Stationary Subspace Analysis

Towards understanding non-stationary data

Paul von Bünau

Stationary Subspace Analysis

Towards understanding non-stationary data

vorgelegt von Paul von Bünau

Von der Fakultät IV — Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation

Promotionsausschuss

Vorsitzender: Prof. Dr. Olaf Hellwich (Technische Universität Berlin)
1. Gutachter: Prof. Dr. Klaus-Robert Müller (Technische Universität Berlin)
2. Gutachter: Prof. Dr. Gilles Blanchard (Universität Potsdam)
3. Gutachter: Prof. Dr. Benjamin Blankertz (Technische Universität Berlin)

Tag der wissenschaftlichen Aussprache: 12. September 2012

Berlin, 2012
D 83

Acknowledgements

Translated from the German, “Ph.D. advisor” becomes “doctor father”. This is quite literally what Klaus-Robert Müller was to me. For the most part, it was his inspirational enthusiasm that drew me into machine learning in the first place, and made me return after a two-year excursion. His kind encouragement and indefatigable optimism helped me not to despair during the dry periods of research, when everything seems pointless. But above all, the inimitable Müller determination not to take anything too seriously, and see the human side of everything, is what makes his research group such an enjoyable place. Thank you.

This thesis is the result of a collaboration with Frank C. Meinecke and Franz J. Kiraly. It has been a privilege, a pleasure, and an incredible learning experience for which I am deeply grateful.

My colleagues have been an invaluable source of scientific advice, camaraderie and amusement, both in and out of the office, and on our various joint conference trips. In particular, I would like to mention Mikio Braun, Stefan Haufe and Felix Bießmann.

I am indebted to Jan Saputra Müller and Duncan A. J. Blythe for the excellent joint work during the last years. Your support has been important in many ways.

My research would not have been possible without the tireless efforts of Andrea Gerdes, Dominik Kühne and Imke Weitkamp. Even though I can only fathom what most of it entails, I am well aware that it matters hugely.

The EEG data sets I used in this thesis were recorded by Claudia Sannelli, Thorsten Dickhaus, Sven Dähne and Johannes Höhne. This is hard work; your attention to detail is very much appreciated.

I am sincerely grateful for the frequent detours from academia with Sebastian Mika and Aldo Benini; to industry, and Bangladesh, respectively. Both have significantly delayed this thesis, but also provided the change in scenery on which I thrive.

Over the years, I have had the good fortune of meeting inspiring teachers who generously shared their knowledge with me. While this list is inevitably incomplete, it certainly includes Lothar Budach, Martyn Quick, Albrecht Klemm, Georg Hermann, and from the first, my parents.

But what matters most is that I have found you, Louise.

Financial support

The bulk of my research was supported by a teaching position at TU Berlin, funded by the federal state of Berlin. In addition, I have received substantial travel grants from the EU Network of Excellence PASCAL2 and the German Academic Exchange Foundation (DAAD). In 2008 and 2010, I was kindly invited to the Research-in-Pairs (RiP) programme of the Mathematisches Forschungsinstitut Oberwolfach (MFO), core-funded by the Leibniz Gemeinschaft and the state of Baden-Württemberg.

I am sincerely grateful to the European tax payer who provided this money, and I appreciate the trust that society puts in the academic community.

Abstract

This thesis is about statistical methods for understanding *change* in the *joint distribution* of observed multivariate data *over time*. The setting we consider is completely *explorative* or *unsupervised*: no auxiliary information regarding the distribution changes is available.

We propose the first *unsupervised* method, stationary subspace analysis (SSA), for finding a *linear coordinate transformation* which factorises the observed data into *stationary* and *non-stationary* components. This is essential because the relevant changes can occur in the *dependencies* between variables, which means that the input variables can be totally uninformative: in fact, the non-stationary and the stationary part can remain *completely invisible* in the observations. In practice, this is often the case when one can only measure superpositions of the actual variables of interest. For example, in EEG analysis, electrodes on the surface of the scalp record activity from several neural sources located inside the brain. As we show in this thesis, investigating changing behaviour in *brain sources* crucially depends on the separation of the stationary and non-stationary components in the recorded signals.

The second main contribution of this thesis is a novel approach to finding particular types of *approximative solutions* to systems of *polynomial equations* of arbitrary degree based on techniques from computational algebraic geometry. Using the concept of generic polynomials, we show how SSA can be formulated in this framework. This leads to a new algorithm which has a unique solution and is more accurate in certain cases. From a theoretical perspective, the most interesting feature of this approach is that it allows us to *solve* the problem algebraically instead of *searching* for the solution guided by an objective function. As the assumptions underpinning the algorithm are rather general, it may be directly applicable to other problems in machine learning whose solution can be formulated in terms of polynomial equations.

Zusammenfassung

Das Thema dieser Dissertation sind statistische Methoden für das Verständnis *zeitlicher Veränderung der gemeinsamen Verteilung multivariater Daten*. Wir betrachten den sogenannten *explorativen Fall*, in dem keinerlei zusätzlichen Informationen, z.B. über relevante Zeitpunkte oder einen kontrollierten Stimulus, verfügbar sind.

Der zentrale Beitrag dieser Arbeit ist die Entwicklung des ersten unüberwachten Verfahrens, Stationary Subspace Analysis (SSA), welches eine linear Koordinatentransformation findet das die beobachteten Daten in eine Gruppe von stationären und nicht-stationären Komponenten faktorisiert. Dies ist unerlässlich zur Analyse multivariater Daten, weil die wesentlichen Änderungen der gemeinsamen Verteilung die Abhängigkeiten zwischen den Variablen betreffen können. Daher erlaubt die univariate Betrachtung der Eingangsgrössen nicht notwendigerweise Rückschlüsse auf Änderungen der gemeinsamen Verteilung. Sowohl die nicht-stationären als auch die stationären Komponenten können nämlich in den beobachten Variablen vollständig unsichtbar bleiben. Dies ist vor allem dann der Fall, wenn die messbaren Grössen Überlagerungen der tatsächlich relevanten Variablen sind, welche nicht direkt gemessen werden können. Ein gutes Beispiel liefert die EEG Datenanalyse: die Elektroden auf der Kopfhaut messen die Beiträge einer Vielzahl neuronaler Quellen im Gehirn. Um die zeitliche Veränderung der Verteilung dieser Quellen zu verstehen ist es daher notwendig, die stationären von den nicht-stationären Signalanteilen zu trennen. In einer Anwendung auf EEG Daten zeigen wir zum Einen, dass SSA dies leistet und zum Anderen, dass die populären Koordinatentransformationen Principal Component Analysis (PCA) und Independent Component Analysis (ICA) dazu nicht in der Lage sind.

Der zweite wesentliche Beitrag dieser Arbeit ist ein neuartiger Ansatz zur approximativen Lösung polynomieller Gleichungssystem eines bestimmten Typs. Aufbauend auf dem Konzept der generischen Polynome zeigen wir, das SSA als ein solches Problem formuliert werden kann. Dies führt zu einem neuen Algorithmus dessen Lösung nicht nur eindeutig, sondern in Spezialfällen auch exakter ist. Von einem theoretischen Standpunkt aus gesehen ist es besonders interessant, dass es dieser Ansatz erlaubt das SSA Problem direkt *algebraisch zu lösen*, anstatt wie in einem Optimierungsverfahren nach der Lösung zu *suchen*. Da die zugrunde liegenden Annahmen eher allgemein sind, lässt sich der Algorithmus möglicherweise di-

rekt auf andere Probleme im Maschinellen Lernen anwenden, die als Lösung polynomieller Gleichungen formuliert werden können.

Contents

Chapter 1

Introduction

1.1 Why this thesis?

This thesis is about methods for the analysis of data. The starting point is a simple question: given a set of *multivariate samples* observed at different points in time, how does the *joint distribution* change over time? The setting is completely *unsupervised* or *explorative*: apart from the data, no further information or prior knowledge which could be helpful in identifying the changes is available.

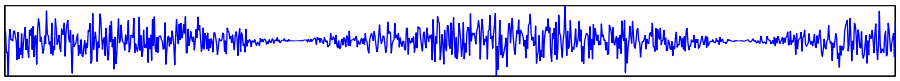

Figure 1.1: A change in variance over time.

In such a situation, a common approach is to resort to a visual inspection of the data. Although unwieldy for large data sets, plotting the time course of a single variable should, in principle, provide us with a good indication of its change in behaviour over time (see e.g. Figure 1.1).

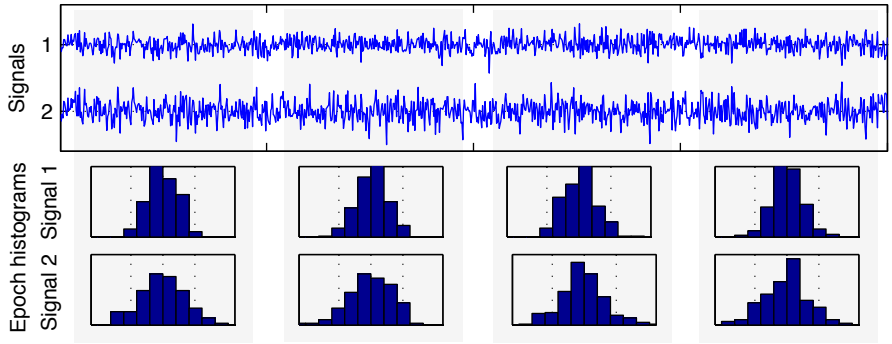

Figure 1.2: Does this joint distribution change over time?

However, this approach fails as soon as we are dealing with the *joint distribution* of more than one variable. Let us consider the example shown in Figure 1.2. In ad-

dition to the time courses, this plot shows the distribution of the individual variables as histograms in four segments. Apart from the fluctuations due to finite samples, this does not reveal any significant changes in distribution.

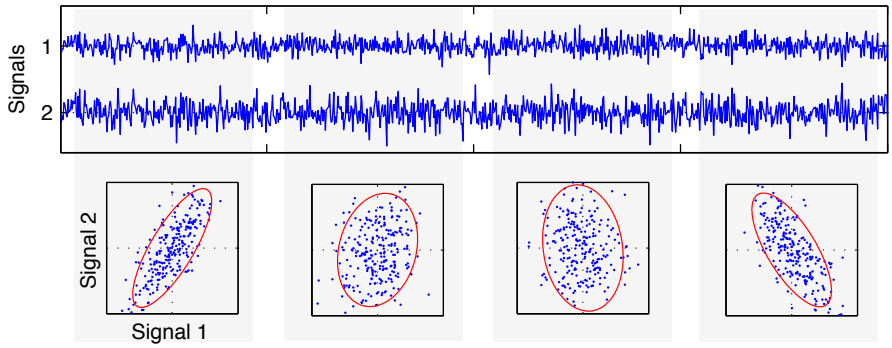

Figure 1.3: The change is confined to the dependencies!

Figure 1.3 provides a different view of the same data set. From the scatter plots, we immediately see what has been missed: the *dependencies* between the two variables. The correlation between the variables changes completely over time, from highly correlated at the beginning, to uncorrelated, to highly anticorrelated towards the end. Looking at variables individually, i.e. the *marginals* of the joint distribution, does not give us any information about the dependencies between them.

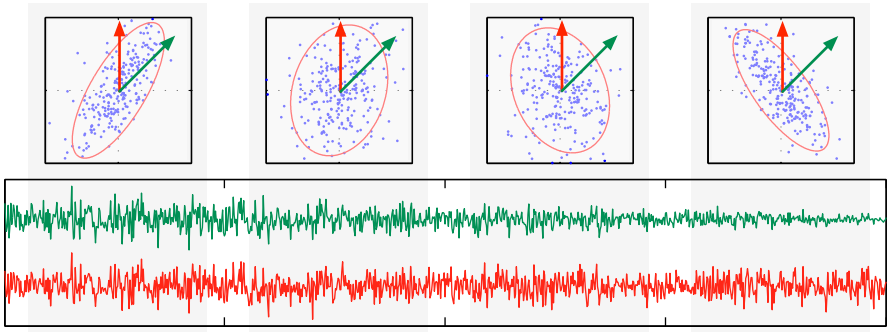

Figure 1.4: More useful directions

In light of this finding, a more useful way of looking at this data set is given by the red and green directions shown in Figure 1.4. By projecting the data points onto these axes, we obtain two new variables with time courses shown in the panel below. The first one corresponds to the green direction. As we can see in the scatter plots, along this direction the change in the joint distribution is clearly visible: from large variance in the first epoch to small variance in the last epoch; this is what we call a *non-stationary* direction. The second variable (red) is a *stationary* variable, because

its distribution stays constant[1]. The projection onto these two directions achieves a separation of the stationary and the non-stationary content of the joint distribution.

This example was benign in that it consisted of only two variables. For higher dimensional data sets, analysing dependencies by looking at scatter plots between all pairs of variables (over a potentially large number of epochs) is not merely cumbersome, but simply infeasible. Moreover, we have no labels or target variables to guide our search. Thus the question posed at the outset turns out to be surprisingly difficult to answer. More importantly, this problem is not only non-trivial, but also *relevant* in practice. Understanding temporal changes lies at the heart of many scientific inquiries. Real data contains significant amount of information in its dependency structure, and is often high-dimensional.

In fact, in many circumstances it is well known that what we measure is a *superposition* of *underlying latent variables* which cannot be measured directly. This creates rich dependencies across dimensions. For example, in EEG analysis, electrodes on the surface of the scalp record contributions from a multitude of sources located inside the brain. Similarly, in the analysis of seismic activity, one is often confined to surface measurements for investigating the activity inside the earth. Thus each observed variable contains contributions from both the stationary, and the non-stationary parts of the underlying data generating system.

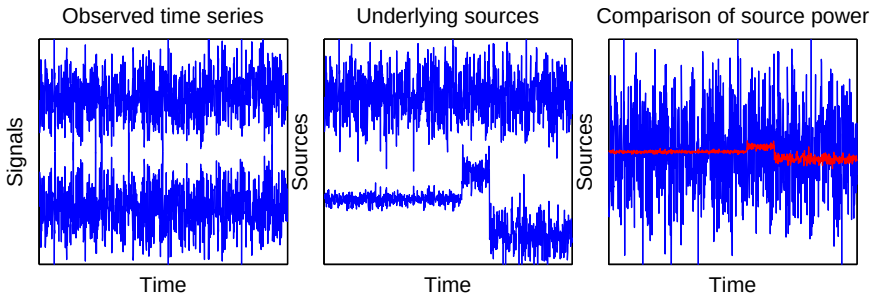

Figure 1.5: Non-stationary component masked by stronger stationary component.

As the example in Figure 1.3 showed, this could mean that the relevant distribution changes are entirely invisible in the measurements, because they are confined to the dependencies. Another way in which relevant distribution changes can go unnoticed is if they are masked by relatively stronger components, see Figure 1.5. The opposite effect is also possible: a *single* non-stationary factor affects all measurements, giving the false impression that the data as a whole is non-stationary. Discerning the stationary from the non-stationary components is, therefore, a key step towards understanding multiviariate non-stationary data. How to do this is the topic of this thesis.

[1] A precise notion of stationarity will be introduced in Chapter 3.

Our main contribution is a method for finding a *linear coordinate transformation*, stationary subspace analysis (SSA), which factorises the observed data into a set of stationary, and a set of non-stationary, sources. In the example shown in Figure 1.3, such a transformation would be given by projecting onto the green and red axes respectively. The type of distribution changes which our method detects are changes in the mean and the covariance matrix. Importantly, SSA does not require labels to specify the timing of the distribution changes (or any other form of supervision); and it is not limited to time *series* but can also be applied to sets of data collected at different time points.

A factorisation into stationary and non-stationary components is useful in many circumstances. As we have seen in the example, SSA is an essential tool for *explorative analysis*, because the observed variables can be arbitrarily ill suited for the analysis of changes in the joint distribution. Moreover, SSA is useful for *feature extraction* [Guyon and Elisseeff (2003)] when only the stationary or the non-stationary part is informative for the subsequent application. For example, high-dimensional change-point detection [Blythe et al. (2012)] can benefit from the prior removal of the uninformative stationary part. In some cases, SSA has been found to be useful for domain adaptation, by restricting parameter estimation of a prediction method to the stationary part [von Bünau et al. (2010), Hara et al. (2010)]. Finally, SSA can extract *meaningful components* when the observations are (modeled as) linear superpositions of the relevant latent variables, and non-stationarity is a sensible criterion for identifying the components of interest [Hara et al. (2012)]. In particular, SSA allows us to extract such components even when no prior knowledge about the timing of the relevant activity is available, e.g. when they are not induced by an experimental paradigm. This is the case e.g. in the analysis of spontaneous neural activity [Bießmann et al. (2011), Biswal et al. (1995)] or default mode networks [Raichle et al. (2001)].

Stationary subspace analysis is not the first method to exploit dependencies between variables to find a coordinate system which is more useful in some sense. However, none of the existing approaches are tailored to the understanding of distribution changes. As an illustration, let us compare the results of SSA with the solutions found by principal component analysis (PCA) and independent component analysis (ICA) on the data shown in Figure 1.3. In Figure 1.6 we see that both PCA and ICA produce uninformative results, albeit in different ways. In the PCA basis (top panel), the distribution changes remain completely invisible. ICA (middle panel), on the other hand, gives us two components that are both non-stationary: suddenly, the stationary components have become invisible. Only SSA achieves a clear separation into stationary and non-stationary components. In principle, this is to be expected. ICA maximises the statistical independence of the coordinates, which is unrelated to discerning the two groups of components. Similarly, the PCA solution is determined by the shape of the *overall* covariance matrix which is not related to non-stationarity.

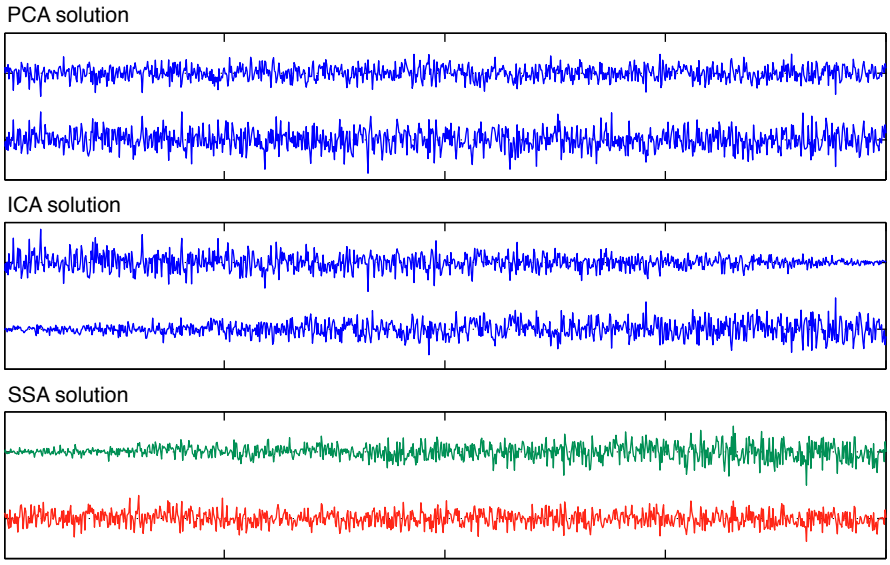

Figure 1.6: PCA and ICA do not separate the stationary and the non-stationary components.

The second main contribution of this thesis is a new framework for the finding approximate solutions (or a certain type) to systems of polynomial equation. We demonstrate that SSA can be formulated in terms of computing the basis of an algebraic variety given in terms of noisy polynomials estimated from data. Or, in other words, we show how to approximate the set of solutions to noisy polynomial equations by a linear space. What makes the SSA problem amenable to this type of formulation is that the projections to the stationary coordinates can be described as the set of linear projections which make the moments of the distribution *equal* in the new coordinates. These projections are therefore given by the solution to a set of polynomials with coefficients derived from moments estimated on the observed data. For example, the equality of the mean translates into linear polynomials and the equality of the covariance matrix is expressed by a quadratic polynomial. Not only does the algebraic approach have a better performance (higher accuracy) in certain scenarios, but this viewpoint also allows us to invoke powerful devices from algebraic geometry for reasoning about learning problems and algorithms.

From a theoretical perspective, the interesting distinction of the algebraic approach is that it allows us to directly solve a problem algebraically, which hitherto required an iterative search procedure using gradient-based optimisation. In some sense, this is due to the fact that in the algebraic formulation, we have explicitly incorporated all available information about the structure of the problem.

1.2 Scope, main contributions and publications

The main contributions of this thesis can be summarised as follows.

1. Motivation and formulation of a new type of unsupervised learning task, stationary subspace analysis.
2. Analysis of the generative model w.r.t. identifiability invariances and spurious solutions.
3. Derivation of a computationally efficient SSA algorithm, based on a custom optimisation procedure over the special orthogonal group.
4. Development of an approximate algebraic algorithm for solving the SSA problem based on an algebraic-geometric viewpoint.
5. Empirical evaluation of the SSA algorithm on synthetic data and in applications to EEG analysis.

1.2.1 Included publications

Parts of this thesis have been presented in the following publications. The results presented in Chapter 5 are unpublished.

- von Bünau, P., Meinecke F. C., Király, F. J., and Müller K.R. *Finding Stationary Subspaces in Multivariate Time Series.* **Phys. Rev. Lett.** 103 (21):214101, 2009.
- Müller, J. S., von Bünau, P., Meinecke F. C., Király, F. J., and Müller K.R. *The Stationary Subspace Analysis Toolbox.* **Journal of Machine Learning Research** 12:3065–3069, 2011.
- Király, F. J., von Bünau, P., Meinecke F. C., Blythe, D. A. J., and Müller K.R. *Algebraic Geometric Comparison of Probability Distributions.* **Journal of Machine Learning Research** 13:855–903, 2012.
- von Bünau, P., Meinecke F. C., Scholler, S., and Müller K.R. *Finding Stationary Brain Sources in EEG Data.* **Proceedings of the 32nd Annual Conference of the IEEE EMBS**, Buenos Aires, 2010.
- Király, F. J., von Bünau, P., Müller, J.S., Blythe, D. A. J., Meinecke F. C., and Müller K.R. *Regression for sets of polynomial equations.* **JMLR Workshop and Conference Proc. (AISTATS 2012)**, Vol. 22, 2012.

1.2.2 Related work not included

Since the publication of the SSA algorithm, the following applications and extensions have been presented which are not part of this thesis. The author of this thesis has been involved in some but not all of them.

Application to domain adaptation The authors show that SSA can be used as a pre-processing step to allow for the domain adaptation of classification algorithms in the context of WiFi localisation [Hara et al. (2010)].

Application to geophysical data analysis and an analytic algorithm In the simultaenous analysis of measurements relating to the dynamics of the earth's magnetic field, the relevant components are known to exhibit strong changes but cannot be assumed to be independent. The authors show that SSA recovers meaningful components where ICA fails. Moreover, a new SSA algorithm is presented that has an analytic solution based on certain assumptions on the latent variables [Hara et al. (2012)].

Application to computer vision The stationary components found by SSA can be interpreted as temporally stable relationships between variables. The authors adopt this viewpoint and demonstrate that it allows the extraction of invariances from image data [Meinecke et al. (2009)].

Application to change-point detection In high-dimensional change-point detection, only the non-stationary directions are informative while the rest contribute to the number of dimensions by adding irrelevant information which is potentially harmful to the performance. Therefore, the authors show that SSA can be used as a feature extraction method in extensive simulations and as an application to fault detection [Blythe et al. (2012)].

An information-theoretical view of SSA The authors show that SSA can be understood from an information-theoretical point of view in terms of a projection in the space of distributions; this leads to a new algorithm [Kawanabe et al. (2011)].

Group-Wise SSA and brain-computer-interfacing In the context of a classification setting, it is often desirable to identify only those non-stationarities that are not class-related. To this end, the authors propose a new algorithm called groupwise-SSA and demonstrate its effectiveness in an application to brain-computer-interfacing [Samek et al. (2012)].

1.3 Related areas in machine learning

With this thesis, we introduce a new family of unsupervised learning tasks. Even though there are no algorithms with exactly the same objective, SSA is related to a number of approaches by formal similarity, shared technical tools or a similar application context, which we briefly summarise below.

Two-sample testing Do two samples come from the same distribution? A two-sample test [Wald and Wolfowitz (1940), Smirnoff (1939)] is a statistical hypothesis test for deciding this problem The result of such tests is a *binary decision*; no more information about the type or extent of a difference in distribution can be gleaned from the result. Two-sample testing may be applied in similar circumstances than SSA, but addresses a fundamentally different question.

At the heart of a two-sample test lies a parametric or nonparametric distance measure between two distributions, and such a function also appears in the SSA objective. In fact, we show that the SSA objective function is equivalent to a certain type of two-sample test. Future work may explore the possibility of deriving high-order SSA algorithms by borrowing from non-parametric two-sample tests, e.g. following [Gretton et al. (2012)].

Domain and covariate shift adaptation The goal of domain and covariate shift adaptation is to improve the performance of prediction methods (classification or regression) in a scenario in which the test data follows a different distribution than the training data. One way of covariate shift adaptation is to weight the training data [Shimodaira (2000), Sugiyama et al. (2007), Sugiyama et al. (2008)] or regularise w.r.t. distribution changes [Schweikert et al. (2008)].

In contrast to SSA, the focus is on adapting a supervised learning to distribution changes between two sets of samples, and not on investigating the distribution change itself. However, it has been shown [Hara et al. (2010)] that for some applications, SSA can be used for domain adaptation by confining the learning to the directions between training and testing data which are stationary. Note that this need not be true in general: the stationary directions are not necessarily correlated with the label; and a certain degree of non-stationarity along strongly informative directions may not be harmful for prediction performance. How the possibly conflicting goals of stationarity and informativeness can be combined in a principled manner is an open question.

Moreover, SSA can be useful as a feature extraction method for density ratio estimation [Sugiyama (2009)], which is an essential step in the reweighting approach to covariate shift adaptation: only the non-stationary directions are relevant for density ratio estimation, and it has been shown [Sugiyama et al. (2011)] that the prior removal of the stationary directions can increase the accuracy of density ratio estimation and hence covariate shift adaptation.

PCA, ICA and factor analysis The generative model of PCA, ICA and factor analysis formally resembles the SSA model: the observed data is a linear mixture of underlying latent variables. This, however, is where the similarities end. ICA assumes that sources are independent; SSA merely presupposes that there are two groups of sources (stationary and non-stationary), which may have arbitrary dependence

structures. Consequently, ICA and SSA algorithms have entirely different goals: recovering maximally independent sources vs. maximizing resp. minimizing a multivariate stationarity measure. Similarly, PCA and factor analysis find orthogonal directions minimizing the reconstruction error which is unrelated to stationarity resp. non-stationarity. Also, whereas PCA, ICA and factor analysis compute a set of one-dimensional components, SSA finds two subspaces.

However, SSA is related to the family of blind source separation algorithms by shared technical tools (a pre-whitening step plus optimisation over the special orthogonal group) and similar terminology.

1.4 A roadmap through this document

Chapter 2 In this chapter, we introduce the the task of finding a coordinate transform of the data space with certain useful properties. Mathematical concepts are briefly reviewed and the two most prominent methods, PCA and ICA, are discussed in depth.

Chapter 3 Stationary subspace analysis is presented: after outlining and analysing the SSA model, we develop a computationally efficient algorithm, and discuss the problem of spurious stationarity and the relationship to stationarity testing.

Chapter 4 This chapter contains the second main contribution: we present an approximate algebraic algorithm for solving the SSA task and analyse its computational complexity.

Chapter 5 In extensive simulations on synthetic data, we investigate the performance of SSA relative to an artificial ground truth; in applications to EEG analysis we show that SSA provides useful new insights.

Chapter 6 The thesis concludes with a summary of the main findings, and a discussion of open problems and directions for future work.

List of all publications

The following is a list of all publications co-authored by the author of this thesis from 2008 until 2012.

Articles in journals

Hara, S., Kawahara, Y., Washio, T., von Bünau, P., Tokunaga, T. and Yumoto, K. *Separation of stationary and non-stationary sources with a generalized eigenvalue problem* **Neural Networks** 33:7–20, 2012.

