesis, Université de Paris Sud, 1988.

M. Girolami and C. Fyfe. Higher order cumulant maximisation using nonlinear hebbian and anti-hebbian learning for adaptive blind separation of source signals. In *Proc. of the IEEE/IEE Int. Workshop on Signal and Image Processing,IWSIP-96*, pages 141–144, 1996.

S. Harmeling, F. Meinecke, and K.-R. Müller. Injecting noise for analysing the stability of ICA components. *Signal Processing*, 84(2):255–266, 2004. Special Issue on Independent Component Analysis and Beyond.

S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller. Kernel-based nonlinear blind source separation. *Neural Computation*, 15:1089–1124, 2003.

J. Herault and C. Jutten. Space or time adaptive signal processing by neural networks model. In J. Denker, editor, *Proc. of the Int. Conference on Neural Networks for computing, AIP conf. proc. n° 151*, pages 206–211, 1986.

G.E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234, 1989.

S. Hosseini and C. Jutten. On the separability of nonlinear mixtures of temporally correlated sources. *IEEE Signal Processing Letters*, 10(2):43–46, 2003.

A. Hyvärinen. A family of fixed-point algorithms for independent component analysis. In *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 3917–3920, 1997.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen, P. O. Hoyer, and J. Hurri. Extensions of ICA as models of natural images and visual processing. In *Proc. of the 4th Int. Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, (ICA 2003)*, pages 963–974, 2003.

A. Hyvärinen, P.O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1525–1558, 2001a.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley&Sons, New York, 2001b.

A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

John Fitzgerald Kennedy Library, Boston. Sound excerpts from the speeches of president John F. Kennedy. `http://www.jfklibrary.org/speeches.htm`, 1996. Retrieved February 6th, 2002.

I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

T.P Jung, S. Makeig, C. Humphries, T-W. Lee, M. McKeown, V. Iragui, and T.J. Sejnowski. Removing electroencephalographic artifcats by blind source separation. *Psychophysiology*, 37:167–178, 2000.

C. Jutten and J. Herault. Blind separation of sources, Part i : An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

C. Jutten, J. Herault, and A. Guerin. *Artificial intelligence and cognitive sciences*, pages 231–248. Manchester Press, 1988.

C. Jutten and J. Karhunen. Advances in nonlinear blind source separation. In *Proc. of the 4th Int. Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, (ICA 2003)*, pages 245–256, 2003.

J. Karhunen and J. Joutsensalo. Representation and separation of signals using nonlinear PCA type learning, neural networks. *Neural Networks*, 7(1):113–127, 1994.

S. L. Lauritzen. Aspects of T. N. Thiele's contributions to statistics. *Bulletin of the International Statistical Institute*, 58:27–30, 1999.

T.-W. Lee. *Independent component analysis: theory and applications*. Kluwer Academic Publishers, 1998.

T.-W. Lee, M. Girolami, and T.J. Sejnowski. Independent component analysis using an extended Infomax algorithm for mixed sub-Gaussian and super-Gaussian sources. *Neural Computation*, 11(2):409–433, 1999.

Wolfram Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51–60, 2002. URL `http://bioinformatics.oupjournals.org/cgi/content/abstract/18/1/51`.

R. Linsker. An application of the principle of maximum information preservation to linear systems. In David S. Touretzky, editor, *Advances in neural information processing systems 1 (NIPS 1988)*, pages 186–194. Morgan-Kaufmann, 1989.

P. McCullagh. *Tensor methods in statistics*. Monographs on Statistics and Applied Probability. Chapmann and Hall, London, 1987.

M. McKeown, T.P. Jung, S. Makeig, G. Brown, S. Kindermann, T-W. Lee, and T.J. Sejnowski. Spatially independent activity patterns in functional magnetic resonance imaging data during the stroop color-naming task. *Proceedings of the National Academy of Sciences*, 95:803–810, 1998.

G. Mitchison. Removing time variation with the anti-hebbian differential synapse. *Neural Computation*, 3 (3):312–320, 1991.

L. Molgedey and G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994.

E. Moreau and N. Thirion-Moreau. Nonsymmetrical contrasts for sources separation. *IEEE Transactions on Signal Processing*, 47(8):2241–2252, 1999.

K.-R. Müller, P. Philips, and A. Ziehe. JADETD: Combining higher-order statistics and temporal information for blind source separation (with noise). In *Proc. of the Int. Workshop on Independent Component Analysis and Signal Separation, Aussois, France, (ICA 1999)*, pages 87–92, 1999.