Kiraly, F. J., von Bünau, P., Meinecke, F. C., Blythe, D. A. J. and Müller, K. R. *Algebraic Geometric Comparison of Probability Distributions* **Journal of Machine Learning Research** 13:855–903, 2012.

Blythe, D. A. J., von Bünau, P., Meinecke, F. C. and Müller, K. R. *Feature Extraction for Change-Point Detection using Stationary Subspace Analysis* **IEEE Transactions on Neural Networks and Learning Systems** 23 (4):631–643, 2012.

Sugiyama, M., Yamada, M., von Bünau, P., Suzuki, T., Kanamori, T. and Kawanabe, M. *Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search* **Neural Networks** 24 (2):183-198, 2011.

Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B. and Müller, K. R. *Toward Unsupervised Adaptation of LDA for Brain-Computer Interfaces* **IEEE Transactions on Biomedical Engineering** 58 (3):587–597, 2011.

Müller, J., von Bünau, P., Meinecke, F. C., Király, F. J. and Müller, K. R. *The Stationary Subspace Analysis Toolbox* **Journal of Machine Learning Research** 12:3065–3069, 2011.

Araujo, J., von Bünau, P., Mitchell, J. and Neunhoeffer, M. *Computing automorphisms of semigroups* **Journal of Symbolic Computation** 45 (3):373-392, 2010.

von Bünau, P., Meinecke, F. C., Király, F. J. and Müller, K. R. *Finding Stationary Subspaces in Multivariate Time Series* **Phys. Rev. Lett.** 103 (21):214101, 2009.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P. and Kawanabe, M. *Direct Importance Estimation for Covariate Shift Adaptation* **Annals of the Institute of Statistical Mathematics** 60 (4):699–746, 2008.

Peer-reviewed contributions to conferences

Kiraly, F. J., von Bünau, P., Müller, J. S., Blythe, D. A. J., Meinecke, F. C. and Müller, K. R. *Regression for sets of polynomial equations* **JMLR Workshop and Conference Proc. (AISTATS 2012)** 22, 2012.

Kawanabe, M., Samek, W., von Bünau, P. and Meinecke, F. C. *An Information Geometrical View of Stationary Subspace Analysis* **Artificial Neural Networks and Machine Learning (ICANN 2011), LNCS** 6792:397–404, Springer, 2011.

von Bünau, P., Meinecke, F. C., Scholler, S. and Müller, K. R. *Finding Stationary Brain Sources in EEG Data* **Proceedings of the 32nd Annual Conference of the IEEE EMBS**, 2010.

Hara, S., Kawahara, Y., Washio, T. and von Bünau, P. *Stationary subspace analysis as a generalized eigenvalue problem* **Proceedings of the 17th international conference on Neural information processing (ICONIP)**, 2010.

Sugiyama, M., Hara, S., von Bünau, P., Suzuki, T., Kanamori, T. and Kawanabe, M. *Direct Density Ratio Estimation with Dimensionality Reduction* **Proceedings of 2010 International Conference on Data Mining (SDM2010)**, 2010.

Meinecke, F. C., von Bünau, P., Kawanabe, M. and Müller, K. R. *Learning invariances with Stationary Subspace Analysis* **ICCV Computer Vision Workshops**, 2009.

von Bünau, P., Meinecke, F. C. and Müller, K. R. *Stationary Subspace Analysis* **Proceedings of ICA Conference 2009 (Paraty), LNCS** 5441:1–8, Springer, 2009.

Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P. and Kawanabe, M. *Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation* **Advances in Neural Information Processing Systems 20 (NIPS)**, 2008.

Chapter 2

Interesting directions in data

2.1 Motivation

In this chapter we explain why a basis transformation of the data space can be useful, review mathematical fundamentals, and discuss two prominent unsupervised coordinate transforms: principal component analysis and independent component analysis.

Data comes from measurements. However, in many circumstances, the actual variables of interest cannot be measured directly. For example, in the neurosciences, one is often limited to measuring non-invasively on the surface of the scalp in order to study the activity of sources located *inside* the brain. In EEG recordings, for instance, each electrode measures a *superposition* of activity from several underlying sources. The input signals are therefore not directly useful for analysing brain activity, because we cannot discern the contributions from the different cortical regions.

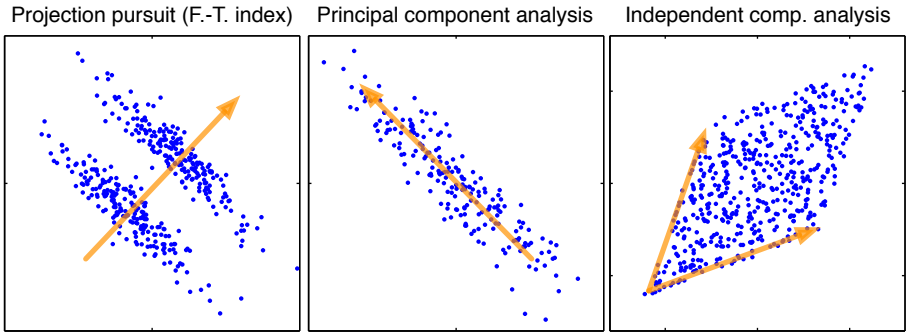

Figure 2.1: Directions found by exploratory projection pursuit (Friedman-Tukey index), principal component analysis and independent component analysis.

However, on multivariate data one can often combine the information of several variables to construct more useful new variables by applying a *coordinate transformation*. In the previous example, a desirable transformation would separate the contribution of the underlying brain sources into new variables. Interestingly, for a diverse range of applications from the neurosciences to the analysis of web data,

useful coordinate transformations can be found based on statistical criteria alone, independent of the particular meaning of the observed variables or prior information. This is the topic of this chapter.

Let us first of all consider intuitive examples for the informativeness of axes in data space. Figure 2.1 shows three datasets. In the left panel, the scatter plot clearly shows two clusters; however, these are not visible by looking at the individual dimensions. Here, a projection onto the orange direction clearly reveals this structure in the data. The variation in the data set shown in the middle is largely confined to the orange axis because the two variables are strongly correlated; coordinates on this axis provide a more compact description of the data. In the right scatter plot, the data points are distributed along two axes in a seemingly independent way, i.e. the value of one coordinate has no influence on the distribution of the other. This suggests that the data is generated as a linear mixture of two underlying (latent) independent factors.

Note that these properties were apparent only because we could look at the whole data in a scatter plot. For higher-dimensional data sets this is not possible. In fact, each of the three examples correspond to a criterion of a family of algorithms: projection pursuit using the Friedman-Tukey index [Friedman and Tukey (1974)] finds directions with cluster structure (left panel); principal component analysis [Pearson (1901)] finds directions that maximize the variance (middle panel), and independent component analysis [Hyvärinen et al. (2001)] finds independent sources from a linear mixture. Incidentally, the latter approach is often used to address the problem sketched at the outset: under the assumption that EEG signals are generated as a linear mixture of independent brain sources, ICA allows us to recover the neural components. Not only the new coordinates are of interest: both the elements of the projection and the basis can often be interpreted in terms of *patterns* if the sensors are spatially distributed. In EEG analysis, the so-called *scalp maps* have become an important tool for the interpretation of sources found by linear methods.

The remainder of this chapter is organised as follows. In the next section we introduce some fundamental concepts from statistics and information theory. In Section 2.3, we review two prominent methods of finding interesting directions. After tracing their historical development, we start with principal component analysis followed by independent component analysis. These are arguably the most widely used coordinate transforms, and the ones we will compare against in the application to EEG analysis (Chapter 5). In the concluding section, we discuss the relative merits of these algorithms.

2.2 Preliminaries and notation

In this section, we briefly review fundamentals from probability theory, statistics and information theory. This overview is largely based on [Mood et al. (1974)],

[Cover et al. (1991)] and [Jaynes and Bretthorst (2003)].

Random variables, joint densities, independence

The building block of statistical modeling is the the concept of the *random variable*. A random variable is a formal variable which represents a sampling or measurement process, governed by a *probability distribution*. Random variables are a convenient tool to formulate probabilistic models in the familiar language of mathematical expressions.

A random variable X is associated with a *probability space* $(\Omega, \mathcal{A}, P)$. We will not give a formal definition; it suffices to say that a probability space specifies the Boolean algebra of random *events* $\mathcal{A} \subset \mathcal{P}(\Omega)$, which consists of subsets of elementary events in Ω, to which we assign probabilities by the function $P : \mathcal{A} \to [0, 1]$. The empty set, the "sure event" Ω and all single elements of Ω are events.

In a probability space, there is no such thing as a value or an order of possible events; it is simply a set of possible outcomes. However, in many cases one wants to model random processes in which the outcome is a real number: we would like to be able to speak of the probability that the outcome lies in a certain interval and to be able to represent this graphically. A *real random variable* does just that: it links the probability space and the real numbers.

Let $(\Omega, \mathcal{A}, P)$ be a probability space. A *real random variable* is a function $X : \Omega \to \mathbb{R}$ such that all subsets $A_r \subset \Omega$ defined as,

$$A_r = \{\omega \in \Omega \,|\, X(\omega) \leq r\}\,,$$

are events in the probability space: $A_r \in \mathcal{A}$ for all $r \in \mathbb{R}$.

This condition ensures that every interval in $\mathbb{R}$ can be assigned a probability. For a continuous random variable, this allows for the definition of the *cumulative distribution function* (CDF) and its derivative, the *probability density function* (PDF), which show the distribution of the probability mass over the real numbers.

The *cumulative distribution function* of a real random variable X is the function $F_X : \mathbb{R} \to [0, 1]$ such that,

$$F_X(x) = P(X \leq x) = P(\{\omega \,|\, X(\omega) \leq x\}),$$

for every real number $x \in \mathbb{R}$. The *probability density function* of X is the function f_x such that $F_X(x) = \int_{-\infty}^{x} f_X(u)du$.

So far, we have considered only single random variables. In many cases, however, the joint behavior of several random variables is of interest. To that end, we introduce *multivariate real random variables* that are a vector of random variables and have a *joint probability density function*.

Let $X_1, \ldots, X_k$ be real random variables over the same probability space. Then $X = (X_1, \ldots, X_k)$ is a *multivariate real random variable* if it has a *joint probability*

density function $f_X : \mathbb{R}^k \to [0,1]$ such that its cumulative distribution function $F_X : \mathbb{R}^k \to [0,1]$ is given by

$$F_X(x_1, \ldots, x_k) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_k} f_X(u_1, \ldots, u_k) du_1 \cdots du_k.$$

A multivariate random variable consisting of k individual random variables is also called k*-variate*. In the context of a multivariate random variable, we call the density of an individual constituent *marginal density*. Given the probability density function of a multivariate random variable, we can obtain its marginals by integrating out the other random variables, respectively.

One motivation for considering the joint distribution of random variables is to study the dependencies between them: how does the outcome of one group of variables influence the density over another group of variables. This is represented by the *conditional density*. Let X and Y be multivariate real random variables that have a joint distribution. The *conditional probability density function* of Y given $X = x$ is defined as, $f_{Y|X}(y|x) = \frac{f_{X,Y}(x,y)}{f_X(x)}$ it is undefined for all $x \in R$ with $f_X(x) = 0$.

A particularly important special case is the *independence* of two random variables. In this case, the outcome of one variable has no influence on the distribution of the other variable. In other words, the conditional distribution is equal to the marginal: $f_{Y|X} \equiv f_Y$ if Y and X are independent. Let X be a k-variate real random variable. Then $X_1, \ldots, X_k$ are *independent*, if their joint probability density function is equal to the product of its marginals, $f_X(x_1, \ldots, x_k) = \prod_{i=1}^{k} f_{X_i}(x_i)$.

Expectation and cumulants

The *expectation* of a function $g(X)$ of a random variable X is, loosely speaking, the average over its random argument X. Let X be a k-variate random variable and $g(X)$ be a scalar function of X. The *expected value of* $g(X)$ is defined as,

$$\mathrm{E}[g(X)] = \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{k \text{ times}} g(x_1, \ldots, x_k) f_X(x_1, \ldots, x_k)\, dx_1 \cdots dx_k,$$

assuming that the integral exists. Since the integral is linear, the expectation is also linear: $\mathrm{E}[\alpha f(X) + \beta g(X)] = \alpha\, \mathrm{E}[f(X)] + \beta\, \mathrm{E}[g(X)]$.

Properties of distributions are often summarised in terms of its *moments* or *central moments*, which are an infinite series of increasing *order*. In general, the r-th order moment of a random variable X is defined as $\mu'_r(X) = \mathrm{E}[X^r]$ and the r-th order central moment is $\mu_r = \mathrm{E}[(X - \mathrm{E}[X])^r]$.

The expectation of a random variable, abbreviated as $\mu(X)$ is its first moment and is commonly interpreted as a location parameter. The *variance* of a random

variable, its second central moment, describe its dispersion around the mean. The *variance* of X, denoted by $\mathrm{var}(X)$ is defined as,

$$\mathrm{var}(X) = \mathrm{E}[(X - \mu(X))^2] = \mathrm{E}[X^2] - \mu(X)^2.$$

The covariance between two random variables describes how they vary together. The value of the covariance is a measure for a linear dependence between them. For a multivariate random variable, all pairwise covariances are summarised in a *covariance matrix*. The covariance of two real random variables X and Y with a joint probability density is given by,

$$\mathrm{cov}(X, Y) = \mathrm{E}[(X - \mu(X))(Y - \mu(Y))].$$

For a k-variate random variable Z, its covariance matrix $\Sigma \in \mathbb{R}^{k\times k}$ contains the covariance for each pair of variables, $\Sigma_{ij} = \mathrm{cov}(Z_i, Z_j)$.

Moments above the mean and covariance are usually called higher order moments. The third central moment, the *skewness*, is often used as a measure of asymmetry around the mean. The *kurtosis* is the fourth central moment which measures the degree of flatness (or peakedness) of a density around its centres. The standardise fourth moment, or Pearson's kurtosis, is defined as $\kappa(X) = \frac{\mu_4}{\mathrm{var}(X)}$.

Information theory

The *entropy* is a measure $\mathrm{H}[X]$ for the uncertainty of a random variable X defined as $\mathrm{H}[X] = -\mathrm{E}[\log f_X(X)]$. The entropy is bounded from below by zero, $\mathrm{H}[X] \geq 0$; it is zero if and only if the random variable is a constant, i.e. non-random. The *conditional entropy* of X given another random variable X, denoted by $\mathrm{H}[X|Y]$ is defined as $\mathrm{H}[X|Y] = -\mathrm{E}_{X,Y}[\log f_{X,Y}(X|Y)]$. The entropy of a joint distribution can be written as $\mathrm{H}[X, Y] = \mathrm{H}[X] + \mathrm{H}[Y|X]$.

The *Kullback-Leibler (KL) divergence* or *relative entropy* D_{KL} is a way of comparing two distributions $p(x)$ and $q(x)$,

$$D_{\mathrm{KL}}[p \,||\, q] = \int_{-\infty}^{+\infty} p(x) \log \frac{p(x)}{q(x)} dx.$$

The KL-divergence is zero if and only if the two densities are identical, and otherwise positive (unbounded). Note, however, that the KL-divergence is not a distance because it is not symmetric and does not satisfy the triangle equation.

A related concept to the KL-divergence is the *mutual information* between random variables. It measures how much information one variable contains about the other. The mutual information $\mathrm{I}[X, Y]$ can be defined using the KL-divergence as,

$$\mathrm{I}[X, Y] = D_{\mathrm{KL}}[p(X, Y) \,||\, p(X)p(Y)],$$

i.e. it is the distance between the joint distribution and the product of its marginals.

The Gaussian distribution

Many statistical techniques are based on distributional assumptions involving the (multivariate) Gaussian distribution. In particular, the noise component in statistical models is usually assumed to be Gaussian distributed. The conventional justification is that by the central limit theorem, the sum of independent identically distributed random variables converges (in distribution) to the normal distribution.

Let X be a k-variate random variable that follows a *multivariate Gaussian distribution* with mean μ and covariance matrix Σ, which we abbreviate by $X \sim \mathcal{N}(\mu, \Sigma)$. The probability density function is of the multivariate Gaussian is given by,

$$f_X(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right).$$

The *entropy of a D-variate Gaussian* $H(\mathcal{N}(\mu, \Sigma))$ is given by $H(\mathcal{N}(\mu, \Sigma)) = \log\sqrt{\det(2\pi\Sigma)} + D/2$. Note that the entropy is thus invariant under orthogonal transformations of variables.

For a given mean and covariance matrix, the Gaussian is the real-valued distribution with the highest entropy [Jaynes (1957)]. From this point of view, the Gaussian distribution is the least restrictive distributional assumption.

The *negentropy* $\mathrm{J}(X)$ of a random variable X with is the difference between its entropy and the entropy of a Gaussian random variable with the same mean and covariance,

$$\mathrm{J}(X) = \mathrm{H}\left[\mathcal{N}(\mu(X), \mathrm{cov}(X))\right] - \mathrm{H}[X].$$

The negentropy is nonnegative.

Maximum likelihood

One the key tasks in statistics is the estimation of model parameters from samples. In the standard *maximum likelihood* (ML) approach, we select those parameters that maximise the likelihood of observing the data at hand under our generative model assumption. Let $\mathcal{X} \subset \mathbb{R}^d$ be a dataset and let $\ell_\theta : \mathcal{P}\left(\mathbb{R}^d\right) \to \mathbb{R}$ the *likelihood function* of a statistical model, which is the likelihood of observing a data set given the parameters θ. The maximum likelihood estimator for the model parameters is given by, $\hat{\theta}_{\mathrm{ML}} = \mathrm{argmax}_\theta \log \ell_\theta(\mathcal{X})$.

The standard estimators for the parameters μ and Σ of a d-variate Gaussian distribution are the *sample mean* $\hat{\mu}$ and the *sample covariance matrix* $\hat{\Sigma}$,

$$\hat{\mu} = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} x \quad \text{and} \quad \hat{\Sigma} = \frac{1}{|\mathcal{X}|-1}\sum_{x\in\mathcal{X}}(x-\hat{\mu})(x-\hat{\mu})^\top,$$

respectively. Note that the latter is not the maximum likelihood estimator, but it has the favorable property of being *unbiased*, i.e. its expectation over realisations of the dataset is equal to the true parameter.

2.3 Maximum variance and independence

In this section, we briefly introduce the family of statistical models corresponding to coordinate transformation in data space and discuss the two most prominent approaches: principal component analysis (PCA) and independent component analysis (ICA).

The generative model for the data is a *linear transformation of latent variables*. The observed D-variate data X is assumed to be generated as a linear transformation of D latent variables Y by an unknown full rank *mixing matrix* $A \in \mathbb{R}^{D\times D}$,

$$X = AY.$$

In a time series context, the data X as called *signals* denoted by $x(t)$ and the latent variables are called *sources* written as $y(t)$. Moreover, the mixing is *instantaneous*, since the observation $x(t)$ at time t only depends on the sources at time t and not on contributions from previous time points. In factor analysis, the matrix A is called *loading matrix*.

The elements of the latent variable Y are referred to as *components* or *factors*. Given only samples from X, the goal is to estimate an inverse for A which allows us to recover the latent variables Y, i.e. find the transformation A^{-1} to coordinates corresponding to the desired components.

This is possible only by making further assumptions on the unknown parameter A and the statistical properties of the latent variables Y. It is these types of assumptions which allow to distinguish the different types of coordinate transformations. The assumption on A and Y also determine up to which symmetries the true mixing matrix A can be identified from the observed data.

2.3.1 Principal component analysis

As an academic discipline, the search for interesting directions in data started with Karl Pearson's work [Pearson (1901)] on principal component analysis (PCA) at the beginning of the 20th century. To this day, PCA is still the most widely used method of extracting useful linear components from a set of high-dimensional data points. The vantage point of Pearson's work was what is now known as ordinary least squares regression (OLS) [Gauß (1809), Legendre (1805)], a method for fitting the mean of a univariate dependent variable (or target) y by a linear combination of independent variables (or covariates) $x_1, \ldots, x_d$. This analysis yields a coefficient α_i for each independent variable x_i, which is commonly interpreted as the effect of the independent variable on the target, conditioned on (or controlled for) fixed values of all other covariates.

The generative model corresponding to linear regression is given by the equa-

tion,

$$Y = \sum_{i=1}^{d} \alpha_i x_i + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma), \tag{2.1}$$

where the only random component is the Gaussian noise term ϵ. The unknown parameters of interest are the coefficients $\alpha_1, \ldots \alpha_d$ and the variance of the noise σ. In this model, the values of the covariates are assumed to be known exactly; only the target is subject to measurement noise. As Pearson points out, it is quite unrealistic that all variables apart from one can be measured exactly. Moreover, the distinction between dependent and independent variable is somewhat arbitrary: one could equally well aim to explain the variation in the covariate x_1 by a linear function of $x_2, \ldots, x_d$ and y. These points are not deficiencies of the linear regression model but reflect the aim of the analysis: to explain the variation of a particular variable in terms of another set of variables in order to investigate the underlying mechanism. In other words, the ordinary least squares model (2.1) belongs to the family of *conditional models*, which model the distribution of a designated variable as a function of another set of variables.

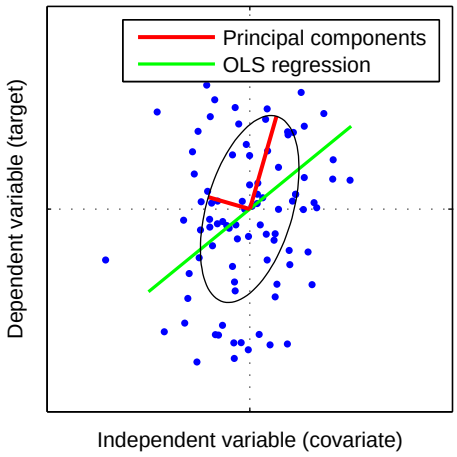

Figure 2.2: Comparison of ordinary least squares regression (OLS) and PCA.

However, as Pearson noted, conditional modeling of a particular variable is not necessarily optimal for explaining the *total variation* in the data, i.e. the joint variation of all variables $x_1, \ldots, x_d$ and y. See Figure 2.2 for an example. To that end, Pearson proposed a method to find the "Closest fit to Systems of Points in Space" [Pearson (1901)], which now known as principal component analysis. PCA finds an orthogonal coordinate system that minimises the reconstruction error of the data. The reconstruction error of a data vector is the L^2-distance to its projection onto a particular basis. The elements of this PCA basis are called *principal components*. The principal components are ordered descendingly by the variance of the data along

each coordinate. That is, the first principal component is the direction in the data that has the highest variance.

We can find the principal components iteratively by minimizing the squared reconstruction error. Assume we are given a d-dimensional centered dataset $x_1, \ldots, x_n \in \mathbb{R}^d$ and we have found the first k orthonormal principal components $v_1, \ldots, v_k$. The next principal component v_{k+1} is the solution to the optimisation problem,

$$\begin{aligned} v_{k+1} &= \operatorname*{argmin}_{\|v\|=1} \sum_{i=1}^{n} \left\| (v^\top x_i)v - x_i \right\|^2 && \text{s.t. } v \perp \operatorname{span}\{v_1, \ldots, v_k\} \\ &= \operatorname*{argmax}_{\|v\|=1} v^\top \left(\sum_{i=1}^{n} x_i x_i^\top \right) v && \text{s.t. } v \perp \operatorname{span}\{v_1, \ldots, v_k\} \\ &= \operatorname*{argmax}_{\|v\|=1} v^\top \hat{\Sigma} v && \text{s.t. } v \perp \operatorname{span}\{v_1, \ldots, v_k\}, \end{aligned} \tag{2.2}$$

where $\hat{\Sigma}$ is the sample covariance matrix of the centered dataset. This optimisation problem has a well known global solution [Golub and Van Loan (1996)] given by the eigenvectors of the sample covariance matrix,

$$\hat{\Sigma} = V D V^\top,$$

where D is a diagonal matrix with ordered positive eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d > 0$ on the diagonal and the columns of V are the corresponding orthonormal eigenvectors $v_1, \ldots, v_d \in \mathbb{R}^d$. In other words, $V^\top$ is an orthogonal transformation which diagonalises the covariance matrix: $V^\top \hat{\Sigma} V = D$, i.e. the data is uncorrelated in the PCA basis. This means that the axis of the ellipsoid representing the covariance matrix are aligned with the principal components, as illustrated in Figure 2.2. Moreover, the eigenvalues correspond to the length of each axis of the ellipsoid. In terms of the optimisation problem (Equation 2.2), each eigenvalue λ_i is the objective function value at the maximum, i.e. the largest variance in the orthogonal complement of $\operatorname{span}\{v_1, \ldots, v_{i-1}\}$. Thus, if we choose to represent our dataset by the first k principal components, then the fraction of the variance that we explain is given by $\sigma_k^{\text{PCA}} = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{d} \lambda_i$. Conversely, the reconstruction error as a fraction of the total variance is $1 - \sigma_k^{\text{PCA}}$. The normalised eigenvalue spectrum therefore gives a good indication of the contribution of each principal component to the variance of the data. In practice, one often encounters spectra of a characteristic shape, where relatively few principal components explain almost all of the variance.

Applications and interpretation

Probably the most common application of PCA is dimensionality reduction (or data compression), guided by the principle of minimizing the reconstruction error: retaining a certain fraction of the variance using the minimum number of dimensions.

Under a Gaussian noise model, this is equivalent to de-noising, if the data lies on a lower-dimensional subspace. In this scenario, one selects the number of dimensions based on the shape of the eigenvalue spectrum, or a pre-specified fraction of the variance that has to be captured. Depending on the application context, one proceeds either in the PCA coordinates, or projects the data onto the PCA subspace in the original coordinates. In factor analysis, the focus is often on interpreting the coefficients of the loading matrix V.

The PCA solution is meaningful only if the relative scale of the input coordinates is not arbitrary. Rescaling a single coordinate changes its variance relative to the other coordinates and therefore the PCA solution. If the relative scaling is arbitrary, then the PCA basis is somewhat arbitrary.

2.3.2 Independent component analysis

Independent component analysis [Hyvärinen et al. (2001)] finds a new coordinate system for the data which maximises the statistical independence of the new coordinates. One motivation of this approach is to recover the underlying (supposedly independent) mechanisms which have generated the data. From an explorative point of view, the advantage of having independent components is that they can be studied univariately, since there are no interactions (dependencies) between them. Whereas PCA is often used as a technical pre-processing step in a semi-automatic fashion, the ICA basis and coordinates are usually subject to further qualitative analysis.

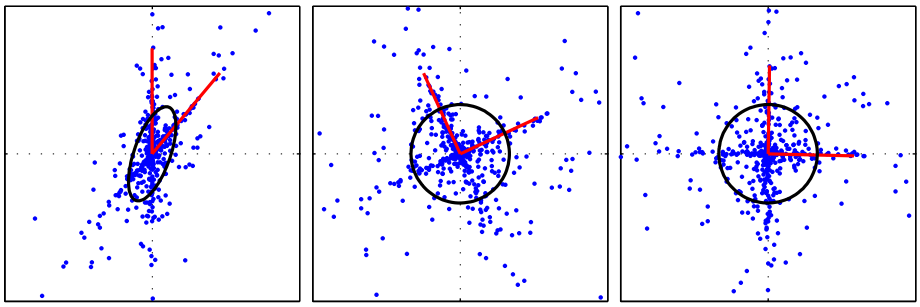

Figure 2.3: ICA example, from left to right: input data; whitening; rotation to independent components. The red lines are the true independent components, the black curve is the contour of the sample covariance matrix.

The generative model of ICA is that the observed data X is generated as a linear mixture of independent latent variables (or sources) Y,

$$X = AY,$$

where we assume that A is invertible and that there are as many observed variables as there are latent variables, both X and Y are assumed to be d-variate.

The aim of ICA is to recover the latent variables from the mixing, i.e. estimate an inverse $\hat{A}^{-1}$ for the mixing matrix such that $\hat{Y} = \hat{A}^{-1}X$. The true mixing matrix is not uniquely identifiable. First of all, the sign and the scaling of the latent sources Y cannot be determined from the observations, because we cannot distinguish between the true sign and scaling and the multiplication of the columns of A by a scalar. Moreover, the true order of the latent sources is not identifiable, because we can absorb a permutation matrix either into the mixing matrix, or into the sources. Thus, we can only identify A up to arbitrary permutation and scaling of its columns[1] In other words, ICA finds an unordered set of one-dimensional subspaces.

Many ICA algorithms proceed in two steps: a whitening followed by a rotation. In the first step, the data is decorrelated by a multiplication with a whitening matrix W, which diagonalises the sample covariance matrix: $W\hat{\Sigma}W^\top = I$. In this basis, the data is uncorrelated but not necessarily independent[2], as we can see in the middle panel of Figure 2.3.

After the decorrelation step, we find a linear transformation that leaves the unity covariance matrix intact, $RIR^\top = I$ and maximises the independence of the latent variables, i.e. we find a orthogonal matrix such that the demixing is given by $\hat{A}^{-1} = RW$. A large family of ICA algorithms are based on minimizing the mutual information between the estimated sources, as a measure of independence. We will see that this is equivalent to maximizing the distance between the distribution of the estimated latent variables and the Gaussian distribution. More formally, let us assume that the data has already been decorrelated by the whitening matrix W such that $\mathrm{cov}(X) = I$. Now we seek the optimal rotation matrix R^* that gives us maximally independent latent variables $\hat{Y} = R^*X$. In terms of an optimisation problem, this rotation is given by,

$$\begin{aligned} R^* &= \underset{RR^\top = I}{\operatorname{argmin}} \ \mathrm{I}\left[\hat{Y}_1, \ldots, \hat{Y}_d\right] \\ &\overset{(1)}{=} \underset{RR^\top = I}{\operatorname{argmin}} \ \sum_{i=1}^{d} \mathrm{H}\left[\hat{Y}_i\right] - \mathrm{H}\left[\hat{Y}\right] \\ &\overset{(2)}{=} \underset{RR^\top = I}{\operatorname{argmin}} \ \sum_{i=1}^{d} \mathrm{H}\left[\hat{Y}_i\right] - \mathrm{H}\left[X\right] - \log\det(R). \end{aligned}$$

Equality (1) results from the definition of mutual information and equality (2) follows from the fact that $\hat{Y}$ is obtained from the observations X by the linear transformation R. Since the entropy of X does not depend on R, and since R is a rotation

[1] Under the assumption that there is not more than one Gaussian-distributed source; if that is not the case, then only the *subspace* of Gaussian sources can be identified.

[2] However, if the data is Gaussian, then uncorrelatedness is equivalent to independence.

(hence $\det R = 1$) we get,

$$\begin{aligned} R^* &= \underset{RR^\top = I}{\operatorname{argmin}} \sum_{i=1}^{d} \mathrm{H}\left[\hat{Y}_i\right] \\ &\overset{(3)}{=} \underset{RR^\top = I}{\operatorname{argmin}} \sum_{i=1}^{d} \left(\mathrm{H}\left[\mathcal{N}(\mu(\hat{Y}_i), \operatorname{cov}(\hat{Y}_i))\right] - \mathrm{J}(\hat{Y}_i)\right) \\ &\overset{(4)}{=} \underset{RR^\top = I}{\operatorname{argmin}} \sum_{i=1}^{d} -\mathrm{J}\left(\hat{Y}_i\right), \end{aligned}$$

where the equality (3) follows from the definition of the negentropy and (4) results from the fact that the entropy of the Gaussian distribution is invariant under the rotation R.

Thus we have seen that minimizing the mutual information is equivalent to maximizing the distance to a Gaussian distribution of the individual sources, as measured by the negentropy. A practical approach to maximizing the non-Gaussianity which lends itself well to numerical optimisation is based on the higher order moments, i.e. moments beyond the mean and the covariance matrix. The JADE algorithm [Cardoso (1999)], for instance, maximises the kurtosis $\operatorname{kurt}\left[\hat{Y}_i\right]$ using Gradient-based numerical optimisation.

A great number of different algorithms have been proposed to find these non-Gaussian projections; some optimise contrast functions like the negentropy by gradient descent [Bell and Sejnowski (1995)], others solve the problem by approximate diagonalisation of the fourth-order cumulant [Cardoso and Souloumiac (1993)] or by deflation-type fixed-point algorithms [Hyvärinen (1999)].

Another family of ICA algorithms exploits the time structure of the data. Examples for these algorithms are e.g. TDSEP [Ziehe and Müller (1998a)] (a generalisation of [Molgedey and Schuster (1994)]) or SOBI [Belouchrani et al. (1997)]. Besides the fact that these methods have access to information that other ICA algorithms do not have, they are often numerically more stable since they rely only on second-order moments.

Applications and interpretation

The archetypal example for an application of independent component analysis is the so-called *cocktail party problem*, where several people are talking simultaneously in a crowded room and one tries to single out one of the speakers from recordings at several microphones distributed in the room. Assuming that the speakers are independent from each other, and that the signals at the microphones are an instantaneous mixture of the speakers, this task can be solved by independent component analysis.

This has lead to the slightly tongue-in-cheek metaphor of the "cerebral cocktail party problem" that one faces in the analysis of non-invasive neural recordings. For example, EEG potentials measured on the surface of the scalp pick up contributions from a variety of sources located inside the brain. In EEG analysis, ICA has been applied successfuly to extract meaningful components in a variety of contexts; see [Jung et al. (2001)] for a review. In particular, the fact that the columns of the estimated mixing matrix can be interpreted as scalp patterns made it possible to use ICA as a visualisation, or imaging, technique.

2.4 Summary and discussion

We have started this chapter by motivating why a linear transformation of variables can be useful: from the perspective of a generative model, we recover the latent variables from the linear mixing. Conversely, the explorative data analyst likes to obtain new variables with desirable properties, such as pair-wise independence. After a brief review of mathematical fundamentals, we introduced the basic setup of linear coordinate transforms, followed by an overview of the two most widely used linear methods: PCA and ICA.

Nonlinear axes?

The linear model is a simplification: the relationship between the latent variables and the observations may well be non-linear. This means that the observed data lies on a non-linear *manifold*, and what we want to find is the intrinsic coordinate system of that manifold. Over the last decade, a large number of methods have been proposed for nonlinear dimensionality reduction or manifold learning. The most popular are kernel PCA [Schölkopf et al. (1998)], locally linear embedding (LLE) [Roweis and Saul (2000)], isomap [Tenenbaum et al. (2000)], self-organizing-maps [Kohonen (1990)], stochastic neighbour embedding [Hinton and Roweis (2002)], and principal curves [Hastie and Stuetzle (1989)].

Despite the fact that the linear model is indeed restrictive, nonlinear methods do not yet enjoy the same level popularity among practitioners outside the statistics and machine learning communities. There are several reasons for this. First of all, most of the nonlinear approaches do not compute an explicit representation for the relationship between latent variables and observations, but only the new coordinates. In that sense, they do not find interesting *directions*. The solution is therefore harder to interpret. Also, it is usually not straightforward to project new data points in the found coordinates—the algorithms need to be run again on the extended data set [Bengio et al. (2004)]. Moreover, as the nonlinear model is more complex, it needs to be regularised in order to avoid overfitting. For kernel PCA [Schölkopf et al. (1998)], this means choosing kernel parameters; for LLE we need

to choose local neighbourhoods and a regularisation parameter, and Isomap depends on the construction of a neighbourhood graph that follows the true manifold exactly. Choosing these parameters is difficult and the solution greatly depend on it. Lastly, nonlinear methods are, overall, less noise robust and require significantly more data. Both of which are characteristics that do not bode well for practical applications.

Useful directions beyond maximum variance and independence

Apart from maximum variance and pair-wise independence, there are other desiderata for linear coordinate transforms. For example, projection pursuit [Huber (1985), Friedman and Tukey (1974)] finds direction that reveals cluster structure in the data; random projections [Rahimi and Recht (2008)] have been found useful for large-scale learning; and methods from EEG source reconstruction [Haufe et al. (2008)] aim to recover brain sources from recordings on the scalp using a combination of prior knowledge, physiological and statistical constraints.

However, none of these criteria aims at understanding *changes in the distribution of data*. This is the problem addressed by stationary subspace analysis (SSA), which we introduce in the next chapter.

Chapter 3

Stationary subspace analysis

How does the distribution of a data set change over time? As we had seen in Chapter 1, this unsuspicious-seeming question can be difficult to answer in the unsupervised multivariate case. That is because from the perspective of the individual input variables, we only observe the *marginals* of the joint distribution, but the relevant changes might occur in the dependencies. Moreover, we do not have labels, timing information, or a target variable that could help us identifying the distribution changes. In a simple example we showed that the input basis can be completely uninformative for understanding changes in the joint distribution.

This is the problem we address in this chapter. We introduce a novel type of linear coordinate transform, stationary subspace analysis (SSA), which finds directions based on the strength of distribution changes. More specifically, SSA identifies two subspaces: the subspace in which the distribution stays constant over time (stationary subspace) and the directions in which the changes are most pronounced (non-stationary subspace).

Such a coordinate transform is useful in many circumstances. "Is there a temporal change in my data?" and "what are these changes?" are common questions in the explorative analysis of data, whenever there is some sort of time structure present. This need not be an explicit time series, but also, for example, ever-growing data sets (such as web data, like that extracted from social media sites) or survey data data that is collected at several points in time. Finding an informative basis for understanding distribution changes is particularly relevant in the high-dimensional case.

Even in a less explorative setting, SSA can be the right tool to extract meaningful components by revealing the *most non-stationary components*. For example, in the analysis of spontaneous neural activity [Bießmann et al. (2011), Allen et al. (2011)], where no timing information or target variables is available, the most non-stationary subspace can contain the information of interest corresponding to the neural activity.

From a more methodological point of view, SSA can be used as an unsupervised feature extraction method for certain tasks. For example, in change-point detection [Blythe et al. (2012)] (or temporal segmentation), one wants to detect the time

points at which the data distribution changes. However, estimating and comparing probability distributions is difficult in high-dimensions. Here, stationary subspace analysis helps by first finding the relevant non-stationary subspace and then confining change point dimension to that lower-dimensional basis. Similarly, in density ratio estimation [Sugiyama (2009)], one is given two data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ and the goal is to compute the ratio of the estimated densities $\hat{p}_{\mathcal{D}_1}(\cdot)/\hat{p}_{\mathcal{D}_2}(\cdot)$ at all points in $\mathcal{D}_1$. This is notoriously difficult in high dimensions. Here, finding a basis in which the distribution of the two data sets differ maximally is a useful pre-processing step. Moreover, many prediction methods rely on the assumption that the distribution of the calibration data is equal to the distribution of the data at application time [Quionero-Candela et al. (2009)]. When this assumption is violated, SSA can contribute to domain adaptation [Hara et al. (2010)] by confining the parameter estimation to the stationary subspace.

The remainder of this chapter is organised as follows. In the next Section 3.1, we introduce the generative model of SSA along with our definition of stationarity and discuss the identifiability of this model. The issue of identifiability is revisited in Section 3.3, where we encounter the phenomenon of spurious stationarity and how it can be overcome. Section 3.2 is concerned with the algorithmic approach to solving the SSA problem and its relationship to statistical testing. Finally, in the last Section 3.4, we summarise and point to open questions and related work.

3.1 Model and identifiability

In the stationary subspace analysis model, the observed data is generated as a linear mixture of two groups of latent variables (or sources): a set of variables whose joint distribution remains constant over time (stationary) and another set of variables whose joint distribution varies (non-stationary). The mixing matrix is assumed to be time-constant and invertible. There are no further assumptions on the latent variables. In particular, both the stationary and the non-stationary variables can have arbitrary dependencies, also across the two groups. More formally, the observed D-variate data X is generated as,

$$X = AS_t = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} S_t^{\mathfrak{s}} \\ S_t^{\mathfrak{n}} \end{bmatrix}, \tag{3.1}$$

where the random variables $S_t^{\mathfrak{s}}$ represents the d stationary variables and $S_t^{\mathfrak{n}}$ are the $D-d$ non-stationary directions and t is the time index $t \in \mathbb{N}$. The first d columns of the mixing matrix $A^{\mathfrak{s}}$ span the stationary subspace and its last $D-d$ columns $A^{\mathfrak{n}}$ span the non-stationary subspace.

The goal of an SSA algorithm is to invert this mixture, i.e. discern stationary and non-stationary contributions in the observed data. Before we turn to the issue of which parts of the model are identifiable, we need to establish more precisely what

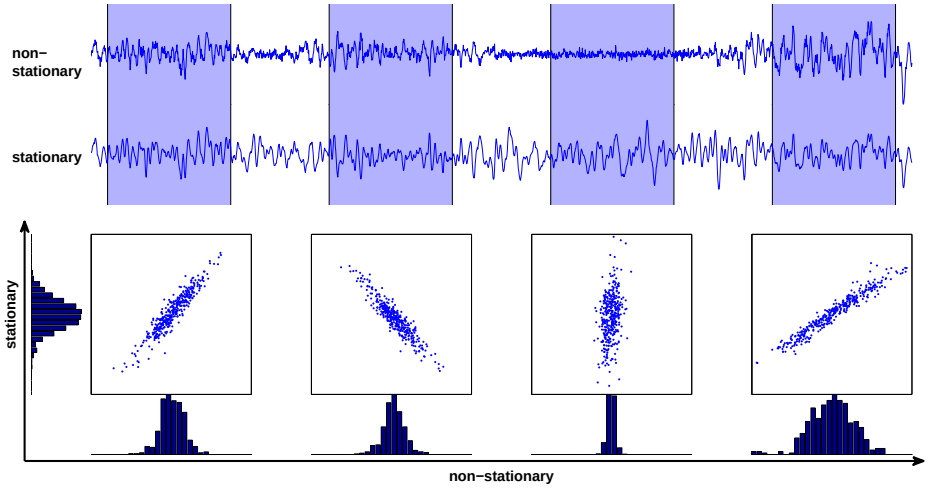

Figure 3.1: Illustration of the SSA model. The non-stationary and the stationary source have a time-variable joint distribution, as shown in the epoch scatter plots. The marginal distribution of the stationary direction (vertical axis) stays constant, whereas the histograms on the horizontal axis change. Note also that the correlation between the two sources changes between epochs, and that they are not distinguishable by their variance (signal power).

is meant by stationarity resp. non-stationarity. In its strongest sense, stationarity means that all properties of the time series $S_t^{\mathfrak{s}}$ remain constant over time. That is, all joint distributions $S_T^{\mathfrak{s}}$ with $T \subset \mathbb{N}$ are exactly equal. This definition does not lend itself well to the derivation of a practical SSA algorithm, since it corresponds to an infinite number of constraints. At the same time, only a few of the quantities to which these constraints pertain can be estimated reliably from finite data.

A pragmatic definition of stationarity

This is reflected in our definition of stationarity: we only consider the first two moments (mean and covariance matrix) of the probability distribution and ignore the time structure. That is, a d-variate time series Y_t is stationary if

$$\mathrm{E}[Y_{t_1}] = \mathrm{E}[Y_{t_2}] \quad \text{and} \quad \mathrm{E}[Y_{t_1} Y_{t_1}^\top] = \mathrm{E}[Y_{t_2} Y_{t_2}^\top],$$

for all pairs of time points $t_1, t_2 \in \mathbb{N}$. This type of stationarity is also called *weak stationarity* [Priestley (1983)] without time structure.

This clearly is a limited notion of stationarity. For example, note that a time series whose frequency content changes while its power stays constant would be deemed stationary. Similarly, a non-stationary time series where all changes are confined to higher order moments is weakly stationary; though this is probably rare in practice.

However, in practical applications we have found that this type of stationarity already achieves interesting novel results, while not requiring large amounts of samples.

What can we recover from the mixture?

The aim of stationary subspace analysis is to invert the mixture of stationary and non-stationary sources given only samples from the mixture X. That is, we want to find a demixing matrix,

$$\hat{A}^{-1} = \begin{bmatrix} B^{\mathfrak{s}} \\ B^{\mathfrak{n}} \end{bmatrix} \text{ with } B^{\mathfrak{s}} \in \mathbb{R}^{d \times D} \text{ and } B^{\mathfrak{n}} \in \mathbb{R}^{D-d \times D},$$

that consists of the matrices $B^{\mathfrak{s}}$ and $B^{\mathfrak{n}}$ to the stationary and non-stationary directions respectively. In the following, we will refer to "changing into the source coordinates" as *projecting* and denote $B^{\mathfrak{s}}$ and $B^{\mathfrak{n}}$ as the $\mathfrak{s}$-projection and the $\mathfrak{n}$-projection respectively. Note that we assume that the true number of stationary directions d is known. Applying these projections to the mixture yields the estimated latent variables,

$$\hat{S}_t^{\mathfrak{s}} = B^{\mathfrak{s}} X \text{ and } \hat{S}_t^{\mathfrak{n}} = B^{\mathfrak{n}} X.$$

Before we turn to the problem of finding this demixing, let us consider its identifiability. Assuming that the generative model is correct and that we know the the true number of stationary and non-stationary directions, an important question is: up to which symmetries can we recover the true model parameters from the mixed signals? Our model has one parameter, the mixing matrix $A = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix}$, which consists of a basis for the stationary and the non-stationary subspace respectively. A criterion for identifying A must be based on recovering the latent variables with the prescribed properties from the mixed data.

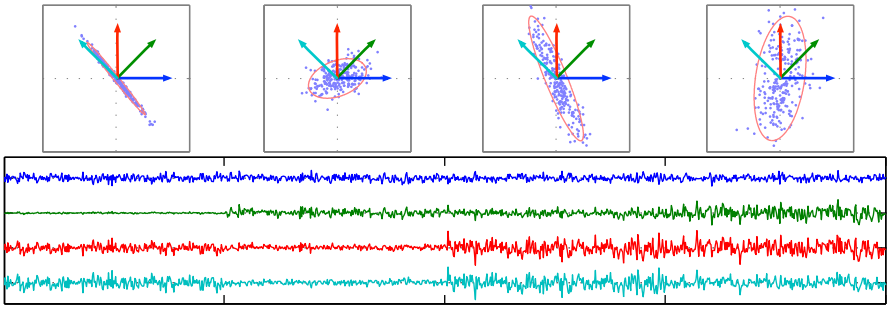

Figure 3.2: There is only one stationary direction (blue) but several non-stationary ones: projecting onto the green, red and turqoise directions yields a non-stationary time series (see below).

The only assumption we have made about the latent variables is that there are d stationary and $D-d$ non-stationary ones. An SSA algorithm which achieves recovers sets of sources with these characteristics has exhausted all available information. Thus, if we apply such an ideal demixing $\hat{A}^{-1}$ to the observed data X_t,

$$\begin{bmatrix} \hat{S}_t^{\mathfrak{s}} \\ \hat{S}_t^{\mathfrak{n}} \end{bmatrix} = \hat{A}^{-1} X_t = \hat{A}^{-1} A \begin{bmatrix} S_t^{\mathfrak{s}} \\ S_t^{\mathfrak{n}} \end{bmatrix} = \begin{bmatrix} B^{\mathfrak{s}} A^{\mathfrak{s}} & B^{\mathfrak{s}} A^{\mathfrak{n}} \\ B^{\mathfrak{n}} A^{\mathfrak{s}} & B^{\mathfrak{n}} A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} S_t^{\mathfrak{s}} \\ S_t^{\mathfrak{n}} \end{bmatrix},$$

we obtain a criterion for a valid demixing in terms of its relationship to the true mixing matrix.

First of all, in order for the stationary project $B^{\mathfrak{s}}$ to yield stationary projections, all non-stationary contributions need to be removed from the mixture, i.e. $B^{\mathfrak{s}} A^{\mathfrak{n}} = 0$. This means that we can identify a basis for the true non-stationary subspace, because it is uniquely given as the dual space of the rows of $B^{\mathfrak{s}}$. However, the product $B^{\mathfrak{s}} A^{\mathfrak{s}}$ is unconstrained since it corresponds to a linear transformation of the stationary sources, which does not change their stationary nature for an example. Thus we can only identify the true stationary sources up to linear transformations. The same holds for the non-stationary sources: $B^{\mathfrak{n}} A^{\mathfrak{n}}$ is arbitrary. Moreover, the $B^{\mathfrak{n}} A^{\mathfrak{s}}$ is also unconstrained, since in general we can add arbitrary stationary contributions to a non-stationary source without making it stationary. Figure 3.2 shows an example, where there is only one stationary direction, but several directions yielding non-stationary time series. The true stationary subspace is therefore not identifiable.

3.2 The SSA algorithm

In the previous section, we have observed that we can only recover the true stationary sources (up to linear transformations), and that the non-stationary sources are not identifiable in principle. We will therefore start by developing an optimisation criterion for finding the stationary projection $B^{\mathfrak{s}}$, before we consider the second part of the demixing matrix $B^{\mathfrak{n}}$.

Measuring stationarity

The starting point for finding the $\mathfrak{s}$-projection is to define a measure of stationarity. That is, given samples $x_1, \ldots, x_m$ from a d-dimensional time series, we need a way of quantifying its degree of stationarity. According to our definition, a time series is stationary if its first two moments are constant over time. Therefore, a natural measure of stationarity is based on the *distance* of the mean and covariance matrix. To jointly compare the first two moments, we need to combine the distance in the mean and the covariance matrix in a principled way. To that end, we use the

Kullback-Leibler divergence between D-variate Gaussians, which is given by,

$$D_{\mathrm{KL}}\left[\mathcal{N}(\mu_0, \Sigma_0) \,||\, \mathcal{N}(\mu_1, \Sigma_1)\right] = \\ = \frac{1}{2}\left(\log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right) + \mathrm{trace}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0) - D\right).$$

This means that we approximate the probability density by a Gaussian distribution. Note, however, that in the following derivation we will not assume that the sources are Gaussian distributed. The Gaussian merely serves as the least restrictive distributional assumption for data with fixed mean and covariance matrix, in the sense that it has the highest entropy [Jaynes (1957)].

In order to compute this stationarity measure from data, we need to estimate the moments at several points in time. This we do either by segmenting the time series into consecutive epochs, or by using a sliding window[1]. Let $\hat{\mu}_1, \ldots, \hat{\mu}_n \in \mathbb{R}^d$ and $\hat{\Sigma}_1, \ldots, \hat{\Sigma}_n \in \mathbb{R}^{d\times d}$ be the epoch mean and covariance matrices respectively, where n is the total number of epochs. Moreover, let m be the total number of samples and $m_1, \ldots, m_n$, the number of samples in each epoch.

Based on the KL-divergence between two epochs, there are now several ways of defining a measure of stationarity for the whole time series. For example, we could sum up the KL-divergence between all pairs of epochs. We choose a different approach, which is computationally more efficient: comparing each epoch against a reference epoch. Clearly, the sum of these KL-divergences is zero, if and only if, all moments are equal and otherwise positive. As a reference epoch, we choose the average epoch, with moments $\bar{\mu}$ and $\bar{\Sigma}$,

$$\bar{\mu} = \sum_{i=1}^{n} \mu_i \;\text{ and }\; \bar{\Sigma} = \sum_{i=1}^{n} \Sigma_i.$$

Therefore, given an epochised time series we measure its stationarity as,

$$L(\mu_1, \ldots, \mu_n, \Sigma_1, \ldots, \Sigma_n) = \sum_{i=1}^{n} D_{\mathrm{KL}}\left[\mathcal{N}(\mu_i, \Sigma_i) \,||\, \mathcal{N}(\bar{\mu}, \bar{\Sigma})\right],$$

which is zero if and only if the time series is perfectly stationary. Note that this measure of stationarity can grow arbitrarily large.

From a measure of stationarity to an objective function

In order to find the stationary projection, we minimise the non-stationarity of the estimated stationary directions. Thus the optimal stationary projection is the solu-

[1]The design of the epoch structure is an important issue that we will discuss later.

tion to the optimisation problem,

$$\begin{aligned} B^{\mathfrak{s}*} &= \operatorname*{argmin}_{B\in\mathbb{R}^{d\times D}} L(B\mu_1,\ldots,B\mu_n,B\Sigma_1B^\top,\ldots,B\Sigma_nB^\top) \\ &= \operatorname*{argmin}_{B\in\mathbb{R}^{d\times D}} \sum_{i=1}^{n} D_{\mathrm{KL}}\left[\mathcal{N}(B\mu_i,B\Sigma_iB^\top)\,||\,\mathcal{N}(B\bar{\mu},B\bar{\Sigma}B^\top)\right]. \end{aligned}$$

In this form, the objective function does not lend itself well to numerical optimisation: without constraints on B, degenerate solutions cannot be avoided (e.g. a rank deficient projection) and the fact that the objective contains inverses of a projected covariance matrix can lead to numerical instability.

In order to make the objective more benign, we exploit the invariances that we observed earlier while studying the identifiability of the model parameters. As we will see in the following, we can fix the mean and the covariance matrix of the average epoch to $\underline{0}$ and I respectively without loss of generality by choosing a different basis and introducing constraints.

First of all, we can translate all epoch mean vectors without changing the solution, because the distance between the epochs does not depend on their overall location. Thus we centre the average epoch, i.e. translate the data such that $\bar{\mu} = \underline{0}$. Moreover, we can perform the optimisation in an arbitrary basis. Thus we first apply a whitening such that the average covariance matrix is the identity, $\bar{\Sigma} = I$. In order to ensure that the moments of the average epoch stay constant under any projection, we add the constraint that $BB^\top = I$. Recall that we can only identify the true stationary projection up to arbitrary linear transformations, so this is not a restriction. This means that we have written the demixing matrix as a whitening of the original data by a matrix W followed by an orthogonal matrix, i.e. $\hat{A}^{-1} = RW$ with $RR^\top = I$. Note, however, that only the first d rows of the demixing matrix enter into the objective when optimizing for the stationary projection, because we only evaluate the estimated stationary sources.

In the following we assume that that the centring and whitening has already been applied to the data such that $\bar{\mu} = \underline{0}$ and $\bar{\Sigma} = I$. Thus we can rewrite the optimisation

problem as,

$$
\begin{aligned}
B^{\mathfrak{s}*} &= \operatorname*{argmin}_{BB^\top = I} \sum_{i=1}^{n} D_{\mathrm{KL}} \left[\mathcal{N}(B\mu_i, B\Sigma_i B^\top) \,||\, \mathcal{N}(0, I) \right] \\
&= \operatorname*{argmin}_{BB^\top = I} \frac{1}{2} \sum_{i=1}^{n} \Big(-\log\det(B\Sigma_i B^\top) + \operatorname{trace}(B\Sigma_i B^\top) + \|B\mu_i\|^2 - d \Big) \\
&= \operatorname*{argmin}_{BB^\top = I} \frac{1}{2} \sum_{i=1}^{n} \Big(-\log\det(B\Sigma_i B^\top) + \|B\mu_i\|^2 \Big) - \frac{nd}{2} \\
&\qquad\qquad + \frac{1}{2} \operatorname{trace}\left(B \left(\sum_{i=1}^{n} \Sigma_i \right) B^\top \right) \\
&\overset{(1)}{=} \operatorname*{argmin}_{BB^\top = I} \frac{1}{2} \sum_{i=1}^{n} \Big(-\log\det(B\Sigma_i B^\top) + \|B\mu_i\|^2 \Big) - \frac{nd}{2} + \frac{n}{2} \operatorname{trace} I_{d\times d} \\
&= \operatorname*{argmin}_{BB^\top = I} \underbrace{\frac{1}{2} \sum_{i=1}^{n} \Big(-\log\det(B\Sigma_i B^\top) + \|B\mu_i\|^2 \Big)}_{=J(B)}, \qquad (3.2)
\end{aligned}
$$

where the equation 1 above follows from the fact that we have set the average covariance matrix to the identity. The gradient of the objective function J with respect to the projection B is given by

$$
\frac{\partial J}{\partial B} = \sum_{i=1}^{n} \left(B\Sigma_i B^\top \right)^{-1} B\Sigma_i + B\mu_i\mu_i^\top, \qquad (3.3)
$$

from which we see that the objective function is not convex due to the log-determinant.