D. Nuzillard and J.-M. Nuzillard. Second-order blind source separation in the Fourier space of data. *Signal Processing*, 83(3):627–631, March 2003.

B. Pearlmutter. 16 clips sampled from audio CDs. `http://snot.cs.unm.edu/~bap/cICA/clips-wav/`, 1996. Retrieved February 7th, 2002.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philisophical Magazine*, 2: 559–572, 1901.

Karl Pearson. Asymmetrical frequency curves. *Nature*, 48:615–616, 1893.

W. Rudin. *Real and Complex Analysis*. McGraw-Hill, Singapore, 1987. Third edition.

M. Scholz, S. Gatzek, A. Sterling, O. Fiehn, and J. Selbig. Metabolite fingerprinting: detecting biological features by independent component analysis. *Bioinformatics*, page 270, 2004. URL `http://bioinformatics.oupjournals.org/cgi/content/abstract/bth270v1`.

C. Shannon. A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423,623–656, 1948.

JV Stone. A learning rule for extracting spatio-temporal invariances. *Network*, 6(3):1–8, 1995.

JV Stone. Blind source separation using temporal predictability. *Neural Computation*, 13(7):1559–1574, 2001.

A. Taleb. A generic framework for blind source separation in structured nonlinear models. *IEEE Transactions on Signal Processing*, 50(8):1819–1830, 2002.

A. Taleb and C. Jutten. Nonlinear source separation: The post-nonlinear mixtures. In *Proc. European Symposium on Artificial Neural Networks, Bruges, Belgium*, pages 279–284, 1997.

A. Taleb and C. Jutten. Source separation in post non linear mixtures. *IEEE Transactions on Signal Processing*, 47(10):2807–2820, October 1999.

T.N. Thiele. *Forelæsninger over almindelig Iagttagelseslære: Sandsynlighedsregning og mindste Kvadraters Methode*. Reitzel, København, 1889.

T.N. Thiele. Theory of observations. *Annals of Mathematical Statistic*, 2:165–308, 1931.

Lang Tong, Ruey-wen Liu, Victor C. Soon, and Yih-Fang Huang. Indeterminacy and identifiability of blind identification. *IEEE Transactions on Circuits and Systems*, 38(5):499–509, may 1991.

D. L. Wallace. Asymptotic approximations to distributions. *Ann. Math. Statist.*, 29:635–654, 1958.

Laurenz Wiskott. Estimating driving forces of nonstationary time series with slow feature analysis. arXiv.org e-Print archive, http://arxiv.org/abs/cond-mat/0312317/, December 2003a.

Laurenz Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, September 2003b.

Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

H.-H Yang, S. Amari, and A. Cichocki. Information-theoretic approach to blind separation of sources in non-linear mixture. *Signal Processing*, 64(3):291–300, 1998.

D. Yellin and E. Weinstein. Multichannel signal separation: methods and analysis. *IEEE Transactions Signal on Processing*, 44(1):106–118, 1996.

A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions On Signal Processing*, 50(7):1545–1553, 2002.

M. Zibulevsky and B.A. Pearlmutter. Second order blind source separation by recursive splitting of signal subspaces. In *Proc. of the 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2000*, pages 489–491, Helsinki, Finland, 2000.

A. Ziehe, P. Laskov, K.-R. Müller, and G. Nolte. A linear least-squares algorithm for joint diagonalization. In *Proc. of the 4th Int. Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, (ICA 2003)*, pages 469–474, 2003a.

A. Ziehe and K.-R. Müller. TDSEP – an efficient algorithm for blind separation using time structure. In *Proc. of the 8th Int. Conference on Artificial Neural Networks (ICANN'98)*, pages 675 – 680, Berlin, 1998. Springer Verlag.

A. Ziehe, K.-R. Müller, G. Nolte, B.-M. Mackert, and G. Curio. Artifact reduction in magnetoneurography based on time-delayed second-order correlations. *IEEE Transactions on Biomedical Engineering*, 47(1): 75–78, 2000.

Andreas Ziehe, Motoaki Kawanabe, Stefan Harmeling, and Klaus-Robert Müller. Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation. *Journal of Machine Learning Research*, 4:1319–1338, 2003b.

A. Zurmühl and S. Falk. *Matrizen und ihre Anwendungen*, volume 1. Springer, Berlin, 6th edition, 1997.

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig ohne fremde Hilfe verfaßt und nur die ange gebene Literatur und Hilfsmittel verwendet zu haben.

Tobias Blaschke
25. August 2004