Gradient descent over the special orthogonal group SO(D)

Even thought we have achieved a simpler, and computationally advantageous objective function (3.2), the orthogonality constraint $BB^\top = I$ makes it difficult to optimise. In the next step, we therefore eliminate this constraint by choosing a parametrisation that ensures that the constraint is automatically fulfilled. To that end, instead of optimizing a projection matrix, we are looking for an orthogonal transformation $RR^\top = I$ of the data such that the first d coordinates are stationary. In the objective function, this means that we write B in terms of R as its truncation $B = I_d R$ to the first d rows (I_d is an identity matrix truncated to the first d rows).

In order to find the orthogonal matrix R, we use an iterative optimisation procedure using multiplicative updates by an orthogonal matrix. Starting with a random orthogonal matrix, in the k-th step we find an orthogonal update U such that

$R_{k+1} = UR_k$. After termination in the ℓ-th step, the found solution is $B^{\mathfrak{s}*} = I_d R_\ell W$, where W is the whitening matrix.

But how do we find each orthogonal update? Following the standard gradient-based optimisation strategy, what we would like to do is to identify a *direction of steepest descent* in the set of orthogonal transformations and then perform a line search along this direction to find the optimal step.

Let us first of all take a closer look at this set of orthogonal matrices. The orthogonal matrices form a multiplicative group: the product of two orthogonal matrices is orthogonal and the inverse of an orthogonal matrix is orthogonal; the identity matrix is orthogonal. This group is called the orthogonal group $\mathsf{O}(D)$ and it consists of elements of two types: rotations with determinant $+1$ and reflections with determinant -1. The rotations are a normal subgroup of the orthogonal group, called *special orthogonal group* $\mathsf{SO}(D) \lhd \mathsf{O}(D)$. To find the true stationary projection, we need to project orthogonal to the non-stationary subspace, i.e. we want to remove stationary contributions from the first d coordinates. This can always be achieved by a rotation of the data; and at each step we can improve on this objective by a applying a rotation — changing the sign of one coordinate (i.e. apply a reflection) does not change the objective. Hence we can constrain our search for the direction of steepest descent to the subgroup $\mathsf{SO}(D)$, the rotation matrices.

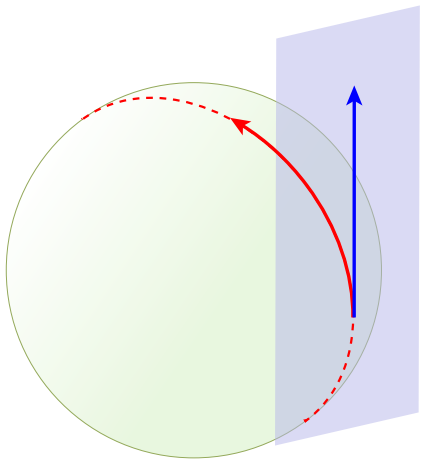

Figure 3.3: Illustration of the tangent space (blue plane) at one point of the group (sphere). A direction in the tangent space (blue arrow) corresponds to a one-parameter subgroup (or geodesic) in the group, shown as the curved red arrow.

In order to find the update rotation U, we compute the gradient of our objective function $J(U)$ and use it as a search direction over the special orthogonal group. This is possible because $\mathrm{SO}(D)$ is a Lie group, i.e. it is a continuous differentiable manifold. However, since the $\mathrm{SO}(D)$ is not a Euclidean space but a manifold, defining the gradient is a bit more involved. The gradient at a point on a manifold is

an element of the *tangent space* (linearisation) at that point. In a Euclidean space, the tangent space at each element is equal to the Euclidean space itself, because it is a linear space. On a curved manifold, this is not the case; see Figure 3.3 for an illustration

In a Lie group, the tangent space at the identity has a special meaning: it is the corresponding *Lie algebra*. Informally speaking, the Lie algebra is a vector space spanned by the infinitesimal transformations corresponding to the *group action*, which in our case, are the infinitesimal rotations.

The Lie algebra $\mathfrak{so}(D)$ of the special orthogonal group $\mathsf{SO}(D)$ is the set of all matrices $M \in R^{D\times D}$ that are anti-symmetric, $M = -M^\top$. The element M_{ij} can be interpreted as the angle of an infinitesimal rotation of axis j towards axis i. Since we absorb the update rotations into the data, at each step we want to find the gradient of $J(U)$ at the identity I. The gradient is thus an element of the Lie algebra and we can find it by using the connection between the group $\mathrm{SO}(D)$ and the algebra $\mathfrak{so}(D)$ provided by the *exponential map*. The exponential map,

$$\exp : \mathfrak{so}(D) \to \mathrm{SO}(D),$$

is a one-to-one correspondence between the skew-symmetric matrices and the rotation matrices, given by the matrix exponential. Geometrically speaking, it maps straight lines in the tangent space onto geodesics on the manifold. See the blue and right arrows in Figure 3.3 for an example.

Following [Plumbley (2005)], we parametrise the update U by an antisymmetric matrix $U = \exp(M)$ and calculate the gradient $\nabla_{\mathfrak{so}(D)} J(\exp(M))$ at $M = 0$ to find the search direction. Using this gradient, we can perform a line search along the corresponding geodesic $\left\{\exp\left(t\,\nabla_{\mathfrak{so}(D)} J\right) \mid t \in \mathbb{R}\right\}$ to find the update U.

To calculate the gradient in the Lie algebra, we need a notation of distance between anti-symmetric matrices. A natural way to define a norm is to adapt the Frobenius-norm to reflect the fact that every element of $M \in \mathfrak{so}(D)$ appears twice,

$$\|M\|^2 = \frac{1}{2}\sum_{i,j} M_{ij}^2,$$

by including the factor $1/2$. This corresponds to an inner product for $M_1, M_2 \in \mathfrak{so}(D)$ given by,

$$\langle M_1, M_2\rangle = \frac{1}{2}\sum_{i,j}(M_1)_{ij}(M_2)_{ij} = \frac{1}{2}\,\mathrm{trace}(M_1^\top M_2).$$

Using this scalar product, we can implicitly define the gradient in the Lie algebra in terms of its projection onto unit vectors. Let $H \in \mathfrak{so}(D)$ be a unit vector $\|H\| = 1$ in the algebra; then the component of the gradient in the direction of H is the scalar

product $\langle \nabla_{\mathfrak{so}(D)} J, H \rangle$. Moreover, we the derivative of the update rotation $U = \exp(tH)$ along the direction H (a so called single-parameter subgroup) is given by,

$$\frac{\partial \exp(tH)}{\partial t} = H \exp(tH). \tag{3.4}$$

In the following, we will obtain an explicit expression for the gradient $\nabla_{\mathfrak{so}(D)} J$ by equating the derivative of J along H with the projection of the gradient $\nabla_{\mathfrak{so}(D)} J$ onto H in the Lie algebra. First of all, using the chain rule, we get an expression for the derivative of J along H,

$$\begin{aligned}
\frac{\partial J(\exp(tH))}{\partial t} &= \operatorname{trace}\left[\left(\frac{\partial J}{\partial U}\right)^\top \frac{\partial U}{\partial t}\right] \\
&\overset{(1)}{=} \operatorname{trace}\left[\left(\frac{\partial J}{\partial U}\right)^\top HU\right] \\
&\overset{(2)}{=} \operatorname{trace}\left[U\left(\frac{\partial J}{\partial U}\right)^\top H\right] \\
&\overset{(3)}{=} \operatorname{trace}\left[\frac{1}{2}\left(U\left(\frac{\partial J}{\partial U}\right)^\top - \left(\frac{\partial J}{\partial U}\right)U^\top\right)H\right] \\
&\overset{(4)}{=} \left\langle \left(\frac{\partial J}{\partial U}\right)U^\top - U\left(\frac{\partial J}{\partial U}\right)^\top, H \right\rangle,
\end{aligned}$$

where equality 1 follows from Equation 3.4; equality 2 is a property of the trace; equality 3 holds because $H \in \mathfrak{so}(D)$ and is therefore antisymmetric; and equality 4 corresponds to our definition of the scalar product in the Lie algebra.

By the definition of the gradient, its projection onto H must be equal to the partial derivative $\partial J(\exp(tH)/\partial t$,

$$\begin{aligned}
\langle \nabla_{\mathfrak{so}(D)} J, H \rangle &= \partial J(\exp(tH)/\partial t \\
&= \left\langle \left(\frac{\partial J}{\partial U}\right)U^\top - U\left(\frac{\partial J}{\partial U}\right)^\top, H \right\rangle,
\end{aligned}$$

and since this equality holds for *all* unit vectors $H \in \mathfrak{so}(D)$ it follows that,

$$\nabla_{\mathfrak{so}(D)} J(\exp(M)) = \left(\frac{\partial J}{\partial U}\right)U^\top - U\left(\frac{\partial J}{\partial U}\right)^\top. \tag{3.5}$$

Using this equation, we can compute the gradient in the Lie algebra $\mathfrak{so}(D)$ at $M = 0$ for any objective function $J(\exp(M)) = J(U)$ for which we know the partial derivative $\partial J/\partial U$. This is a classical approach, for example in ICA [Plumbley (2005)].

Let us now have a look at the structure of the gradient of $J(U)$. Since only the first d rows of the update rotation U affect the objective, only the first d rows of the derivative $\partial J/\partial U$ are non-zero. The non-zero part of $\partial J/\partial U$ is given by Equation 3.3. As we evaluate the gradient at the identity $M = 0$, and since the upper $d \times d$ submatrix of $\partial J/\partial U$ is symmetric, it follows that the gradient in the Lie algebra has the form,

$$\nabla_{\mathfrak{so}(D)} J(\exp(M))\big|_{M=0} = \frac{\partial J}{\partial U} - \left(\frac{\partial J}{\partial U}\right)^\top = \begin{bmatrix} 0 & Z \\ -Z^\top & 0 \end{bmatrix}, \qquad (3.6)$$

where $Z \in \mathbb{R}^{d \times D}$ is the non-zero part. This block-structure has an intuitive interpretation: the two square zero parts of dimension d and $(D-d)$ correspond to rotations *within* the stationary and non-stationary source space respectively. These do not change the objective. The non-zero part Z contains the angles *between* the two subspaces. Thus we can reduce the number of variables in the optimisation to $d(D-d)$.

The gradient (Equation 3.6) defines a search direction in the Lie algebra, which allows for geodesic line search over the rotations using the exponential map. The whole procedure is summarised in Algorithm 1.

Algorithm 1 Stationary subspace analysis: finding the projection $B^{\mathfrak{s}*}$ to the most stationary sources

function SSA($d, \mu_1, \ldots, \mu_n, \Sigma_1, \ldots, \Sigma_n$)
  Compute the average mean $\bar{\mu} \leftarrow \frac{1}{n}\sum_{i=1}\mu_i$.
  Compute the average covariance matrix $\bar{\Sigma} \leftarrow \frac{1}{n}\sum_{i=1}\Sigma_i$.
  Center the epochs $\mu_i \leftarrow (\mu_i - \bar{\mu})\ \forall i$
  Compute the whitening matrix $W \leftarrow \bar{\Sigma}^{-\frac{1}{2}}$
  Let $B \leftarrow$ random rotation matrix in $\mathrm{SO}(D)$
  Initialise the epochs $\mu_i \leftarrow (BW)\mu_i$ and $\Sigma_i \leftarrow (BW)\Sigma_i(BW)^\top\ \forall i$
  repeat
    Let $G \leftarrow \nabla_{\mathfrak{so}(D)} J(U)$ be the gradient of the objective at $U = I$
    Perform a line search for t^* on the geodesic $\{\exp(tG) \mid t \in \mathbb{R}\}$
    Let $U \leftarrow \exp(t^* G)$ be the update rotation
    Store the update $B \leftarrow UB$
    Rotate the epochs $\mu_i \leftarrow U\mu_i$ and $\Sigma_i \leftarrow U\Sigma_i U^\top\ \forall i$
  until convergence
  Let $B^{\mathfrak{s}*} \leftarrow I_d BW$ be the stationary projection
  return $B^{\mathfrak{s}*}$
end function

Finding the most non-stationary sources

As we have seen in the previous sections, the *true* non-stationary sources cannot be identified from the mixture because one can add arbitrary stationary components without changing their non-stationary nature. However, in practice one is often interesting in finding the *most* non-stationary directions instead of recovering the *true* non-stationary directions. This, however, can not necessarily be achieved by projecting orthogonal (in the whitenend basis) to the stationary components, i.e. choosing the bottom $D - d$ rows of the data rotation as the non-stationary projection (see Algorithm 1). The right panel of Figure 3.4 shows a situation where the projection that is orthogonal to the true $\mathfrak{s}$-projection yields an almost stationary latent variable, as the projected variance in the two epochs is almost equal. As it turns out, whenever there are changes in the correlation between the stationary and non-stationary variables, we can find more non-stationary directions by explicitly maximizing the non-stationarity in a second step.

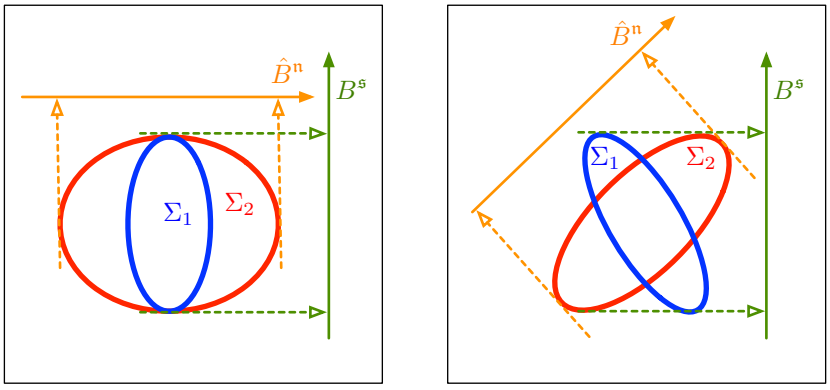

Figure 3.4: The left panel shows two epoch covariance matrices where the covariance between the stationary and the non-stationary direction does not change. Here, projection orthogonal to the stationary subspace yields the most non-stationary sources. In the right panel, the non-stationarity is mostly contained in the time-variable covariance; projecting orthogonal to the stationary subspace yields only mildly non-stationary sources.

Let us analyse the situation more rigorously. We consider the simple case where we have one stationary and one non-stationary source with corresponding normalised basis vectors $\|A^{\mathfrak{s}}\| = 1$ and $\|A^{\mathfrak{n}}\| = 1$ respectively, where ϕ be the angle between the two spaces, i.e. $\cos\phi = A^{\mathfrak{s}\top} A^{\mathfrak{n}}$. We will consider an arbitrary pair of epochs, $\mathcal{T}_1$ and $\mathcal{T}_2$, and show which projection maximises the difference in mean Δ_μ and variance Δ_σ between them, in relation to the true stationary projection $B^{\mathfrak{s}}$.

Let X_1 and X_2 be bivariate random variables modeling the distribution of the data in the two epochs. According to the linear mixing model, we can write X_1 and

X_2 in terms of the underlying sources,

$$X_1 = A^{\mathfrak{s}} X_s + A^{\mathfrak{n}} X_{n_1}$$
$$X_2 = A^{\mathfrak{s}} X_s + A^{\mathfrak{n}} X_{n_2}$$

where the univariate random variable X_s represents the stationary source and the two univariate random variables X_{n_1} and X_{n_2} model the non-stationary source, whose probability distribution changes between the two epochs, i.e. X_{n_1} is not equal to X_{n_2} in distribution. In the following, we will assume without loss of generality, that the true $\mathfrak{s}$-projection $B^{\mathfrak{s}} = (A^{\mathfrak{n}})^{\perp}$ is normalised, $\|B^{\mathfrak{s}}\| = 1$. We can write the estimated $\mathfrak{n}$-projection as a linear combination of $B^{\mathfrak{s}}$ and $A^{\mathfrak{n}}$,

$$\hat{B}^{\mathfrak{n}} = \alpha B^{\mathfrak{s}} + \beta A^{\mathfrak{n}\top}, \tag{3.7}$$

with $\alpha, \beta \in \mathbb{R}$ such that $\|\hat{B}^{\mathfrak{n}}\| = 1$.

In the next step, we will observe which $\mathfrak{n}$-projection maximises the difference in mean and covariance between the two epochs. Let us first consider the difference in the mean of the estimated $\mathfrak{n}$-sources,

$$\Delta_\mu = \mathrm{E}[\hat{B}^{\mathfrak{n}} X_1] - \mathrm{E}[\hat{B}^{\mathfrak{n}} X_2] = \hat{B}^{\mathfrak{n}} A^{\mathfrak{n}} (\mathrm{E}[X_{n_1}] - \mathrm{E}[X_{n_2}])$$

which is maximal for $\hat{B}^{\mathfrak{n}} A^{\mathfrak{n}} = 1$, i.e. when $\hat{B}^{\mathfrak{n}}$ is orthogonal to $B^{\mathfrak{s}}$. Thus, with respect to the difference in the mean, choosing the $\mathfrak{n}$-projection $\hat{B}^{\mathfrak{n}}$ to be orthogonal to the $\mathfrak{s}$-projection is always optimal. The difference in the variance of the estimated $\mathfrak{n}$-sources is,

$$\begin{aligned}\Delta_\sigma = \mathrm{var}[\hat{B}^{\mathfrak{n}} X_1] - \mathrm{var}[\hat{B}^{\mathfrak{n}} X_2] &= \beta^2 (\mathrm{var}[X_{\mathfrak{n}_1}] - \mathrm{var}[X_{\mathfrak{n}_2}]) \\ &+ 2\left[\alpha \cos\left(\phi + \frac{\pi}{2}\right) + \beta \cos\phi\right] \underbrace{(\mathrm{cov}[X_{\mathfrak{s}}, X_{\mathfrak{n}_1}] - \mathrm{cov}[X_{\mathfrak{s}}, X_{\mathfrak{n}_2}])}_{=\Delta_{\sigma_{\mathfrak{sn}}}}.\end{aligned}$$

Clearly, when there is no change in the covariance of the $\mathfrak{s}$- and the $\mathfrak{n}$-sources between the two epochs ($\Delta_{\sigma_{\mathfrak{sn}}} = 0$), this difference is maximised when β is maximal, which by Equation 3.7 implies that $\hat{B}^{\mathfrak{n}} = (B^{\mathfrak{s}})^{\perp}$. However, when the covariance between $\mathfrak{s}$- and $\mathfrak{n}$-sources does vary ($|\Delta_{\sigma_{\mathfrak{sn}}}| > 0$), the derivative of Δ_σ with respect to α at $\alpha = 0$ is ,

$$\partial \Delta_\sigma / \partial \alpha|_{\alpha=0} = 2 \cos\left(\phi + \frac{\pi}{2}\right) \Delta_{\sigma_{\mathfrak{sn}}}.$$

Hence $\alpha = 0$ is not an extremum when $|\Delta_{\mathfrak{sn}}| > 0$, which means that the most non-stationary projection is not orthogonal to the $\mathfrak{s}$-projection in this case.

Thus, in order to find the projection to the most non-stationary sources, we also need to maximise the non-stationarity of the estimated $\mathfrak{n}$-sources. To that end, we

simply maximise the SSA objective function (Equation 3.2) for the $\mathfrak{n}$-projection, i.e.

$$B^{\mathfrak{n}*} = \underset{BB^\top = I}{\operatorname{argmax}} \frac{1}{2} \sum_{i=1}^{n} \left(-\log\det(B\Sigma_i B^\top) + \|B\mu_i\|^2 \right).$$

The procedure is completely analogous to Algorithm 1, apart from the fact that we reverse the sign of the objective.

Stationary subspace analysis in deflation mode

SSA factorises the observed data into two groups of sources, both of which can only be determined up to arbitrary linear transformations. In contrast to a PCA or ICA solution, the results of SSA are two multidimensional subspaces and not a set of one-dimensional components. This can make the solution more difficult to interpret, because there is no unique set of projections resp. basis elements. Therefore, in some scenarios it is desirable to perform SSA univariately, in so-called deflation mode. This means that we find a set of uncorrelated one-dimensional components which are ordered by their degree of non-stationarity.

In each step, we optimise the non-stationarity of a single component; after convergence, we find the next component in the orthogonal complement (in the whitened basis). This procedure is called deflation mode because in each step, we reduce the dimensionality of the data space by one. In this way, we ignore changes in the correlation between variables. However, the advantage of this approach is that the solution is unique (up to local minima). What is more, the fact that we have factorised the input space into a set of one-dimensional subspace allows for a comparison of their non-stationarity in terms of the value of the objective function.

3.2.1 Relationship to statistical testing

The absolute value of our measure of stationarity is hard to interpret: it is bounded from below by zero for perfectly stationary sources, but it can grow arbitrarily large. On finite samples it will never be exactly zero even for perfectly stationary data, since fluctuations in the moment estimates introduce differences across epochs. However, the value of the objective function is an important ingredient for interpreting an SSA solution. For instance, in order to determine an adequate number of stationary directions in a data-driven manner, one can increase the number of stationary sources until the non-stationarity reaches a certain critical threshold. In order for this work, one needs to normalise the objective function value such that it becomes comparable across different numbers of dimensions, epochs and sample sizes.

In this section, we show that our measure of stationarity is equivalent to a test statistic: under the null hypothesis of perfectly stationary Gaussian data, its distribution converges to a χ^2-distribution [Blythe et al. (2012)]. The test statistic compares two competing Gaussian models for the data: the simple model H_0 that presuppose

that each epoch has the same mean and covariance; and the alternative model H_A, under which each epoch has its own set of parameters. This is a test for *model comparison*: it tells us whether we should reject the simple model H_0 in favour of the more complex H_A. Since the two models are nested (i.e. H_0 is a special case of H_A), and since we obtain the parameters by maximum likelihood, this hypothesis can be tested in the likelihood ratio testing framework [Neyman and Pearson (1933)], where the test statistic is the log of the ratio of the likelihood of the data under H_0 and H_A.

More formally, let $X_1, \ldots, X_n$ be d-variate random variables modeling the distribution of the data in each of the n epochs. After centering and whitening, the hypothesis can be written as follows.

$$H_0 : X_1, \ldots, X_n \sim \mathcal{N}(0, I)$$
$$H_A : X_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \ldots, X_n \sim \mathcal{N}(\mu_n, \Sigma_n)$$

Let $\mathcal{X}$ be our data set that is segmented into epochs $\mathcal{X} = \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n$ and let $L_{H_0}(\mathcal{X})$ and $L_A(\mathcal{X})$ be the likelihood of the data under H_0 and H_A respectively where the parameters are the maximum likelihood estimates. Then the likelihood ratio test statistic $\Lambda(\mathcal{X})$ is,

$$\Lambda(\mathcal{X}) = -2 \log \frac{L_{H_0}(\mathcal{X})}{L_A(\mathcal{X})}, \tag{3.8}$$

which under the null hypothesis H_0 is approximately χ^2 distributed with $k = \frac{1}{2}nd(d+3)$ degrees of freedom. It can be shown that by applying the variance-stabilizing transformation,

$$\Lambda'(\mathcal{X}) = \sqrt{2\Lambda(\mathcal{X})},$$

we arrive at a test statistic that is approximately Gaussian distribution with mean $\mu = \sqrt{2k-1}$ and unit standard deviation $\sigma = 1$.

We will now show the equivalence between the test statistic Equation 3.8 and the SSA objective function. For the sake of simplicity, let us assume that all epochs are of equal size $m = |\mathcal{T}_1| = \cdots = |\mathcal{T}_n|$. The estimates of the epoch means and covariance matrices are given by $\hat{\mu}_i = \frac{1}{m}\sum_{x \in \mathcal{T}_i} x$ and $\hat{\Sigma}_i = \frac{1}{m-1}\sum_{x \in \mathcal{T}_i}(x - \hat{\mu}_i)(x - \hat{\mu}_i)^\top$ for all epochs $1 \leq i \leq n$. As in the derivation of the SSA algorithm, we assume that the data is centred and whitened such that $\frac{1}{n}\sum_{i=1}^n \hat{\mu}_i = \underline{0}$ and $\frac{1}{n}\sum_{i=1}^n \hat{\Sigma}_i = I$.

Then the log-likelihood of the data under H_0 is given by,

$$\begin{aligned}
\log L_{H_0}(\mathcal{X}) &= \sum_{i=1}^{N}\sum_{x\in\mathcal{T}_i}\log p_{\mathcal{N}}(x;0,I) = -\frac{1}{2}\sum_{i=1}^{N} md\log(2\pi) + \sum_{x\in\mathcal{T}_i} x^\top x \\
&\overset{(1)}{=} -\frac{1}{2}\sum_{i=1}^{N} md\log(2\pi) + \operatorname{trace}\left[\left(\sum_{x\in\mathcal{T}_i} xx^\top - \hat{\mu}_i\hat{\mu}_i^\top\right) + m\hat{\mu}_i\hat{\mu}_i^\top\right] \\
&= -\frac{1}{2}\sum_{i=1}^{N} md\log(2\pi) + m\hat{\mu}_i^\top\hat{\mu}_i + \operatorname{trace}\left[(m-1)\hat{\Sigma}_i\right] \\
&\overset{(2)}{=} -\frac{1}{2}\sum_{i=1}^{N}\left(md\log(2\pi) + m\hat{\mu}_i^\top\hat{\mu}_i\right) + \left(\sum_{i=1}^{n}(m-1)\right)\operatorname{trace}[I] \\
&= -\frac{1}{2}\sum_{i=1}^{N} md\log(2\pi) + m\hat{\mu}_i^\top\hat{\mu}_i + (m-1)d,
\end{aligned}$$

where equality 1 follows from the identity that $x^\top x = \operatorname{trace}(xx^\top)$ and the equality 2 is a consequence of the fact we have set the average epoch mean and epoch covariance matrix to $\underline{0}$ and I respectively.

The log-likelihood of the data under the alternative hypothesis H_A is given by,

$$\begin{aligned}
\log L_{H_A}(\mathcal{X}) &= \sum_{i=1}^{N}\sum_{x\in\mathcal{T}_i}\log p_{\mathcal{N}}(x;\hat{\mu}_i,\hat{\Sigma}_i) \\
&= -\frac{1}{2}\sum_{i=1}^{N} md\log(2\pi) + m\log\det\hat{\Sigma}_i + (m-1)d.
\end{aligned}$$

Thus the test statistic is,

$$\begin{aligned}
\Lambda(\mathcal{X}) &= -2(\log L_{H_0} - \log L_{H_A}) \\
&= m\sum_{i=1}^{n}\left(-\log\det\hat{\Sigma}_i + \hat{\mu}_i^\top\hat{\mu}_i\right),
\end{aligned}$$

which is equivalent to the SSA objective function in the sense that its value is the same on source estimates up to the multiplication by the epoch sample size m. In this derivation, we assumed that each epoch consists of the same number of samples. If we allow arbitrary sample sizes, then in the test statistic each term in the sum is weighted by the respective epoch sample size.

As we noted earlier, the distribution of the statistic Λ is only approximately χ^2-distributed and, most importantly, this holds only under the assumption that the

data is in fact Gaussian distributed. How useful this test statistic is in practice therefore depends crucially on (a) how well its distribution is approximated by the χ^2 distribution and (b) what happens when the data is non-Gaussian.

A non-parametric approach to testing for stationarity is based on resampling. The basic idea is to generate samples from the objective function on stationary data sets that have the same properties as the input in terms of the shape of the distribution (kurtosis etc.). To this end, we put all data together and divide it *randomly* into epochs of the same size as specified in the application of the SSA algorithm. Since the data points are assigned randomly to epochs, every epoch is sampled from the same distribution. Evaluating the objective function on one such assignment provides us with one sample from the objective function on stationary data of this kind. By doing this repeatedly, we can approximate this distribution using a histogram, of estimate quantiles for statistical testing. In Chapter 5, we compare the parametric and the non-parametric approach to stationarity testing in controlled simulations.

3.3 Spurious stationarity

In Section 3.1, we have seen that we can identify a basis for the true non-stationary subspace: recovering stationary sources is equivalent to projecting orthogonal to the true non-stationary subspace. However, this argument relies on the assumption that *every* contribution from the non-stationary subspace yields a non-stationary latent variable. If we observe a limited amount of variation, then this may not be the case as the following example shows.

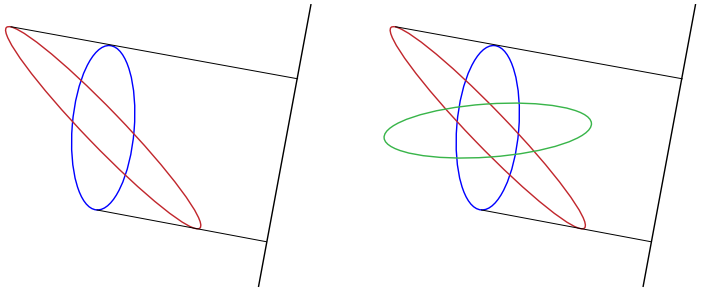

Figure 3.5: Example of a spurious stationary direction. Both plots show the joint distributions of non-stationary latent variables. In the left panel, we have observed two different joint distributions, with the same epoch but different covariance matrices (blue and red contours). There exists a direction on which the projection agrees (black line). This is a spurious stationary direction, because it is located in the non-stationary space. In the right panel, we add a third joint distribution: now there exists no spurious stationary direction.

Imagine, for instance, that we have two non-stationary latent variables and that

we observe two different joint distributions X_1^n and X_2^n, which differ only with respect to their covariance matrix, $\Sigma_1 = \mathrm{cov}(X_1^n)$ and $\Sigma_2 = \mathrm{cov}(X_2^n)$ whereas $\mu(X_1^n) = \mu(X_2^n)$. The left panel of Figure 3.5 shows the contours of the covariance matrices Σ_1 and Σ_2. Note that there exists a direction $w \in \mathbb{R}^2$ (black line) on which the variance of the two random variables is equal: $w^\top \Sigma_1 w = w^\top \Sigma_2 w$. According to our definition of stationarity, this is a stationary direction, even though it lies in the non-stationary subspace: this is a *spurious stationary* direction. Only after we add a third covariance matrix (right panel) does this spurious stationary direction disappear. The existence of spurious stationary directions makes it impossible to identify the true non-stationary subspace. We therefore need a criterion which guarantees that there are no spurious stationary directions.

In order to derive a theoretical result, we consider the *generic case* in order to exclude pathological cases. In other words, we assume that the true covariance and mean of the epochs do not have any special properties, other than being equal on the true stationary coordinates. Under this assumption, we can derive the following bound on the number of necessary epochs.

Proposition 1 Let n be the number of epochs in D dimensions and let $d < D$ be the number of stationary directions. Moreover, let us assume that the mean and the covariance matrix of the non-stationary sources in each epoch is generic, in the sense that with probability one no equation between their elements is fulfilled. Then for any number of epochs n for which holds,

$$n > \frac{D-d}{2} + 1,$$

there exists no spurious stationary direction. In the case where the mean vectors are known to be constant, this bound becomes,

$$n > D - d + 1.$$

□

The rigorous proof relies on results form generic algebraic geometry; it can be found in [Kiraly et al. (2012)]. The geometric intuition is that each available epoch moment reduces the degrees of freedom for a spurious stationary direction by one. The numerator in the first bound therefore stems from the fact that each epoch provides us with two moments. Thus if integrated higher order moments into the SSA algorithm, the number of required epochs would be smaller—under the assumption that they carry information, i.e. variation in the non-stationary directions.

This bound is good news for practical applications: the number of necessary epochs grows only *linearly* with the size of the non-stationary subspace. This is consistent with the results of the simulations presented in Chapter 5.

3.4 Summary and discussion

In this chapter, we have introduced a new type of latent variable model: the observed data is a linear mixture of stationary and non-stationary variables, which are not assumed to be independent. Based on a pragmatic definition of stationarity, we have analysed which parameters are identifiable in principle, and up to what symmetries. We derived a computationally efficient SSA algorithm, which optimises over the special orthogonal group $\mathrm{SO}(D)$ using Lie group properties. The equivalence of the objective function with a statistical hypothesis test for stationarity suggests a normalisation. Finally, we investigated the phenomenon of spurious stationarity: given a set of generic mean vectors and generic covariance matrices, under which condition is there no subspace on which they are identical?

Beyond weak stationarity

The most relevant limitation of the current algorithm is its definition of stationarity, which ignores temporal structure. As long as the power of a source remains constant it is deemed stationary, no matter what happens to the frequency content. However, in many situations it is the variation in the time structure thats is most interesting. Similarly, changes in higher order moments go unnoticed by the current SSA objective. Even though the first two moments are easier to interpret and estimate, there may be situations where changes e.g. in the kurtosis are relevant.

There is a number of strategies for addressing both issues. A straightforward approach for integrating temporal information would be to consider time-lagged covariance matrices in addition to the epoch covariance matrices as in temporal ICA. The advantage of this technique is that most of the optimisation problem's appealing features can be retained. Using higher order information is less straightforward for various reasons. Whereas the Gaussian approximation is a sensible choice for comparing distributions up to their first two moments, it is less clear how to do this for higher moments in a way which corresponds to a natural weighting of the distances in the individual moments. Expansions of the Kullback-Leibler divergence for higher moments already become computationally unwieldy in the case of the third and fourth moment. Non-parametric approaches often do not lend themselves well to optimization, because one can not compute a derivative with respect to a linear projection of the data.

Finding the number of stationary sources

Throughout this chapter, we have tacitly assumed that the true number of stationary sources is known. While this may be true in certain special cases, for most real data sources it is not. Manual approaches of selecting this parameter have their limits. In some scenarios, increasing the number of stationary sources until the estimated

sources become "too non-stationary" by visual inspection may be adequate. Drawing this line, however, is not straightforward using just a subjective assessment of the result. Firstly, the change in correlations is difficult to assess by visual inspection. Secondly, one needs to account for the fact that by increasing the number of sources, more changes in the distribution are inevitable. We outlined a solution to this problem using two kinds of stationarity tests, which are evaluated empirically in Chapter 5. In the next chapter, we present an alternative approach based on an algebraic view. This has the advantage that one can evaluate all possible choices for the number of stationary sources simultaneously by looking at an eigenvalue spectrum.

Design of the epoch structure

The SSA objective function evaluates distribution changes between epochs of a time series. Unless the way in which the data is segmented into epochs is somehow determined by the application context (e.g. aligned to experimental markers or corresponding to meaningful temporal intervals), the setup of the epoch structure is a crucial model parameter. It defines the time scale on which distribution changes are visible to the objective function, and the time points at which they are detectable. In the purely unsupervised case, where one wants to make as little assumptions about the data as possible, a good choice is to use a sliding window instead of a segmentation. In order to increase the temporal resolution, a short window is clearly desirable. Even though this increases the effect of small sample errors, our simulations in Chapter 5 show that until the windows get very short (approximately less than $2D$ samples), the impact on the accuracy is negligible.

Post-processing and pre-processing, interpretation

The result of inverting the SSA model are two multivariate sets of sources: d estimated stationary sources and $D - d$ estimated non-stationary sources, where the latter have been optimised for maximum non-stationarity. As we have seen before, both groups can only be determined up to arbitrary linear transformations. The individual sources, therefore, have no particular meaning; the same is true of the basis of the estimated non-stationary subspace. This makes it difficult to interpret an SSA solution.

There are several ways of making an SSA solution unique, and possibly meaningful, in a post-processing step. In order to allow for univariate interpretation and since independent sources perhaps correspond to underlying latent variables, one can apply ICA to both sets of sources. A way of post-processing that is in line with analysing non-stationarities is applying SSA in deflation mode to both sets of sources. This provides us with two unique sets of sources, each of which is ordered by its degree of non-stationarity.

As in factor analysis [Harman (1976)], in some applications one may be primar-

ily interested in the coefficients of the basis of the non-stationary subspace. For ease of interpretation, it is desirable that each basis vector has only a few large coefficients, so that one can assign (meaningful) latent variables to directions in data space. To that end, there exist a variety of methods for post-rotating a basis to achieve this [Abdi (2003)]; the most prominent is Varimax [Kaiser (1958)], which maximises the variances of the coefficients in each basis vector by application of a rotation.

PCA and ICA are not SSA

PCA, ICA and SSA are similar in that their generative model is a linear mixing of latent variables (see Chapter 2). The crucial distinction lies in the specific assumptions about the mixing matrix and the latent variables. First of all, in the SSA model there are *two groups of sources* with prescribed properties whereas PCA and ICA assume the existence of D *univariate components*. For this reason, it is not straightforward to compare PCA and ICA solutions to an SSA solution. But most importantly, it is in general not possible find the most stationary or the most non-stationary projection by combining PCA or ICA directins in a post-processing step. That is because the criteria of PCA and ICA algorithms are completely unrelated to stationarity resp. non-stationarity.

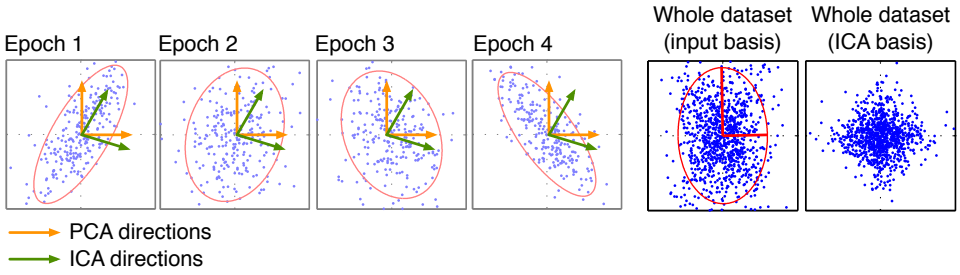

Figure 3.6: Exemplary data set from Chapter 1 with the PCA and ICA solutions.

Figure 3.6 shows the PCA and ICA solutions for the illustrative data set first introduced in Figure 1.3. As we had already seen in Chapter 1, both PCA and ICA yield uninformative bases in the sense that they do not separate the stationary from the non-stationary components. In the two right panels, we see why this is the case. PCA diagonalises the *average covariance matrix*. Since the vertical axis has higher variance and is, on average, uncorrelated with the horizontal axis, PCA does not change the basis (significantly) — the distribution changes remain hidden. ICA, on the other hand, is looking for two independent coordinates. As we can see from the scatter plot of the whole data set, the input coordinates are uncorrelated on average but not independent: the data appears to be generated by a Gaussian mixture model

with (at least) two independent components, and it is these that the ICA algorithm [2] identifies.

[2]Here we have used the JADE algorithm [Cardoso (1999)].

Chapter 4

An algebraic approach

In the previous chapter, we presented an optimisation algorithm for finding the stationary projection. In essence, this algorithm searches the space of all possible projections guided by an objective function that is zero at the optimal solution (see Figure 4.1). In this chapter, we explore a different route: instead of *searching* the space of possible solutions, we directly *solve* the SSA problem algebraically.

This is possible because the set of stationary projections can be described in terms of *equations*: in the optimal stationary coordinates, all epoch mean vectors are *equal* to the null vector and all epoch covariance matrices are *equal* to the identity matrix. More formally, an optimal stationary projection $B^{\mathfrak{s}}$ is defined by the system of linear and quadratic matrix equations,

$$\begin{aligned} B^{\mathfrak{s}}\mu_1 &= 0 \\ &\vdots \\ B^{\mathfrak{s}}\mu_n &= 0 \\ B^{\mathfrak{s}}\Sigma_1(B^{\mathfrak{s}})^\top &= I \\ &\vdots \\ B^{\mathfrak{s}}\Sigma_n(B^{\mathfrak{s}})^\top &= I, \end{aligned}$$

where n is the number of epochs. From this point of view, the SSA objective function can be seen as measuring the extent to which this system of polynomial equations is not fulfilled for a particular projection.

The gist of the algebraic algorithm is that we manipulate this system of equations in a way that reveals the stationary projection. This can be understood in analogy to solving a system of polynomial equations. However, as the coefficients of these equations are derived from epoch moments estimated on finite sets of samples, the exact set of solutions will always be empty in the generic case. To this end, we develop a notion of finding an *approximate linear space of solutions*. Even though this is an approximate solution, it can be found in closed form.

The presented algorithm was originally derived from an algebraic-geometric

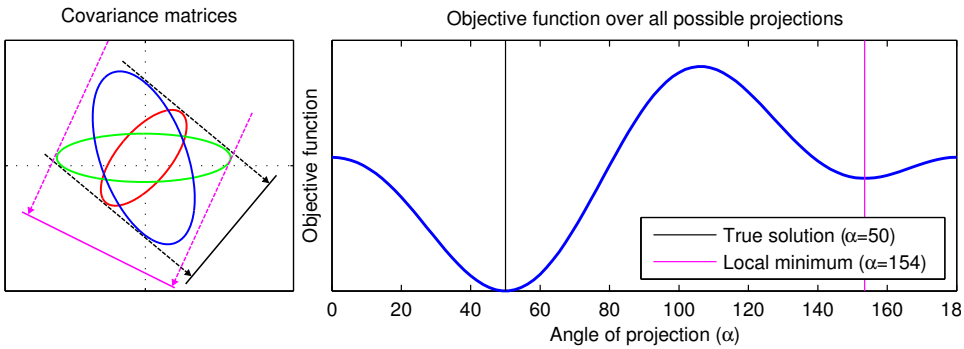

Figure 4.1: Illustration of the optimisation approach to SSA. The left panel shows the contour plots of three sample covariance matrices. The black line is the true one-dimensional subspace on which the projected variances are exactly equal while the magenta line corresponds to a local minimum of the objective function. The right panel shows the value of the objective function over all possible one-dimensional subspaces, parameterised by the angle α to the horizontal axis.

perspective, and previous publications [Kiraly et al. (12), Kiraly et al. (2012)] used this terminology extensively. In this chapter, we attempt a different presentation that is accessible by a wider audience.

The remainder of this chapter is organised as follows. In Section 4.1, we formulate the objective of SSA in terms of a system of polynomial equations. Then, in Section 4.2, we introduce a representation for these polynomials: the coefficient vector space. In Section 4.3, we show how we can find an approximate solution to our set of equations by linear algebra in the coefficient vector space. The connection to algebraic geometry is briefly described in Section 4.4. In the last Section, 4.5, we summarise our findings and discuss the relative merits of the algebraic algorithm and the prerequisites for using the algebraic approach in other circumstances in machine learning.

4.1 From moments to polynomials

In this section, we start by reformulating the objective of the SSA algorithm in terms of polynomials. Until Section 4.3, we will assume that these polynomials are *exact*, in the sense that they have been derived from the exact (true) epoch moments, and not estimates thereof. This is of course never true in practice, where moments are estimated from finite and noisy samples. However, this assumption allows for an easy introduction of the main concepts before we turn to an approximate treatment.

According to the SSA mixture model, an optimal $\mathfrak{s}$-projection $B^{\mathfrak{s}}$ can be defined as being dual to the non-stationary subspace. Alternatively, we can define the optimal stationary projection in terms of the moments in the stationary coordinates (under the assumption that there are spurious stationary directions). That is, an

optimal $\mathfrak{s}$-projection fulfills the equations,

$$\begin{aligned} B^{\mathfrak{s}}\mu_i &= B^{\mathfrak{s}}\mu_j \\ B^{\mathfrak{s}}\Sigma_i(B^{\mathfrak{s}})^\top &= B^{\mathfrak{s}}\Sigma_j(B^{\mathfrak{s}})^\top \qquad \forall\, 1 \leq i \neq j \leq n. \end{aligned}$$

This set consists of $n(n-1)$ equations between all pairs of epochs, which are redundant because of transitivity; equivalently, we can compare each projected pair of moments against one reference epoch. As in the optimisation-based algorithm, we chose the average over all epoch as the reference, which has mean $\bar{\mu}$ and covariance matrix $\bar{\Sigma}$. Thus we characterise the set of solutions by a number of matrix equations that is linear in the number of epochs,

$$\begin{aligned} B^{\mathfrak{s}}\mu_i &= B^{\mathfrak{s}}\bar{\mu} \\ B^{\mathfrak{s}}\Sigma_i(B^{\mathfrak{s}})^\top &= B^{\mathfrak{s}}\bar{\Sigma}(B^{\mathfrak{s}})^\top \qquad \forall\, 1 \leq i \leq n. \end{aligned}$$

Note that this set of equations is still not independent, because the equality of $n-1$ projected moments with the average epoch implies the equality of all. As we have seen before, if we centre the mean vectors such that $\bar{\mu} = \underline{0}$ (translating all the data by the same vector does not change solution) and choose a basis in which the average covariance matrix is the identity, $\bar{\Sigma} = I$, (whitening), we arrive at a set of $2(n-1)$ homogeneous matrix equations which equivalently describe the set of solutions,

$$\begin{aligned} B^{\mathfrak{s}}\mu_i &= \underline{0} \\ B^{\mathfrak{s}}\left[\Sigma_i - I\right](B^{\mathfrak{s}})^\top &= \underline{0} \qquad \forall\, 1 \leq i \leq (n-1), \end{aligned} \tag{4.1}$$

where $B^{\mathfrak{s}}$ is a matrix with linearly independent rows.

The optimal stationary projection $B^{\mathfrak{s}}$ is of course only unique up to arbitrary linear transformations of its image, which leaves the stationary nature of the estimated sources unchanged. In the following, we therefore state the goal as finding a *basis* for the space of $\mathfrak{s}$-projections.

Definition 1 The d-dimensional *linear subspace of all $\mathfrak{s}$-projections* $S \subset \mathbb{R}^D$ is the dual space of the true non-stationary subspace, or in other words, the row space of an optimal stationary projection $B^{\mathfrak{s}}$. □

In the next step, we translate the system of Equations 4.1 into a set of polynomials. This establishes the left hand sides of these equations as objects in their own right, which are amenable to algorithmic algebraic manipulations.

Definition 2 A *monomial* m in D variables $T_1, \ldots, T_D$ is a formal product,

$$m = T_1^{\alpha_1} T_2^{\alpha_2} \cdots T_D^{\alpha_D},$$

which we abbreviate as $m = T^\alpha$, where $\alpha \in \mathbb{N}_0^D$ is the vector of exponents. A monomial is therefore uniquely identified by its exponent vector. The *degree* of a

monomial $\deg(m)$ is the sum of its exponents,

$$\deg(T^\alpha) = \sum_{i=1}^{D} \alpha_i.$$

A *polynomial* p in D variables over $\mathbb{C}$ is a sum of finitely many monomials,

$$p = \sum_{\alpha \in \mathbb{N}_0^D} a_\alpha T^\alpha,$$

where $a_\alpha \in \mathbb{C}$ is the coefficient of the monomial T^α, and only finitely many of the coefficients are nonzero. A polynomial is *homogeneous* if all monomials with nonzero coefficients have the same degree. The *evaluation of a polynomial* p on a vector of values $w \in \mathbb{C}^D$ is a number in $\mathbb{C}$ written as $p(w)$ and is the result of replacing all formal variables by values,

$$p(w) = \sum_{\alpha \in \mathbb{N}^D} a_\alpha w_1^{\alpha_1} \cdots w_D^{\alpha_D} \in \mathbb{C}.$$

□

Using this definition, we write the left hand sides of the Equations 4.1 as polynomials, where the variables $T_1, \ldots, T_D$ correspond to the coordinates in the input vector space and the coefficients are given by the elements of the mean vectors and covariance matrices. Thus we obtain a set of linear polynomials $\ell_1, \ldots, \ell_{n-1}$ and quadratic polynomials $q_1, \ldots, q_{n-1}$, using formal vector notation as,

$$\begin{aligned} \ell_i &= \begin{bmatrix} T_1 \cdots T_D \end{bmatrix} \mu_i \\ q_i &= \begin{bmatrix} T_1 \cdots T_D \end{bmatrix} \Sigma_i \begin{bmatrix} T_1 \\ \vdots \\ T_D \end{bmatrix} \qquad \forall\, 1 \leq i \leq (n-1). \end{aligned}$$

Note that these polynomials are homogeneous of degree one and two, respectively.

Let us now turn to its *set of solutions*. In the language of algebraic geometry, the set of solutions to a set of polynomials is their *vanishing set*.

Definition 3 The *vanishing set* V of a set of polynomials $p_1, \ldots, p_m$ in D variables is their common set of solutions in $\mathbb{C}^D$,

$$\mathrm{V}(p_1, \ldots, p_k) = \left\{ x \in \mathbb{C}^D \mid p_1(x) = \cdots = p_m(x) = 0 \right\}.$$

□

Whereas by definition S is a subset of $\mathrm{V}(\ell_1, \ldots, \ell_{n-1}, q_1, \ldots, q_{n-1})$, the vanishing set need not be to the space of $\mathfrak{s}$-projections S. Not only is the null vector an element of the vanishing set (but not an $\mathfrak{s}$-projection), but by translating

the *matrix* Equations 4.1 into polynomials, we have implicitly dropped the constraint that the $\mathfrak{s}$-projections must be elements of a d-dimensional linear space. The goal of our algorithm is to find the d-dimensional linear subspace contained in $\mathrm{V}(\ell_1, \ldots, \ell_{n-1}, q_1, \ldots, q_{n-1})$. Whether this subspace is unique depends on the observed input polynomials $\ell_1, \ldots, \ell_{n-1}, q_1, \ldots, q_{n-1}$. This question is addressed in the next section.

4.2 The vector space of polynomials

The representation in which we compute with polynomials is the *vector space* of their *coefficients*. For example, a homogeneous quadratic polynomial p in two variables,

$$p = \alpha_{11} T_1^2 + \alpha_{12} T_1 T_2 + \alpha_{22} T_2^2,$$

is represented by its coefficient vector, $\vec{p} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{22} \end{bmatrix}^\top \in \mathbb{C}^3$, where the coordinates correspond to the monomials of degree two in two variables, T_1^2, $T_1 T_2$ and T_2^2. Figure 4.2 illustrates how the equality of two projected covariance matrices is translated into a polynomial equations that is embedded in coefficient vector space.

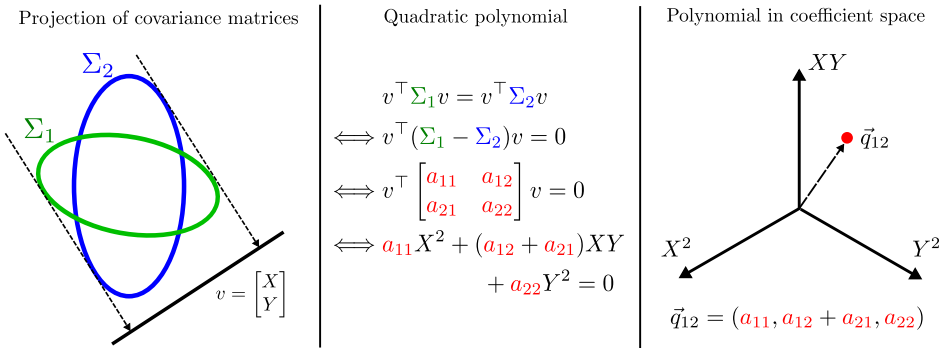

Figure 4.2: The left panel shows the contours of two covariance matrices along with the direction v on which their projections are equal. The middle panel shows that v is the solution to a quadratic equation, and the right panel illustrates how this equation becomes a vector in the coefficient space C_2 of quadratic homogeneous polynomials.

By representing polynomials in terms of coefficient vectors in a finite dimensional vector space we lose structure. The coefficient vector space is not isomorphic to the polynomial ring: whereas the addition of polynomials translates into the addition of coefficient vectors, this is not true for the multiplication of polynomials. In particular, the product of two polynomials of degree at least one is not embeddable in the same coefficient vector space, because it increases the degree of the monomials. However, the coefficient vector space is a suitable representation for our algorithm, as we will see later. In particular, the coefficient vector space comes with a

natural distance measure between polynomials, which allows us to treat uncertainty in the polynomial coefficients using familiar tools from machine learning.

The representation of a polynomial in the coefficient vector space of degree k depends on an arbitrary ordering of all monomials of degree k. For computational reasons, we adopt the lexicographical ordering, because it lends itself to fast enumeration and selection of certain monomials.

Definition 4 The *coefficient vector space* of homogeneous polynomials in D variables of degree k is a $\mathbb{C}$-vector space denoted by C_k. A polynomial p of degree k is represented in terms of its coefficient vector $\vec{p} \in C_k$ where the monomials are ordered *lexicographically*. In this ordering, monomial $M_1 = T_1^{\alpha_1} \cdots T_D^{\alpha_D}$ appears after $M_2 = T_1^{\beta_1} \cdots T_D^{\beta_D}$, denoted by $M_1 \succ M_2$, if there exists a variable T_l such that $\alpha_l > \beta_l$ and for all variables with smaller index $T_1, \ldots, T_{l-1}$ it holds that $\alpha_i = \beta_i$. □

For example, the lexicographical ordering of the monomials in three variables of degree two is given by,

$$T_1^2 \succ T_1T_2 \succ T_1T_3 \succ T_2^2 \succ T_2T_3 \succ T_3^2.$$

Let us now have a closer look at the coefficient vector space. The dimension of C_k is equal to the number of monomials in D variables of degree k. This is equal to the number of ways that we can distribute the exponents among the D variables.

Proposition 2 The dimension of the coefficient vector space C_k is given by,

$$\dim C_k = \Delta(k, D) = \binom{k + D - 1}{k}.$$

□

Proof. A monomial of degree k in D variables is product of k variables chosen with repetition from the set $\{T_1, \ldots, T_D\}$. The number of different monomials is therefore equivalent to the number of multisets of cardinality k chosen from a set of cardinality D. □

An important sub-vectorspace

The algebraic algorithm hinges on the fact that our input polynomials are elements of a certain sub-vectorspace of the coefficient vector space (in the exact, noise-free, case). For the linear polynomials derived from the mean vectors this is clear: by definition, $\vec{\ell}_1, \ldots, \vec{\ell}_{n-1}$ are dual to S and therefore elements of a $(D-d)$-dimensional linear subspace. In the following, we see that there is such a linear subspace in the corresponding coefficient vector space for polynomials of arbitrary degree.

Definition 5 The set of coefficient vectors corresponding to the polynomials of degree k vanishing on S is denoted by C_k^S. □

For degrees $k > 1$, the number of dimensions of the coefficient vector space is not immediately obvious but straightforward to calculate, as the next result shows.

Proposition 3 The set of coefficient vectors $C_k^S \subset C_k$ of homogeneous polynomials of degree k vanishing on S is a linear subspace of C_k with dimension,

$$\dim C_k^S = \Delta(k, D) - \Delta(k, d).$$

□

Proof. C_k^S is a linear subspace of C_k because it is closed under linear combination: any linear combination of polynomials in C_k^S is of degree k and also vanishes on S, hence it is also an element of C_k^S. In order to get its dimension we do a linear change of the polynomials variables as follows. Let $b_1, \ldots, b_d \in \mathbb{R}^D$ be an orthonormal basis for S and let $b_{d+1}, \ldots, b_D$ be an orthonormal basis for its complement in $\mathbb{R}^D$. We now consider polynomials in this basis, i.e. we write them in terms of new variables $T'_1, \ldots, T'_D$ where each variable T'_i corresponds to the coordinate along b_i. Then for any polynomial $\vec{p} \in C_k^S$, the coefficients of all monomials that contain only the variables $T'_1, \ldots, T'_d$ are zero, because p vanishes on S, and this is a sufficient condition for membership in C_k^S. Hence the dimension of C_k^S is the total number of monomials $\Delta(k, D)$ minus the $\Delta(k, d)$ monomials that are fixed to zero. □

4.2.1 Generic polynomials and identifiability

In the previous sections, we formulated the SSA problem in terms of polynomials; introduced the coefficient vector space, and showed that the (exact, noise-free) input polynomials are elements of certain subvectorspaces. In order to derive theoretical results, we need to assume a *generative model* for the input polynomials. In essence, this will allow us to exclude pathological cases in our arguments.

According to the SSA model, we assume that the input polynomials $\ell_1, \ldots, \ell_{n-1}$, $q_1, \ldots, q_{n-1}$ do in fact vanish on the space of $\mathfrak{s}$-projections S. Apart from that, the polynomials should be *generic*, in the sense that they have no special property. For a rigorous definition, we refer to the supplemental material of [Kiraly et al. (2012)]. Intuitively, we can think of a generic polynomial as a polynomial-valued random variable that has no "algebraic property" with probability one, apart from those properties that are implied by the assumption that it vanishes on S. Algebraic properties of a polynomial are those that can be expressed in terms of polynomial equations in the coefficients of the polynomial. As a concrete example, any random variable that has a positive continuous probability density over the coefficient vector space C_k^S yields a generic polynomial in our context.

Using the concept of generic polynomials, we can investigate under which condition the true space of $\mathfrak{s}$-projections S can be uniquely identified, given the input polynomials. This means: under which condition is S the *only* d-dimensional linear subspace in the vanishing set $\mathrm{V}(\ell_1, \ldots, \ell_{n-1}, q_1, \ldots, q_{n-1})$?

Proposition 4 Let $\ell_1, \ldots, \ell_{n-1}$ and $q_1, \ldots, q_{n-1}$ be generic linear and quadratic polynomials derived from n epochs which vanish on the d-dimensional linear subspace of $\mathfrak{s}$-projections S.

Then S is the the unique d-dimensional linear subspace in the vanishing set $\mathrm{V}(\ell_1, \ldots, \ell_{n-1}, q_1, \ldots, q_{n-1})$ if the number of epochs is at least,

$$n - 1 \geq \frac{D - d + 1}{2}.$$

□

The proof can be found in [Kiraly et al. (2012)]. Intuitively, one can think of each of the $2(n-1)$ generic polynomials as reducing the degrees of freedom by one, until S remains as the only d-dimensional linear subspace.

An important consequence of the genericity assumption is that the coefficient vectors in C_1^S and C_2^S corresponding to the observed polynomials are each linearly independent with probability one.

Proposition 5 The coefficient vectors corresponding to the generic homogeneous polynomials $p_1, \ldots, p_m$ of degree k with $m \leq \dim C_k^S$ are linearly independent with probability one. □

Proof. If $\vec{p}_1, \ldots, \vec{p}_m$ are linearly dependent then there exists a vector $\vec{p}_i$ such that $\vec{p}_i \in \mathrm{span}\{\vec{p}_j\}_{j \neq i}$. However, in order for this to be the case, at least one equation between the coefficients of $\vec{p}_i$ and the other vectors need to be fulfilled, which holds with probability zero. □

4.2.2 Generating polynomials of higher degree

In the last section, we have shown that under genericity assumptions, our (exact, noise-free) linear and quadratic input polynomials are linearly independent elements of the subspaces C_1^S and C_2^S in coefficient vector space respectively. However, they may not provide us with a basis (which is essential for the algorithm to work), as the number of dimensions might be larger than the number of observed polynomials. In fact, this is likely to be the case in practice. As we had seen before, the number of dimensions of the coefficient vector space grows rapidly with the number of dimensions of the data space; recall that

$$\dim C_2^S = \Delta(2, D) - \Delta(2, d) = 0.5[D(D+1) - d(d+1)].$$

Even for a moderate number of input dimensions, one would thus need a large number of epochs $n \geq \dim C_2^S + 1$ in order for the derived polynomials to span C_2^S (see Figure 4.3). This is the problem addressed in this section.

For reasons that will become clear later, we focus on the quadratic input polynomials and show how we can generate a larger set of polynomials of higher degree

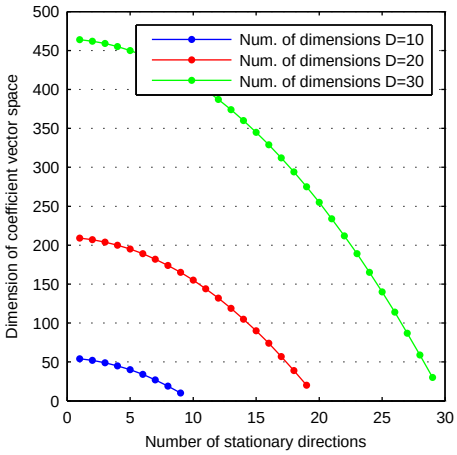

Figure 4.3: Growth of the number of dimensions of the vector space C_2^S of quadratic homogeneous polynomials vanishing on S (vertical axis) for different numbers of input dimensions D (blue, red, green curve) and number of stationary directions d (horizontal axis). The number of dimensions of C_2^S is quadratic in D.

$k \geq 2$ which (a) is guaranteed to span C_k^S and (b) has the same vanishing set as the input.

Recall that a vector $x \in \mathbb{C}^D$ is an element of the vanishing set $\mathrm{V}(q_1, \ldots, q_{n-1})$ if and only if x is a solution to the system of polynomial equations,

$$\begin{aligned} q_1(x) &= 0 \\ &\vdots \\ q_{n-1}(x) &= 0. \end{aligned}$$

Since these equations are homogeneous quadratic, multiplying each with *all* non-zero monomials M of a fixed degree k' does not change the set of solution, i.e. the extended system,

$$\begin{aligned} Mq_1(x) &= 0 \\ &\vdots \\ Mq_{n-1}(x) &= 0 \; \forall M = T_1^{\alpha_1} \cdots T_D^{\alpha_D} \text{ with } \sum_{i=1}^{D} \alpha_i = k', \end{aligned}$$

has the same vanishing set as $\mathrm{V}(q_1, \ldots, q_{n-1})$. Moreover, every choice of monomial M yields a distinct set $\{Mq_1, \ldots, Mq_{n-1}\}$ of polynomials of degree $\deg M + 2$. Thus if we multiply by *all* monomials of a certain degree, we obtain a larger set of polynomials which is equivalent to the input in terms of its vanishing set.

Proposition 6 Let $q_1, \ldots, q_{n-1}$ be generic homogeneous quadratic polynomials vanishing on the set of $\mathfrak{s}$-projections S. The set of polynomials obtained by multiplication with all monomials of fixed degree,

$$\mathcal{P}_k = \{ Mq_1, \ldots, Mq_{n-1} \mid M \text{ is a monomial of degree } k-2 \},$$

contains $(n-1)\Delta(k-2, D) = |\mathcal{P}_k|$ distinct polynomials of degree k. □

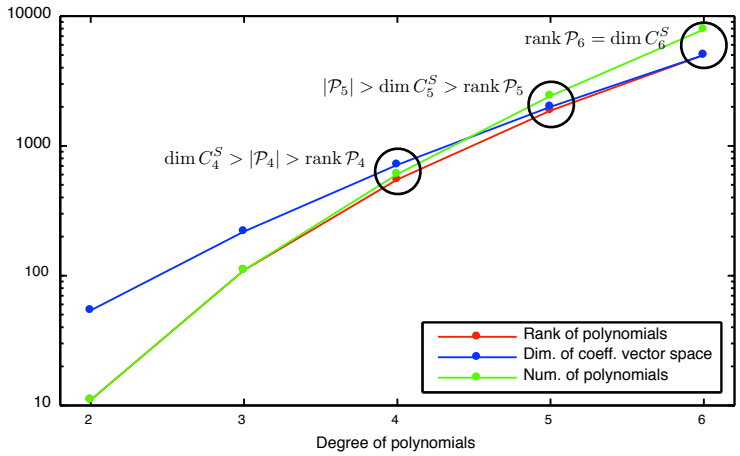

Figure 4.4: Growth of the rank of the set of polynomials $\mathcal{P}_k$ (red curve) in coefficient vector space, the number of polynomials $|\mathcal{P}_k|$ (green curve), and the dimensions of the coefficient vector space C_k^S (blue curve) with the degree k (horizontal axis), for 11 generic polynomials in $d = 10$ dimensions, of which one is stationary. For degrees smaller than three, $\dim C_k^S$ is much larger than the number of polynomials; $\mathcal{P}_5$ contains more polynomials than $\dim C_5^S$, but the rank is smaller. Here we need to multiply up to $k = 6$ to ensure that $\mathcal{P}_6$ spans C_6^S.

As the number of polynomials in $\mathcal{P}_k$ grows with the degree k, so does the number of dimensions of the vector space C_k^S. The number of polynomials $|\mathcal{P}_k|$ grows faster than the number of dimensions of the corresponding coefficient vector space,

$$\lim_{k\to\infty} \frac{|\mathcal{P}_k|}{\dim C_k^S} = \lim_{k\to\infty} \frac{(n-1)\Delta(k-2, D)}{\Delta(k, D) - \Delta(k, d)} > 1.$$

However, the obtained set of polynomials is in general *not linearly independent*, i.e. $\operatorname{rank} \mathcal{P}_k < |\mathcal{P}_k|$. Figure 4.4 shows experimental results on simulated data. While it can be shown that there always exists a k such that the coefficient vectors of $\mathcal{P}_k$ span C_k^S [Kiraly et al. (2012)], in practice it is very important to know what that number is, because the computational complexity of the overall algorithm is dominated by the number of dimensions of the coefficient vector space. Thus the central question is: how does the rank of a set of generic polynomials grow with the degree k' of monomials by which we multiply?

This question is related to Fröberg's conjecture [Froeberg and Hollman (1994), Froeberg (1985)] from algebraic geometry. If it holds, then we have an exact condition on the minimum necessary degree, given by the following result.

Conjecture 1 (based on Fröberg's conjecture) Let $\mathcal{P}_2 = \{q_1, \ldots, q_{n-1}\}$ be generic homogeneous quadratic input polynomials vanishing on the linear subspace of $\mathfrak{s}$-projection S with $n \geq D + 1$ and let $\mathcal{P}_k$ be derived from $\mathcal{P}_2$ by multiplication with all monomials of degree $k - 2$,

$$\mathcal{P}_k = \{Mq_1, \ldots, Mq_{n-1} \mid M \text{ is a monomial of degree } k-2 \} .$$

Then the smallest degree $k^* \geq 2$ such that the coefficient vectors corresponding to the polynomials $\mathcal{P}_{k^*}$ span the subspace $C^S_{k^*}$ in coefficient vector space is the index of the first non-positive coefficient a_{k^*} in the power series,

$$\sum_{\ell=1}^{\infty} a_\ell t^\ell = \frac{\prod_{i=1}^{n-1}(1-t^2)}{(1-t)^D} - \frac{1}{(1-t)^d}$$

□

For a proof see the supplemental material of [Kiraly et al. (2012)]. Since the coefficient of this power series are straightforward to calculate, using this result we can compute the minimum necessary degree at a negligible computational cost.

Algorithm 2 Enlarge set of polynomials by increasing the degree until it spans the corresponding vector space of polynomials.

1: **function** MULTIPLYUP($q_1, \ldots, q_{n-1}$)
2: Initialise $k \leftarrow 2$ to the degree of the input polynomials.
3: **while** $a_k > 0$ **do** ▷ Coefficient of the power series in Conjecture 1
4: Increment degree $k \leftarrow k + 1$.
5: **end while**
6: Multiply input polynomials by all monomials of degree $k - 2$,

$$\mathcal{P}_k \leftarrow \{mq_1, \ldots, mq_{n-1} \mid m \text{ monomial with } \deg m = k-2\} .$$

7: **return** $\mathcal{P}_k$
8: **end function**

This result leads to Algorithm 2 which computes for any generic input the polynomials $\mathcal{P}_k$ spanning the corresponding coefficient vector space C^S_k. The computational cost is negligible, since enumerating all monomials of a certain degree (Line 6) is linear in the number of all monomials.

Note that naïve approaches for generating the set $\mathcal{P}_k$ are infeasible in almost all practically relevant cases. First of all, verifying whether a set of vectors are linearly

independent entails applying a rank-revealing algorithm (e.g. Gaussian elimination) which is at least quadratic in the number of dimensions; and coefficient vector spaces are notoriously high-dimensional. Secondly, simply generating a "very large number" is of course not guaranteed to work in general but also leads to exceedingly high dimensional coefficient vector spaces, in which subsequent steps of the algebraic algorithm need to work. As some of these steps are quadratic in the number of dimensions, it is important to keep the degree as small as possible.

4.3 An approximate algebraic algorithm

In the previous sections we have laid the groundwork for our algorithm: we reformulated the SSA problem in terms of a set of exact polynomial equations which are represented as vectors in the coefficient vector space. Under genericity assumptions, we have seen that the (exact, noise-free) coefficient vectors are linearly independent elements of subspace of known dimensions (but unknown basis). Then we showed how we can generate a larger set of polynomials $\mathcal{P}_k$ of higher degree k such that its coefficient vectors are guaranteed to *span* the desired subspace in coefficient vector space.

So far, we have assumed that the input polynomials are known exactly, in the sense that they were derived from the true epoch mean and covariance. In practice, where cumulants are estimated from finite and noisy data, this is not the case. This means that there exists no exact solution; a true $\mathfrak{s}$-projection matrix $B^{\mathfrak{s}}$ will not make the projected moments exactly equal. Hence the vanishing set of the input polynomials does not contain the true set of $\mathfrak{s}$-projections S, but only the null vector. We therefore need to find an approximate solution.

It is here the vector space view on polynomials comes in handy, because it comes with a natural geometric interpretation which includes a *distance* between polynomials. As we have seen before, all quadratic homogeneous polynomials that vanish exactly on S lie on a linear subspace C_2^S of the coefficient vector space. In the approximate case, we will think of our input as lying approximately on C_2^S.

The overall strategy of our algorithm is to algebraically manipulate our set of polynomials C_k^S until we obtain linear polynomials that have approximately the same vanishing set. From linear polynomials, we can readily obtain the space of $\mathfrak{s}$-projections S as their approximate d-dimensional dual. This procedure can be seen as *solving* the SSA problem algebraically. We proceed step-wise: starting at the minimal necessary degree (determined by Proposition 1), with each step we reduce the degree of our set of polynomials by one.

Approximate division by a single variable

The fundamental idea behind the step-wise reduction of degree can be loosely described as "approximate division of a set of polynomials by a single variable". That is,

we go from a set of polynomial $\mathcal{P}_k$ of degree k to set of polynomials $\mathcal{P}_{k-1}$ of degree $k-1$ that has approximately the same vanishing set as $\mathcal{P}_k$.

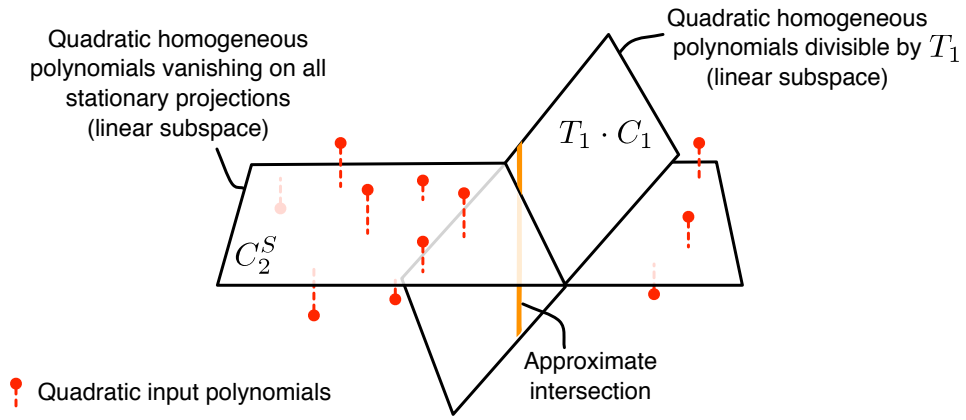

Figure 4.5: Illustration of the approximate division of a set of polynomials by a single variable T_1. The planes illustrate the *true* sub-vectorspaces, not our estimate from the input polynomials.

The choice of variable by which we divide is arbitrary; by our genericity assumption any variable T_i is not a basis element of S. In the following we fix it to T_1. As we have seen before, the polynomials in $\mathcal{P}_k$ lie approximately on $C_k^S \subset C_k$. Since we have applied Algorithm 2, we can also assume that they are an approximate basis. Let us now consider another subspace of C_k, namely the polynomials of degree k that are divisible by T_1. They also form a linear subspace spanned by all monomials that are divisible by T_1. Let us denote this subspace by $T_1 \cdot C_{k-1}$. What this abuse of notation indicates is that $T_1 \cdot C_{k-1}$ is in fact isomorphic to the coefficient vector space C_{k-1} of lower degree. Figure 4.5 shows a cartoon illustrating the relationship between these two linear subspaces in the quadratic case.

Approximate division of $\mathcal{P}_k$ by T_1 consists of two steps. In the first step, we compute an *approximate* basis for the *intersection* of C_k^S and $T_1 \cdot C_{k-1}$. More specifically, we compute a basis for the approximate intersection of C_k^S (for which we only have an approximate basis in terms of the input polynomials) and $T_1 \cdot C_{k-1}$ whose basis is known exactly by definition. We find the elements of this basis as linear combinations of the polynomials in $\mathcal{P}_k$ by minimizing the L^2-distance to $T_1 \cdot C_{k-1}$ using the singular value decomposition. The elements of this basis are polynomials that are both divisible by T_1 *and* linear combinations of our input, i.e. vanish approximately on S. Let us denote the set of these polynomials by $\mathcal{P}'_{k-1}$, and note that they are still of degree k. In Figure 4.5, $\mathcal{P}'_{k-1}$ is a basis for the one-dimensional approximate intersection shown as the orange line. In the next step, we remove from each polynomial in $\mathcal{P}'_{k-1}$ all terms that are not divisible by T_1. Finally, by dividing each remaining monomial by T_1, we get a set of polynomials of degree $k-1$ called

$\mathcal{P}_{k-1}$, whose elements lie approximately on C^S_{k-1}. This concludes the approximate division of $\mathcal{P}_k$ by T_1.

Note that the choice of variable by which we divide is arbitrary. In fact, we can divide by T_1 in an *arbitrary basis* to obtain a set of polynomials of lower degree that lie approximately on C^S_{k-1}. We exploit this property in the final algorithm to generate a larger set of such polynomials, which we then reduce to a basis by applying PCA, in order to increase the accuracy on noisy data.

The complete algorithm, explained

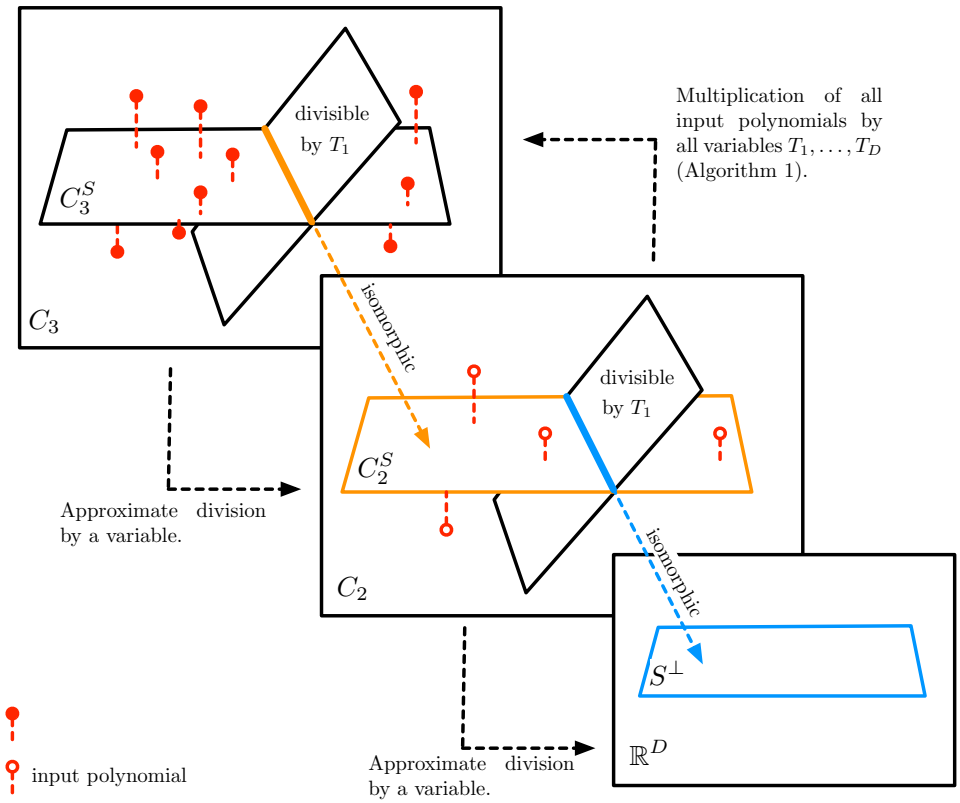

Figure 4.6: Illustration of the algorithm. The quadratic input polynomials are embedded in C_3 (by multiplication with all variables) and lie approximately on C^S_3 (red dots in the top panel). The intersection of C^S_3 with the subspace of polynomials divisible by T_1 (orange line) is isomorphic to C^S_2 in the vector space of homogeneous quadratic polynomials C_2. By dividing out T_1 again (blue line) we arrive at a basis for the linear polynomials vanishing on S, which is a basis for the orthogonal complement $S^\perp$.

Let us now step through the complete Algorithm 3. Throughout the text we refer to line numbers in that algorithm. Figure 4.6 illustrates the steps of the algorithm,

Algorithm 3 Compute an approximate basis for the d-dimensional space of $\mathfrak{s}$-projections S given generic homogeneous polynomials corresponding to epoch mean and covariance matrices respectively.

1: **function** AlgebraicSSA($\ell_1, \ldots, \ell_{n-1}, q_1, \ldots, q_{n-1}, d$)
2: Compute set of polynomials $\mathcal{P}_k \leftarrow$ MultiplyUp($q_1, \ldots, q_{n-1}$)
3: Let $\pi \leftarrow (1 \cdots D)$ be the cyclic permutation of variables.
4: **while** $k > 1$ **do**
5: Initialise $\mathcal{P}_{k-1} \leftarrow \{\}$
6: **for** $i = 1, \ldots, D$ **do**
7: Permute polynomial variables $\mathcal{P}_k^\pi \leftarrow \{\pi^i(p) \,|\, p \in \mathcal{P}_k\}$
8: Compute a linearly independent set of polynomials $\mathcal{P}'_{k-1}$ in the span of the coefficient vectors in $\mathcal{P}_k^\pi$,

$$\mathcal{P}'_{k-1} \subset \operatorname{span} \mathcal{P}_k^\pi,$$

of size $|\mathcal{P}'_{k-1}| = \dim C_{k-1}^S$ that is closest to the subspace of polynomials divisible by T_1 in terms of the L^2-distance of coefficient vectors (SVD).

9: Remove from each polynomial in $\mathcal{P}'_{k-1}$ all monomials not divisible by T_1 (i.e. set the corresponding coefficients to zero).

10: Divide by T_1, invert the permutation of variables, and add to the set of polynomials of lower degree,

$$\mathcal{P}_{k-1} \leftarrow \mathcal{P}_{k-1} \cup \left\{ \pi^{-i}(p/T_1) \;\middle|\; p \in \mathcal{P}'_{k-1} \right\}.$$

11: **end for**
12: Reduce $\mathcal{P}_{k-1}$ to an approximate basis of C_{k-1}^S using PCA on the coefficient vectors.
13: Let $k \leftarrow k - 1$.
14: **end while**
15: Compute an approximate basis $b_1, \ldots, b_{D-d}$ for the orthogonal complement $S^\perp$ from the coefficient vectors of $\mathcal{P}_1 \cup \{\ell_1, \ldots, \ell_{n-1}\}$ using PCA.
16: **return** $b_1, \ldots, b_{D-d}$
17: **end function**

and the isomorphisms across coefficient vector spaces associated with different degrees.

The main part of the algorithm deals with the quadratic input polynomials; the linear polynomials are added at the end. In the first step, we compute a new set of polynomials C_k^S of possibly larger degree $k \geq 2$ from the input, which is guaranteed to approximately span C_k^S (Line 2). This corresponds to the arrow going from C_2 to

C_3 in Figure 4.6.

The loop starting at Line 4 is to descend the degree from k down to one (linear polynomials). In the nested loop beginning at Line 6, we repeated the approximate division for different permutations of variables to increase the accuracy of our estimate for the basis of C^S_{k-1}. The approximative division by T_1 (under the permutation of variables) is implemented in Lines 8 to 10.

In Line 8, we compute the approximate intersection with the set of polynomials divisible by T_1. This is a quadratic optimisation problem which can be solved by a singular value decomposition (SVD): let Q be a matrix defined as the row-wise concatenation of all coefficient vectors in $\mathcal{P}_k$, where we remove the columns corresponding to monomials not divisible by T_1. Using the SVD, we compute a basis $v_1, \ldots, v_m \in \mathbb{R}^{|\mathcal{P}_k|}$ for the approximate left kernel of Q of dimension $m = \dim C^S_{k-1}$. The elements of this basis provide us with coefficients for m linear independent linear combinations of polynomials in $\mathcal{P}_k$ such that the sum of the squared coefficients of monomials not divisible by T_1 are minimised; this is the set $\mathcal{P}'_{k-1}$. In Lines 9 and 10 this is then simply converted into polynomials of degree $k-1$.

In each step in Line 10 we approximate the basis for the polynomials of lower degree C^S_{k-1} using PCA (Line 12). The quality of this estimate increases with the number of available vectors that lie approximately on C^S_{k-1}. In order to obtain more such coefficient vectors, we repeatedly divide out the variable T_1, but each time under a different permutation of the polynomial variables $T_1, \ldots, T_D$. This corresponds to a basis change in the input space, of which the permutation of variables is computationally attractive when rearranging and ordering polynomial coefficients. Each permutation of variables yields different sets of polynomials, which we combine to an approximate basis of C^S_{k-1} by first inverting the permutation of variables and then performing PCA to reduce the number of basis elements to the correct dimension $\dim C^S_{k-1}$ (Line 10). Finally, in the last step from quadratic to linear polynomials we include the linear polynomials $\ell_1, \ldots, \ell_{n-1}$ obtained from the mean vectors.

Computational complexity

The computational complexity of Algorithm 3 is determined by the singular value decomposition (Line 8) of the coefficient matrix of the highest (initial) degree k, which consists of the row-wise concatenation of the coefficient vectors of the initial set of polynomials $\mathcal{P}^\pi_k$.

The worst-case complexity of computing the full singular value decomposition for a general $r \times c$ matrix is $\mathcal{O}(min\{rc^2, r^2c\})$. The number of columns c of the coefficient matrix is equal to the number of dimensions of the coefficient vector space C_k of homogeneous polynomials in $D-1$ variables,

$$c = \Delta(k, D-1),$$

because we include only those monomials that are not divisible by T_1. This can be written as a polynomial function in the degree k and in the number of input variables D,

$$c(k) = \frac{1}{(D-2)!}(k+1)\cdots(k+D-2),$$
$$c(D) = \frac{1}{k!}(D-1)\cdots(D-2+k),$$

respectively. The number of columns c grows like k^{D-3} in the degree k and like D^{k-1} in the number of input variables D. The number of rows r of the coefficient matrix is equal to the number of polynomials in the set $\mathcal{P}_k^\pi$,

$$r = |\mathcal{P}_k^\pi| = (n-1)\Delta(k-2, D).$$

As for the number of columns, the number of rows r can be written as a polynomial in k and D with leading terms k^{D-2} and D^{k-3} respectively.

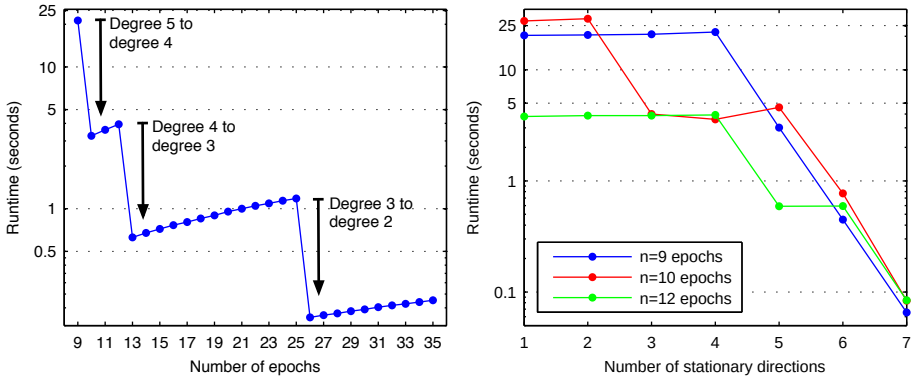

Figure 4.7: The left panel shows the runtime (vertical axis, log scale) of Algorithm 3 for $D = 8$ dimensions and $d = 4$ stationary directions for various numbers of epochs (horizontal axis). Overall, the runtime decreases when the number of epochs grows since the necessary degree becomes smaller, and hence the number of dimensions of the coefficient vector space. The right panel shows the runtime (log scale) of the algorithm for three different numbers of epochs over varying numbers of stationary directions.

For fixed number of input dimensions D, the computational complexity depends on the necessary initial degree k, which depends on the number of polynomials $n-1$ and the number of stationary directions d (see Conjecture 1). The fewer polynomials available, the higher the necessary degree. This is illustrated in the left panel of Figure 4.7. As the number of polynomials grows (horizontal axis), we observe characteristic drops (phase transitions) in the runtime which correspond to decreases in the necessary degree k that translates into fewer number of columns c of the coefficient matrix. For a constant necessary degree k (i.e. between the jumps),

we see that the runtime grows with the number of polynomials, because it increases the number of rows r. Moreover, we observe that for $D = 8$ input dimensions and $d = 4$ stationary directions, $n = 26$ generic epochs are sufficient to span the whole subspace C_2^S of the coefficient vector space.

The right panel of Figure 4.7 shows that the runtime also decreases with the number of stationary directions. That is because the number of dimensions of the linear subspace of polynomials C_k^S vanishing on S decreases, which in turn leads to a lower necessary degree.

4.4 Relationship to algebraic geometry

So far, we have largely eschewed concepts and terminology from algebraic geometry in order to make the presentation accessible by the wider machine learning community. In this section, we briefly establish this connection. We do not attempt an exhaustive or rigorous exposition (see e.g. [Cox et al. (2007)] for a good introduction with a view to computation); the purpose is to point to the main concepts and rephrase the task in the language of algebraic geometry.

Algorithm 3 exploits the algebra-geometry connection: to find a basis for the linear subspace of $\mathfrak{s}$-projections (a geometrical object), we operate on its representations in terms of polynomials (elements of an algebra, the ring of polynomials). Assuming that the quadratic input polynomials $q_1, \ldots, q_{n-1}$ are known exactly, this corresponds to a classical problem in computational algebraic geometry: computing generators for the radical of an ideal.

An ideal is a set of polynomials in the ring $\mathbb{C}[T_1, \ldots, T_D]$ that correspond to a vanishing set in $\mathbb{C}^D$. In Section 4.1 we have defined the vanishing set of a set of polynomials. Moreover, we have observed that the space of $\mathfrak{s}$-projections S is a linear subspace of the vanishing set $\mathrm{V}(q_1, \ldots, q_{n-1})$. Conversely, we can associate an algebraic set $V \subset \mathbb{R}^D$ with a set of polynomials, its ideal. The ideal of a vanishing set is the set of *all polynomials* that vanish on it, denoted by $\mathrm{I}(V)$. A purely algebraic way to define an ideal is in terms of a set of so-called polynomial generators $f_1, \ldots, f_m \in \mathbb{C}[T_1, \ldots, T_D]$. If we think of the generators as polynomial equations defining the vanishing set $\mathrm{V}(f_1, \ldots, f_m)$, then the generated ideal corresponds to the set of all polynomial consequences. These we can derive from the generators by multiplication with all elements of the polynomial ring and all summations of such products. From this point of view, we can interpret Algorithm 2 as generating further elements of the ideal $\langle q_1, \ldots, q_{n-1} \rangle$, up to a certain degree.

Since the space of $\mathfrak{s}$-projections S is a linear space, its ideal is generated by linear polynomials, $\mathrm{I}(S) = \langle b_1, \ldots, b_{D-d} \rangle$. This means that every polynomial that vanishes on S can be written as a sum of polynomials, each having a linear factor corresponding to an element of the orthogonal complement $S^\perp$. If we knew the linear generators $b_1, \ldots, b_{D-d}$, then we could obtain a basis for S. However, since

our input consists of *quadratic* polynomials $q_1, \ldots, q_{n-1} \in \mathrm{I}(S)$, it is not straightforward to extract this basis.

By assumption, the input polynomials are generic and $n \geq D+1$, which implies that the ideal $\langle q_1, \ldots, q_{n-1} \rangle$ contains *all polynomials* of degree at least two which vanish on S. This is a unique subset of the ideal $\mathrm{I}(S)$. Thus, on the level of ideals, the correspondence between the input and S is one-to-one. In fact, $\mathrm{I}(S)$ is the so-called radical ideal of the input $\langle q_1, \ldots, q_{n-1} \rangle$, which is denoted by,

$$\sqrt{\langle q_1, \ldots, q_{n-1} \rangle} = \mathrm{I}(S) = \langle b_1, \ldots, b_{D-d} \rangle.$$

We can think of the radical of an ideal $\mathcal{I}$ as the largest maximal ideal containing $\mathcal{I}$. Solving the SSA problem is therefore equivalent to computing a linear generating set for the radical ideal of the input. This is a classical task in computational algebraic geometry, for which the existence of algorithms has been known for a long time [Hermann (1926)].

However, algorithms for radical computation that are suitable for implementation have only been developed in recent decades. The best known algorithms are those of [Gianni et al. (1988)] (implemented in AXIOM and REDUCE); Macauly 2 provides [Eisenbud et al. (1992)]; [Caboara et al. (1997)] is implemented in CoCoa along with [Krick and Logar (1991)], the modification by [Laplagne (2006)], available in SINGULAR.

All of these algorithms have two points in common. First of all, their computational worst case complexities are doubly exponential in the square of the number of variables [Laplagne (2006)]. Secondly, they can only be applied when the coefficients of the polynomials are known exactly, which is never the case in practical applications of SSA where moments are estimated from data. It is for these two reasons that we cannot solve the SSA problem using off-the-shelf general algorithms for radical computation. Our algorithm addresses a new type of problem in the nascent field of approximate computational algebra [Corless et al. (1995), Stetter (2004), Kreuzer et al. (2009), Heldt et al. (2009)].

4.5 Summary and discussion

In this chapter we have presented an algebraic algorithm for solving the SSA problem. Exploiting the algebra-geometry connection, we do not search the space of all possible projections guided by an objective function, but compute an approximate solution to a system of polynomial equations.

The algebraic approach has several advantages over optimisation-based techniques. First of all, the algorithm has a unique solution (no local minima) which can also be shown to be consistent, i.e. it converges to the true solution as the sample size grows. Moreover, it is straightforward to integrate cumulants of arbitrary

order, as they corresponds to polynomials of higher degree. Parallel to the combination of mean (degree one) and covariance matrices (degree two), the highest moment determines the initial coefficient vector space and the moments of lower order are added during the step-wise reduction of the degree. In Algorithm 3, the mean vectors are added in the last step (Line 15), after the quadratic polynomials have been reduced to an approximate linear basis.

Choosing the number of stationary directions

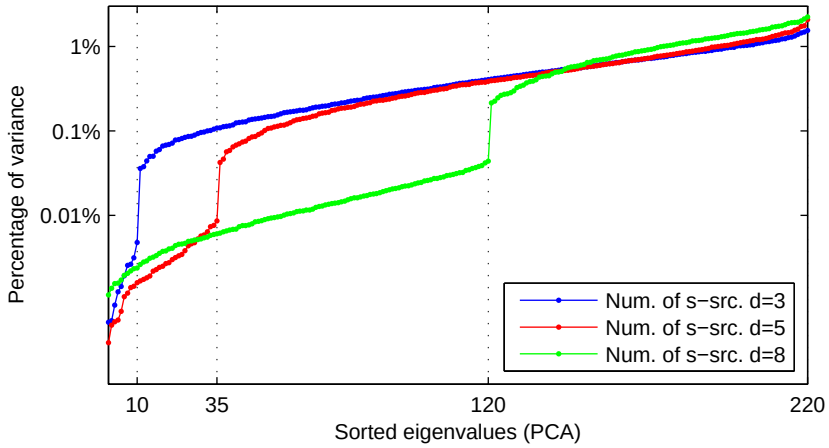

Figure 4.8: PCA eigenvalue spectrum in coefficient vector space C_3 for three synthetic data sets in ten dimensions with varying numbers of stationary sources.

The true number of stationary directions d determines the number of dimensions of the subspace C_k^S in coefficient vector space, on which the input polynomials (multiplied up to the necessary degree k) lie approximately. By estimating the effective number of dimensions of that subspace from the input, we can select an appropriate number of stationary directions. To that end, we inspect the PCA eigenvalue spectrum of the polynomials $\mathcal{P}_k$ in coefficient vector space C_k.

Figure 4.8 shows an example computed on synthetic data. For $d = 5$ stationary directions in ten dimensions, we see that there exists a 35-dimensional subspace that has significantly less variance than the rest of the data. For polynomials of degree three in $D = 10$ dimensions this corresponds to five stationary directions, because a five-dimensional stationary subspace induces a (220-35)-dimensional effective subspace in coefficient vector space,

$$\dim C_3^S = \Delta(3, 10) - \Delta(3, 5) = 220 - 35 = 185.$$

Similarly, the low-variance subspaces of 10 and 120 dimensions (blue and green curve) correspond to stationary subspaces of three and eight dimensions respec-

tively. The advantage of this approach, relative to stationarity testing, is that it allows us to inspect all possible number of dimensions simultaneously by computing a single eigenvalue spectrum instead of running the optimisation procedure for each candidate and then performing time-consuming resampling.

Alleviating the computational cost

The biggest drawback of the algebraic algorithm is the scaling of the computational cost. The singular value decomposition (Line 8 of Algorithm 3) is cubic in the number of dimensions of the coefficient vector space — which grows rapidly with the degree and the number of input dimensions. However, there are several directions which could bring about a dramatic reduction of the computational complexity. First of all, the matrix of coefficient vectors (corresponding to the set $\mathcal{P}_k^\pi$ in Algorithm 3) is very sparse when it is large: while the number of dimensions of the coefficient vector space grows quickly with the degree k, the number of nonzero coefficients (i.e. nonzero entries in each row of the matrix) stays the same. Moreover, the matrix has a special structure which can be exploited. The multiplication of an input polynomial with all monomials of a certain degree (Algorithm 2) essentially sends its coefficients to different places (columns), each corresponding to a row in the matrix. Finally, for the purpose of approximate division by a variable, we do not need to compute all singular values, but only the smallest ones. There exist several algorithms for iterative large-scale sparse structured singular value decomposition [Kaltofen et al. (2007), Park et al. (1999)] which are yet to be explored in this context.

Algebra beyond SSA?

The algebraic approach is not limited to stationary subspace analysis; our algorithm solves a rather general problem: approximating the set of solutions to a system of homogeneous polynomial equations of arbitrary degree by a linear space. Moreover, it would be straightforward to generalise this procedure to find approximate "descriptions" of sets of solutions in terms of polynomials of higher degree or with certain prescribed properties.

The main requirement for a machine learning problem to fit our algebraic approach is that its set of solutions needs to be describable in terms of a system of *approximate* polynomial equations. This is an uncommon formulation, perhaps because so far there have been no efficient ways of solving such problems known to the machine learning community. What is attractive about this framework is that polynomials of arbitrary degree can be combined naturally (e.g. corresponding to certain constraints on the solutions), without suddenly rendering the problem infeasible.

Chapter 5

Simulations and applications

5.1 Introduction

In this chapter, we evaluate the SSA algorithm in simulations and present applications to EEG data analysis.

Since SSA is a new unsupervised learning task, there is no established evaluation strategy. To date, there exist no competing algorithms or benchmark data sets that allow for a relative assessment. Moreover, since SSA is an unsupervised problem there exists no natural quantitative performance measure that is inherently meaningful as e.g. for classification problems. Whether a particular SSA solution is useful is ultimately a qualitative decision, which depends on the application context. This is reflected in our evaluation methodology, which consists of four different angles.

1. *Synthetic data.* An artificially constructed ground truth allows us to quantitatively measure the performance in terms of the deviation from the true solution. We investigate the influence of key parameters, the relative efficacy of the algebraic approach, and two methods for stationarity testing. The results serve as a basic validation and provide insights for the domain of applicability and parameter choice in practice.

2. *Comparison against PCA and ICA on real data.* Even though PCA and ICA solve a different problem than SSA, in certain situations they may yield a similar results. For a large EEG data set, we establish that neither PCA nor ICA yield a basis that is useful for understanding distribution changes. This underlines the original contribution of the SSA approach.

3. *Real data with ground truth.* In two situations from EEG analysis where an expected results is induced by an experimental paradigm, we show that SSA successfully identifies the corresponding neural components. The results are evaluated qualitatively, in terms of the frequency content and the scalp maps, and quantitively using the correlation with an experimental stimulus.

4. *Indirect evaluation in a classification setting.* In the context of brain-computer-interfacing, we demonstrate that in certain cases, SSA can remove harmful

non-stationary brain sources. The efficacy of this approach is reflected in lower misclassification rates on the test set.

The remainder of this chapter is organised along these lines. In the next Section 5.2, we present the results on simulated data. The applications to EEG analysis are contained in Secton 5.3. In the last Section 5.4, we discuss our findings and point to future work motivated by the numerical results.

5.2 Simulations on synthetic data

Before we present results, we consider two main ingredients: (a) a method for generating synthetic data that provides us with a ground truth and allows us to vary the parameters of interest, and (b) a measure of deviation between the true and the found solution.

Data generation

The data is generated exactly according to the SSA model. That is, we directly sample the data for each epoch. In doing so, we assume that we have chosen the segmentation into epochs optimally in the sense that we extract the maximum possible amount of information from the data without any redundancy (e.g. overlaps introduced by a sliding window). This is unrealistic; however, any other way of dividing our synthetic data into epochs is equally arbitrary and introduces further parameters which make the results harder to interpret.

For each epoch, we randomly sample the mean and the covariance of the sources, $\mu_1, \ldots, \mu_n$ and $\Sigma_1, \ldots, \Sigma_n$, such that the moments are identical along the first d coordinates. The elements of the mixing matrix A are drawn uniformly at random from the interval $[-0.5, 0.5]$. In all our simulations, we focus on the more difficult case where there is no information in the mean, i.e. the mean of the non-stationary sources is constant in each epoch. If we include information in the mean, the contribution of the new SSA algorithm becomes somewhat unclear: the separation of stationary and non-stationary sources using differences in their mean can be achieved analytically by a simple matrix inversion.

The marginal covariance of the stationary sources is fixed to the identity I. For each epoch, the variance of the non-stationary sources is sampled uniformly over the intervals $(1, \alpha)$ and $(1/\alpha, 1)$ with $\alpha > 1$. That is, each non-stationary source is at most α times larger or α times smaller than the stationary sources. The covariance between the two sets of sources is chosen by drawing random mixing coefficients.

In order to investigate the more realistic scenario of non-gaussian data (heavier tails, outliers), we sample Gaussian data $X \sim \mathcal{N}(0, 1)$, exponentiate it and preserve the sign: $X' = \mathrm{sign}(X)|X|^k$, where the parameter k controls the degree of the

non-gaussianity; we report Pearson's kurtosis instead of k. The data is then linearly transformed and translated to have the chosen epoch mean and covariance.

A performance measure for SSA

As we have seen in Chapter 3, only the true non-stationary subspace can be identified from the mixed signals. A natural performance measure for SSA is therefore the deviation between the true and the found non-stationary subspace. The alternative, a measure based on the level of stationarity of the estimated stationary sources, is more difficult to define because even for an ideal separation, the estimated epoch mean and covariance matrices will not be exactly equal. The expected deviation from equality in the case of ideal separation depends on properties of the data, such as the number of samples or the heaviness of the tails (see Section 5.2.2 for a more in-depth discussion). Since these are precisely the parameters of interest in our simulations, it is more convenient to adopt a performance measure based on the deviation from the true subspace.

Following [Meinecke (2011)] we define a *subspace error* based on the *principal angles* between two subspaces. Let U and V be two linear subspaces of the vector space $\mathcal{V}$ with $\dim U \leq \dim V$. The first principal angle θ_1 is defined as the smallest angle between any two vectors $u \in U$ and $v \in V$,

$$\cos\theta_1 = \min\left\{ \frac{u^\top v}{\|u\|\|v\|} \middle| u \in U, v \in V \right\} = \angle(u_1, v_1),$$

where the vectors u_1 and v_1 that minimise the angle are called the principal vectors. The remaining principal angles $\theta_2, \ldots, \theta_{\dim U}$ are found recursively in the orthogonal complement of the previous principal vectors,

$$\cos\theta_{k+1} = \min\left\{ \frac{u^\top v}{\|u\|\|v\|} \middle| u \in U, u \perp u_1, \ldots, u_k, v \in V, v \perp v_1, \ldots, v_k \right\}.$$

Thus the principal vectors in each subspace are mutually orthogonal. If a principal vector θ_k is zero then U and V share a common subspace and $u_k, v_k \in U \cap V$. Hence the dimension of the intersection of U and V is equivalent to the number of zero principal angles. The principal angles and vectors can be found using the singular value decomposition (SVD). Let the columns of the matrices U' and V' be orthonormal bases for the spaces U and V respectively. From the singular value decomposition $\hat{U}\Sigma\hat{V}^\top = (U')^\top V'$, we obtain the principal angles from the singular values, $\cos\theta_k = \Sigma_{kk}$ and the principal vectors are the columns of $\hat{U}$ and $\hat{V}$.

To quantify the difference between two k-dimensional subspaces spanned by the columns of $A, B \in \mathbb{R}^{d\times k}$, we define the subspace error as the average $\sin^2$ of the

principal angles $\theta_1, \ldots, \theta_k$ between them,

$$\mathcal{E}(A, B) = \frac{1}{k} \sum_{i=1}^{k} \sin^2(\theta_i).$$

This error is zero when the true subspaces are identical and one if they are orthogonal to each other. The subspace error can be interpreted in terms of the sources. It is the percentage of the signal power that a non-stationary sources loses if it is projected from the true onto the estimated non-stationary sources. Conversely, for the estimated stationary sources $\hat{s}^{\mathrm{s}}(t) = \hat{P}^{\mathrm{s}} x(t)$ this means that $\mathcal{E}(\hat{A}^{\mathrm{n}}, A^{\mathrm{n}})$ is the percentage of non-stationary signal power that they contain.

However, the maximum possible subspace error depends on the number of dimensions of the two subspaces in relation to the total number of dimensions, i.e. on the number of dimensions in which they can possibly differ. For example, two nine dimensional subspaces U and V in ten dimensions have a maximal subspace error if there exists a one-dimensional subspace which is contained in U and in $V^{\perp}$ or vice versa. In this case, exactly one of the principal angles is $\frac{1}{2}\pi$ and all others are zero, so that the subspace error is 1/9. In general, two k-dimensional subspaces can only differ by a $d - k$ dimensional subspace, so that the maximum subspace error is $\min\left\{1, \frac{d-k}{k}\right\}$. This worst case is a very special case, in the sense that unless we specifically construct it, any set of basis vectors is expected to have a lower subspace error. As a less pathological baseline, in each simulation we compare against random projections [Rahimi and Recht (2008)].

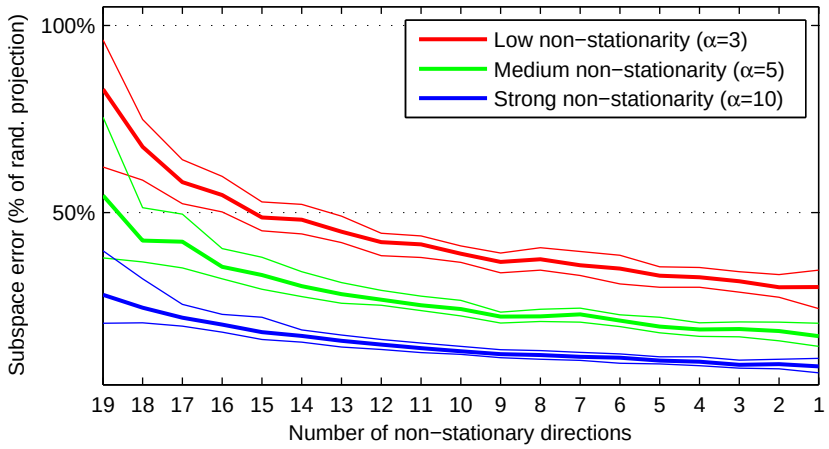

Figure 5.1: Performance for varying numbers of non-stationary directions (horizontal axis) measured in terms of the subspace error relative to random projections. The total number of dimensions is fixed to 20.

5.2.1 Dimensions, sample size, epochs, kurtosis

In this section we present the results of our simulations. Unless otherwise stated, each plot shows median subspace errors over 100 random realisations of the datasets, with error bars extending from the 25% to the 75% quantile.

In the first set of experiments, we investigate the influence of the number of stationary dimensions. The total number of dimensions is set to 20, the number of epochs is 20 and in each epoch we observe 100 samples with non-stationarity in the covariance matrices of degree $\alpha = 3$. Figure 5.2 shows the result. In order to obtain comparable results across different dimensionality, we report the subspace error relative to random projections. We see that in terms of the subspace error, larger non-stationary subspaces are more difficult to find. This is not only a numerical effect: as we have kept the number of distinct epochs constant, the relative amount of available information (i.e. observed variation in the distribution of the non-stationary sources) is smaller for the larger subspaces, which makes them more difficult to identify.

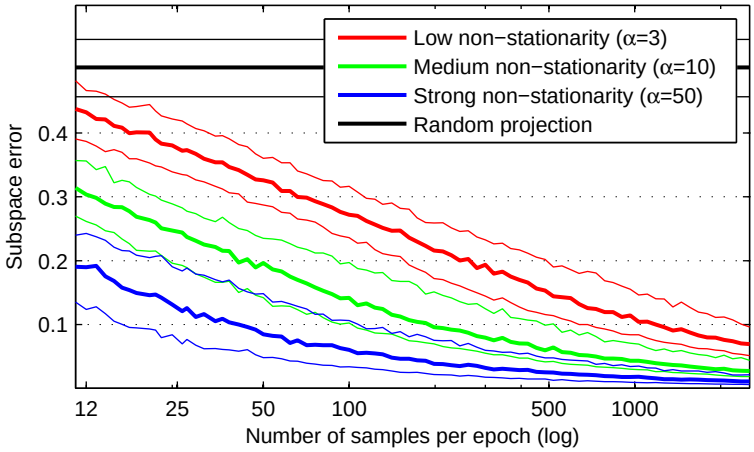

Figure 5.2: Influence of the epoch sample size (horizontal axis) on the performance of SSA, measured in terms of the subspace error (vertical axis).

In the next step, we analyse how the error of SSA depends on the number of available samples in each epoch. The setup is somewhat artificial: in practice, we divide a fixed number of samples into epochs; we can usually only vary the number of samples in each epoch by changing the overall segmentation, which changes the true distribution of the data in each epoch. However, in order to isolate the effect of small sample errors, we vary the number of samples available in each epoch from a fixed epoch source distribution. The total number of dimensions is 10 and the number of stationary directions is 5. Figure 5.2 shows the results. We see that even for epochs as small as eleven samples, SSA can still find a projection that is significantly better than a random projection. This suggests that small sample er-

rors average out across epochs to some extent, an effect that we also observe in the next set of simulations. Moreover, we see that the effect of estimation errors in the epoch moments is relative to the strength of the non-stationarity. This makes sense: a strong change in variance is less easily masked by small sample fluctuations than a weaker non-stationarity.

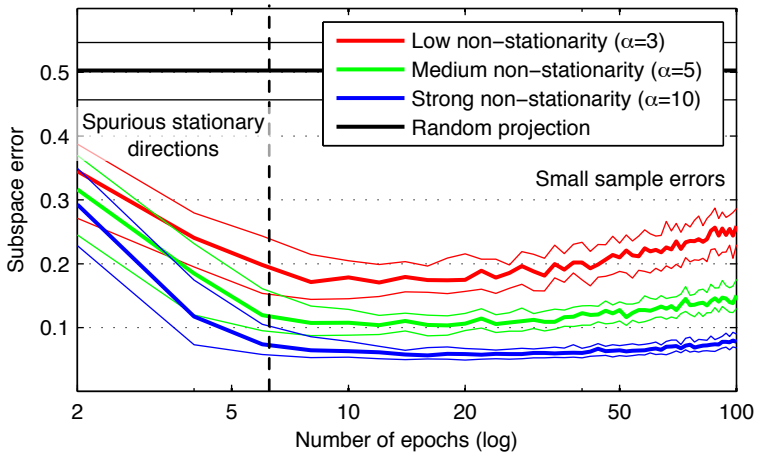

Figure 5.3: Influence of the number of epochs (horizontal axis) with distinct moments on the performance of SSA, measured in terms of the subspace error (vertical axis). The total number of samples is fixed to 1100.

Now we consider a different scenario: the total number of samples is fixed to 1100 and we vary the number of epochs with random source covariance matrix (five stationary and five non-stationary directions). This means that for two epochs, their sample size will be 550 each, whereas for the largest number of epochs (100), each contains only 11 samples. In other words, we trade small sample errors for the number of distinct distributions, or accuracy of information for amount of information. While this is an artificial scenario (we can never vary the amount of information in a dataset), it is directly related to an important practical problem, namely the design of the epoch structure. One usually faces the dilemma of wanting a fine-grained epoch setup, to capture as much variation as possible and increase the temporal resolution, while controlling the detrimental effect of small sample estimates.

The results shown in Figure 5.3 convey a positive message: only as the epochs become extremely small (11 samples in 10 dimensions) does the performance deteriorate markedly. As we have seen before, small sample errors average out across epochs, and at the same time, SSA benefits from further variation in the non-stationary space. In the opposite case, few large epochs, we observe the effect of spurious stationary directions; the subspace error reaches the plateau exactly at the number of epochs given by our theoretical result (see Section 3.3).

All simulations presented so far have one unrealistic aspect in common: Gaus-

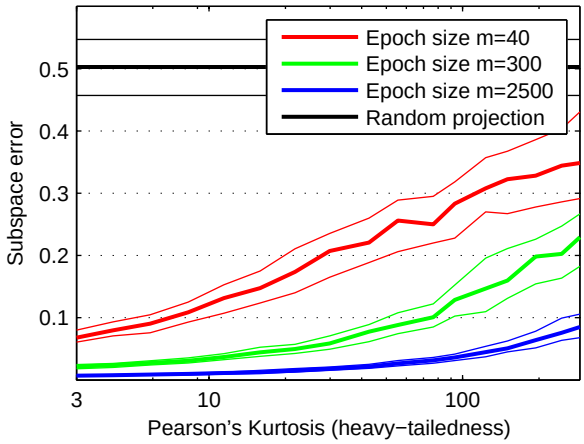

Figure 5.4: Influence of the fatness of the tails of the sources (horizontal axis) on the performance of SSA, measured in terms of the subspace error (vertical axis). The source data is sampled from a super-gaussian with Pearson's kurtosis shown on the horizontal axis.

sian sources. Real data, however, is notoriously non-gaussian, apart perhaps from pure noise components; real data has heavier tails and is almost always contaminated by outliers. Moreover, interesting components are usually distinctively non-gaussian, which is reflected by the fact that many algorithms explicitly search for components with non-Gaussian properties, e.g. ICA [Hyvarinen (1999)], projection pursuit [Huber (1985), Friedman and Tukey (1974)] or non-gaussian component analysis [Blanchard et al. (2006)]. Outliers and heavy tails lead to stronger fluctuations in finite samples between epochs, even for stationary sources, leading to higher estimation errors of the true non-stationary subspace. The absolute scale of the errors we have reported is therefore probably too optimistic for most real data sets[1].

Therefore, in the final set of simulations, we investigate how the error behaves as we fatten the tails of the source distribution. As before, the total number of dimensions is ten with five stationary directions, we have 20 epochs and the non-stationarity level is set to $\alpha = 10$. Figure 5.4 shows the results over various epoch sample sizes. Interestingly, the performance does not deteriorate rapidly as the kurtosis grows. As one would expect, small sample sizes strongly exacerbate the influence of outliers.

[1]This is in line with the aim of these simulations, which is to show the relative performance across parameters. Any assumption about the source distribution is somewhat arbitrary and introduces a systematic bias.

5.2.2 Testing for stationarity

In Section 3.2.1, we have seen that the SSA objective function can also be derived as a likelihood ratio test [Neyman and Pearson (1933)]. In this section, we evaluate the effectiveness of this test and compare it against a data-driven nonparametric test based on resampling.

The value of the SSA objective function is a measure of stationarity. However, for a data set at hand, its value alone does not provide a clear indication of whether the data is truly stationary, in the sense that its epochs have been sampled from distributions with the same mean and covariance matrix. That is because small sample errors introduce differences in moment estimates between epochs, even for a data that has been sampled from a stationary model.

Therefore, one way of making sense of the objective function is to compare its value against its *distribution* on truly stationary data, and decide for non-stationarity if it exceeds a chosen quantile (critical value). In terms of hypothesis testing, stationary data is our null hypothesis H_0 which we reject at a certain (one-sided) confidence level.

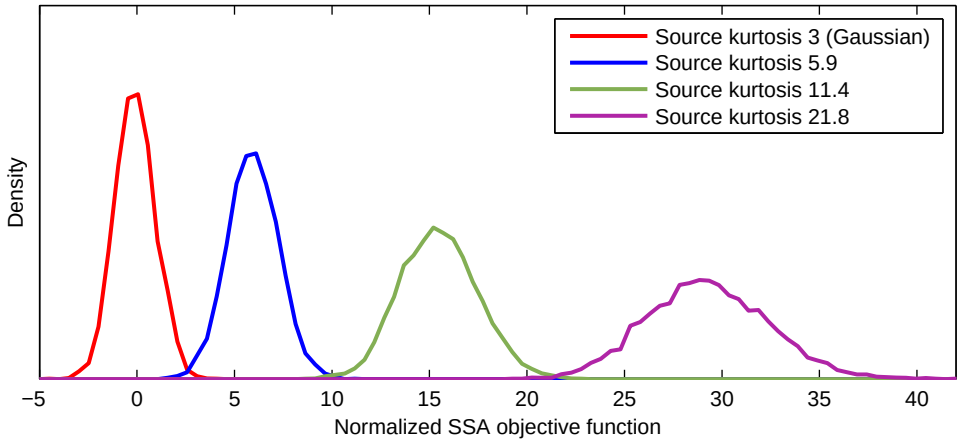

Figure 5.5: Distribution of the normalised SSA objective function for stationary sources with varying kurtosis estimated from random samples.

In Section 3.2.1, we mentioned that the distribution of the normalized objective function,

$$\sqrt{2\Lambda(\mathcal{X})} - \sqrt{nd(d+3) - 1},$$

approximately follows a standard normal distribution on stationary data with Gaussian sources, where Λ is the unnormalized objective function, n is the number of epochs and d is the number of dimensions. Figure 5.5 shows estimates of the distribution of the normalised objective function on simulated stationary data in five dimensions of 20 epochs, each with 100 samples, over 5000 random realisations.

For Gaussian sources, the empirical distribution of the normalised objective resembles a normal density, but it is not normal according to the Kolmogorov–Smirnov test. However, even for moderate levels of kurtosis, the distribution moves away from $\mathcal{N}(0,1)$. Thus a stationarity test based on quantiles of the χ^2 distribution is bound to perform poorly on almost all real data sets, in that it has a high false positive rate (type I error).

However, for non-Gaussian data the distribution of the objective function under H_0 is in general unknown. Let us therefore take a closer look at the resampling approach (see Chapter 3). In short, we put the data from all epochs together, randomly divide it into epochs, on which we then evaluate the SSA objective function. Each random shuffling provides us with one sample from the distribution of the SSA objective function on stationary data. By doing this repeatedly, we obtain a set of samples from which we can estimate a critical value (e.g. the 95% quantile) for testing.

As an illustration, we compare the performance of the χ^2 test for stationarity with the test based on resampling on a synthetic data set. To this end, for each degree of non-gaussianity, we randomly generate 100 stationary and 100 non-stationary data sets and apply both test strategies at the 95% confidence level; the parameters of the data set are the same as the one used for Figure 5.5, the level of non-stationarity is low ($\alpha = 2$). In light of our previous observations, we are mainly interested in the false positive rate (FPR).

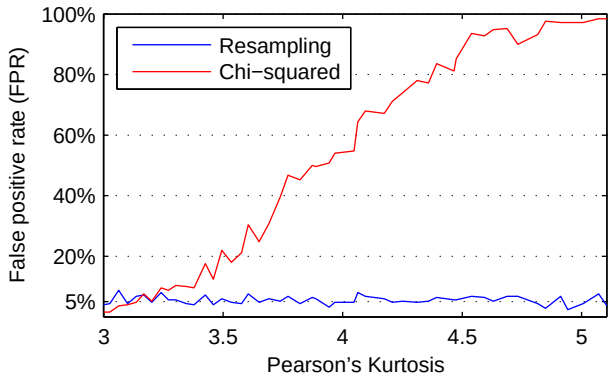

Figure 5.6: Comparison of two strategies for stationarity testing: comparing against quantiles from the standard normal distribution (the theoretical limit) vs. quantiles estimated from resampling the data set.

Figure 5.6 shows the result. For Gaussian sources (with Pearson's kurtosis equal to 3), the false positive rate of both tests corresponds exactly to the chosen confidence level of 95%. However, already for moderate deviations form normality, the increasing likelihood of outliers leads to rapidly rising false positive rate of the parametric test; at kurtosis 5, it has become completely useless, which is in line with our observation in Figure 5.5. On the other hand, the quantiles estimated from 100 re-

samples of the shuffled data set are remarkable reliable: the false positive rate stays at 5% while the true positive rate (not shown here) is constantly 100%. Of course, the performance of the resampling test also depends on several factors (number of samples, type of stationary distribution, etc.), and we do not have theoretical guarantees yet. Even so, resampling can be a viable approach in practice.

5.2.3 Algebraic SSA vs. optimisation

Comparing the performance of the algebraic and the optimisation-based SSA algorithm is not straightforward: for a given dataset, the former has a unique solution computed in a fixed amount of time whereas the latter is initialised randomly and stops when a somewhat arbitrary convergence criterion has been met. Moreover, for "standard" setting of these convergence criterion, the runtime of both algorithms differ markedly for a wide range of setting. As we had seen before, the runtime of the algebraic algorithm grows rapidly, in particular when the number of available epochs is not very large.

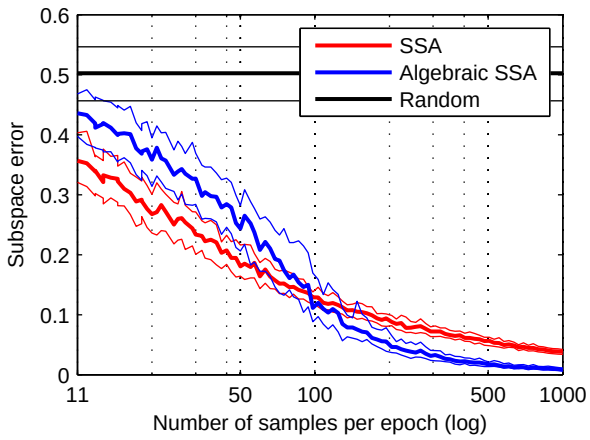

Figure 5.7: Comparison of the algebraic and the optimisation-based SSA algorithm w.r.t. the epoch samples size.

It is for these reasons that we do not attempt an exhaustive comparison, the results of which would be difficult to interpret. Instead, we focus on one case where the runtime of both algorithms is comparable and illustrates a general trend. In this simulation, we use ten dimensional data (five stationary directions), a moderate level of non-stationarity ($\alpha = 5$), and vary the number of samples in each of the 20 epochs.

Figure 5.7 shows the results: until the epochs have reached a critical size of 100 samples, the optimisation approach outperforms the algebraic algorithm. The reason for this is that the algebraic algorithm is more prone to small sample errors than optimisation-SSA. That is because the coefficient vector space that it operates in has

much higher dimension than the data space, in which the optimisation takes place. However, as the small sample errors get weaker in magnitude (on large epochs), it clearly outperforms the standard SSA algorithm. That is because the gradient-based optimisation converges slowly due to the flatness of the objective function around the true solution.

5.3 Application to EEG analysis

In this second part of this chapter, we apply SSA to data from electroencephalography (EEG) [Berger (1929), Caton (1875)] recordings. EEG data analysis is a prime application domain for SSA, because the EEG potential is well approximated by a linear transformation of fluctuations of macroscopic potentials in the cortex. This has lead to the tremendous success of linear methods for the analysis of brain activity (see e.g.[Parra et al. (2005)]).

First of all, we show that PCA and ICA do not find non-stationary brain sources on a data set from brain-computer-interfacing (BCI), and that the most non-stationary task-locked components correspond to the neural response to the task. Next, we investigate the role of non-stationarities as one of the possible causes behind deteriorating performance of BCIs over time. We do this by removing certain non-stationary directions in a pre-processing step, which for some subjects leads to a significant improvement of the error rate. In the last part of this section, we demonstrate that SSA recovers a certain type of non-stationary brain source (auditory steady state evoked responses) that has been induced by an experimental paradigm. Here we can measure the effectiveness using the correlation with the stimuli. The analysis shows that both ICA and PCA fail to recover the desired source.

5.3.1 Electroencephalography

The electroencephalography (EEG) [Berger (1929), Caton (1875)] is an electric potential[2] measured on the surface of the scalp. The EEG is caused both by brain activity and other physiological sources, e.g. nearby muscles.

The following overview is largely based on [Kandel et al. (2000)]. The cortical source of the EEG is the synchronous activity of *neurons*, which are often considered as the atomic units of the cortex in models of the brain. The human brain consists of around 10^{10} neurons, each of which is connected to its neighbouring neurons by a *dendritic tree* and one *axon*. In a neuron, information is processed and transmitted in the form of electric potentials. Every neuron maintains a *membrane potential*, i.e. a voltage difference across its plasma membrane, which is controlled by the opening and closing of ion channels through a complex electrochemical mechanism. A

[2]More precisely, it is a difference in potential between an electrode in the scalp and a reference electrode, often positioned on the nose or ear lobes.

neuron receives electrochemical stimulation from other neurons via its dendrites; if the membrane potential exceeds a certain threshold, it is released as an *action potential* which travels along the axon to other neurons. This effect is called *spiking*. When large, spatially aligned neuron populations spike synchronously, their current flow combines to produce macroscopic field potentials, which are measurable on the surface of the scalp.

The EEG signal not only reflects the fluctuations of the field potentials generated inside the brain, but also non-neural artefacts, which are of physiological or technical origin. Prominent physiological artefacts result from muscle activity (e.g. face, tongue, eye blinks and movement), heart beat and pulse. Technical artefacts are often generated by the power supply or other equipment. The signal power of artefactual sources is usually much stronger than the brain sources. Removing artefacts is a key step in EEG signal processing. For an extensive review of artefact rejection methods see [Fatourechi et al. (2007)].

There are two general strategies for removing artefacts: discarding subsets (trials) of the data with strong artefact activity, or removing artefactual sources by regression and substraction [Croft and Barry (2000)], projection or spatio-temporal filtering. The first strategy leads to a loss of data points while the second reduces the number of dimensions and, if successful, preserves the relevant neural signal. A common approach to artefact removal, which we will adapt in this chapter, is based on independent component analysis [Makeig et al. (1996), Jung et al. (2000)]. In a first step, the data is decomposed into independent components which are then individually assessed to identify the artefactual components, often in a semi-automatic way. Rejecting independent components is appealing, because it ensures that one does not remove information contained in dependencies and artefacts are usually considered to be independent of cortical activity.

The manual classification into artefact/non-artefact is based on a number of heuristics, that take into account the frequency content, the scalp pattern, the time course and sometimes also the correlation with auxiliary measurements. For an extensive evaluation of the standard manual artefact removal protocol, we refer the reader to [McMenamin et al. (2010)].

For our purpose, we used an ICA basis computed by the TDSEP algorithm [Ziehe and Müller (1998b)], which was preceded by a PCA pre-processing to 99% of the variance (reducing the number of dimensions from 88 to approximately 40). Each component was inspected manually. No information other than the frequency spectrum, the scalp plot and the time course was available during artefact classification.

EEG-based Brain-Computer-Interfacing

The goal of brain-computer-interfacing (BCI) [Dornhege (2007), Vidal (1973)] is to transmit information directly from a human brain to a computer, circumventing

peripheral nerves or muscles. This is achieved by voluntary changes in the brain state, that can be detected by a computer and turned into a control signal.

The most popular BCI paradigm is based on modulations of the sensorimotor rhythm (SMR), also called μ-rhythm. In most subjects, the SMR can be observed in cortical EEG after appropriate signal processing. The SMR can be used for brain-computer-interfacing because it is modulated by the neural processing of motor commands, which can be controlled voluntarily by most subjects. In particular, the μ-rhythm responds to motor commands that are only *imagined* and not actually executed. This makes it possible to generate a control signal by thought alone.

Imagined movements cause the so-called event-related desynchronisation (ERD) [Pfurtscheller and Lopes da Silva (1999)] of neuron populations in the motor cortex which lead to an attenuation of the SMR. The location of the SMR inside the brain corresponds to the part of the body that is to be moved. Imagined movements of the left and right hand cause ERD in the right and left motor cortex respectively. In order to translate imagined movements into a control signal, the central task is to distinguish between the spatial localisation of changes in the SMR. However, since the SMR has low power relative to other brain sources (e.g. from the visual cortex) and artefactual activity, it is not visible in the raw EEG signal, but requires spatio-temporal filtering to enhance the signal-to-noise ratio e.g. using the common spatial patterns (CSP) algorithm [Koles (1991), Blankertz et al. (2008)].

Dataset and experimental setup

For our comparison to PCA and ICA in Section 5.3.2 and the application to BCI in Section 5.3.3, we use a dataset from a study [Blankertz et al. (2010)] which included only BCI-novices. From this study, we select the 40 subjects (50%) who were recorded in Berlin.

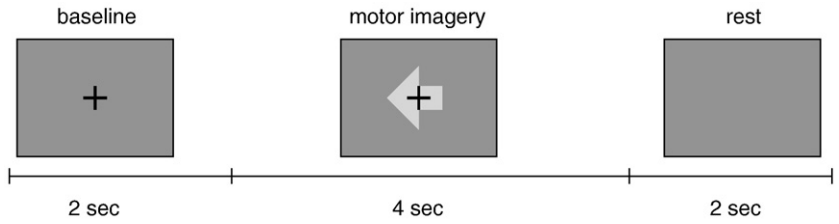

Figure 5.8: Trial structure during calibration: first, a fixation cross appears for 2s, then a visual cue instructs the user to imagine a certain movement for 4s (here: left hand), which is followed by a 2s rest period. Figure taken from [Blankertz et al. (2010)].

The EEG was recorded from the scalp with a multi-channel amplifier (BrainAmp DC by Brain Products, Munich, Germany) using 119 Ag/AgCl electrodes (reference

at nasion; manufacturer EasyCap, Munich, Germany) and sampled at 1000 Hz with a band-pass filter of 0.05 Hz to 200 Hz.

For each subject, a single session was conducted, divided into a calibration and a feedback phase. During calibration, a visual cue prompted the subject to imagine one of the following motor tasks: movement of left hand, right hand, and right foot or both feet movement[3]. In total, 75 trials for each class were recorded; see Figure 5.8 for an illustration of the trial structure. These trials were used to train a classifier on bandpower features extracted with CSP. After the calibration phase, the participants performed the same task with feedback, i.e. a visual cursor indicating the current classifier output.

5.3.2 Comparison to PCA and ICA

Before we look at the results of SSA, we ask the baseline question: could existing methods do the job? Even though PCA and ICA optimise different criteria, and will in general find different bases than SSA, the important question in practice is whether the solutions differ significantly enough to justify a new method.

On EEG data, it is not inconceivable, a priori, that the strongest distribution changes are exhibited by independent brain sources. Moreover, the degree of non-stationarity may be related to the average power. If that were the case, PCA or ICA could provide us with components that are useful for analysing non-stationarities.

Before we proceed to results, let us clarify what it means for a basis to be "useful for analysing non-stationarities". Informally speaking, it implies that it separates the stationary from the non-stationary signal contributions. In the first step, we assess this by considering the degree of non-stationarity of the individual basis elements univariately. The sorted values provide us with a *non-stationarity spectrum* for a basis on a particular data set. In terms of this spectrum, a basis is informative if it decays rapidly and is uninformative if it is flat. In this way, the interpretation is similar to the eigenvalue spectrum of PCA. However, unlike PCA's eigenvalue spectrum, the non-stationarity spectrum does not sum to the same value for all orthogonal bases. Whereas every basis captures the same total variance, the same is not true for the non-stationarities. As we had seen in Chapter 1, a non-stationary data set can seem completely stationary in a basis when the distribution changes are entirely confined to changes in the dependencies.

In a second step, we analyse whether a multivariate approach is necessary to find the most non-stationary *subspace*. To that end, we compare the subspace obtained by combining the components found univariately to a subspace obtained by the joint optimisation of its basis.

[3]The type of foot movement could be chosen by the subject.

Setup and results

The general setup of our analysis is as follows. For each subject, we select all calibration trails from the data set described in the previous section. Firstly, we apply a PCA pre-processing to preserve 99% of the variance, which reduces the number of dimensions from 88 input channels (a subset of electrodes) to around 40. Then we apply a bandpass filter between 0.5 and 45 Hz. No trial is rejected. The SSA epochs are sliding windows of 0.5s length with a 50% overlap. SSA is applied in deflationary mode, unless otherwise stated.

We begin our analysis by looking at the results of a single subject. The non-stationarity spectrum of PCA, ICA and SSA is shown in Figure 5.9. The directions are ordered by degree of non-stationarity (SSA objective function value). For ICA and PCA, we applied the SSA objective function using the same epoch structure as for SSA.

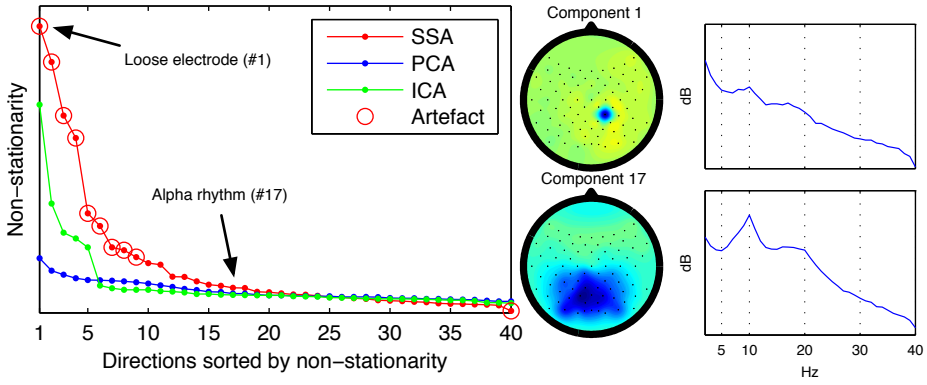

Figure 5.9: Non-stationary spectrum for SSA (red), PCA (blue) and ICA (green) for one exemplary BCI subject. For each method, the found directions are ordered (horizontal axis) by the degree of non-stationarity (vertical axis). The red circle indicates that an SSA direction corresponds to an artefact in the EEG data. The scalp maps visualise the basis of component 1 and 17, next to it are the frequency spectra of the sources.

As we can see, all PCA projections (blue curve) pick up a similar amount of non-stationary contributions. Moreover, we observe that the total non-stationarity captured by PCA is much less than both ICA and SSA. This suggests that signal power is not strongly correlated with non-stationarity. The basis found by ICA (green curve) picks up a similar amount of non-stationarity among its first five directions to the SSA solution (red curve). However, a closer inspection of the most non-stationary sources found by SSA reveals that they are non-neural artefacts (indicated by the red circles).

Since we are interested in finding and analysing brain sources, we remove the artefacts in a pre-processing step. The results in the left panel of Figure 5.9 show that

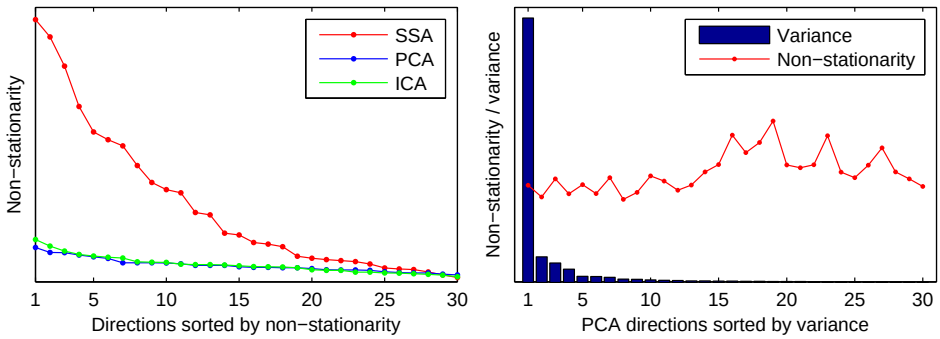

Figure 5.10: Non-stationary spectrum after artefact rejection: degrees of non-stationarity for SSA (red), PCA (blue) and ICA (green) basis. Variance of PCA directions vs. non-stationarity: the blue bars show the eigenvales, the red curve is the degree of non-stationarity.

after artefact rejection, PCA and ICA do not provide a basis that discerns stationary from non-stationary contributions.

Even though PCA did not find non-stationary directions relative to the SSA result, the variance of the PCA components may still be correlated with their strength of non-stationarity. In the panel of Figure 5.10, we plot the eigenvalues against the degree of non-stationarity. However, we see that on this dataset, there is no association between signal power and non-stationarity.

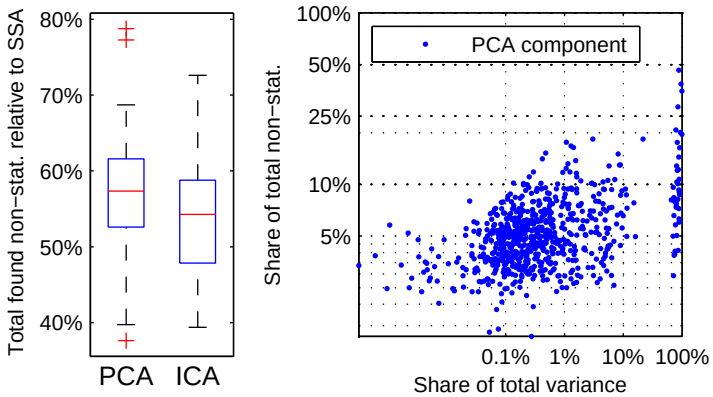

Figure 5.11: The left panel shows the distribution of the total non-stationarity of the PCA and the ICA basis relative to the SSA basis over all subjects. In the right panel, we plot for each PCA component its variance against its degree of non-stationarity.

The analysis of all remaining 39 subjects reveals that these findings are consistent over all subjects. Figure 5.11 shows the results after artefact rejection. In the

left panel, we see that SSA finds significantly more non-stationary components than PCA and ICA over all subjects. The right panel confirms that there is no systematic relationship between the power of a PCA component and its non-stationarity.

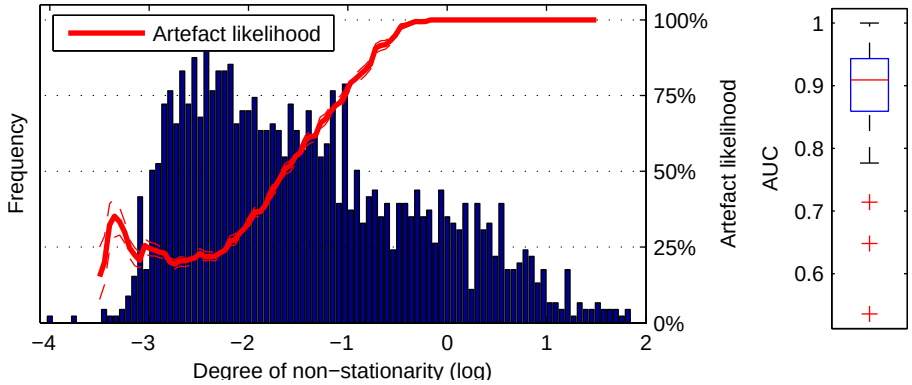

Figure 5.12: Degree of non-stationary (horizontal axis) vs. artefact likelihood (right vertical axis) for all 40 subjects. The plot combines a histogram over non-stationarity strengths (blue bars, left vertical axis) with a curve (red) indicating the artefact likelihood. The artefact likelihood is the percentage of artefactual directions in a centered window of length 0.6

The results for the single subject suggest that there is a relationship between the strength of non-stationarity and the likelihood that an SSA component is artefactual. In order to assess this systematically, we manually classify the SSA components of all subjects into artefact/non-artefact.

Figure 5.12 shows the results. The histogram shows the distribution of the non-stationarity over all $40 \cdot 40 = 1600$ SSA directions, along with an empirical estimate of the conditional artefact likelihood (red curve). The shape of this curve indicates that the non-stationarity is a strong predictor for artefacts; in fact, above a certain threshold every component found in this dataset is artefactual. These finding are confirmed by the AUC scores shown in the right panel of Figure 5.12. Not only is the degree of non-stationarity a good predictor across all directions of all subjects, but it also performs well for most subjects individually: the median AUC over subjects is above 0.9 and the 25% percentile well above 0.85.

So far, we have evaluated and compared the SSA solution from the perspective of the non-stationary spectrum. This is a univariate measure, which ignores the changes in the correlations between the non-stationary variables. Correspondingly, the non-stationary directions have been optimised univariately in deflation mode.

Whether a univariate approach to finding the non-stationary *subspace* is sufficient depends on the underlying data generating mechanisms. In order to verify whether this is the case, we compare the non-stationarity of subspaces found by univariate (deflation mode) and multivariate SSA. More specifically, for univariate SSA we choose as a basis for the d-dimensional non-stationary subspace the top d most

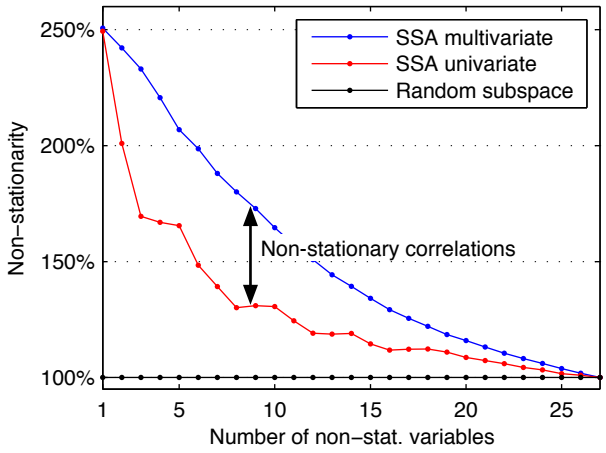

Figure 5.13: Comparison of univariate (deflation mode) SSA vs. multivariate optimisation of a non-stationary subspace by SSA over subspaces of dimension one to 26 (horizontal axis). The vertical axis shows the non-stationarity relative to the median of a random subspace.

non-stationary directions. If there is no advantage to a multivariate optimisation, then the degree of non-stationarity of the two subspaces should be similar.

Figure 5.13 shows the results for one subject. Multivariate SSA clearly outperforms deflation mode SSA: for a ten-dimensional subspace, the subspace found by multivariate SSA is nearly twice as non-stationary. This indicates that there are strong non-stationarities in the correlation between latent variables, which calls for a multivariate approach to the analysis of distribution changes in EEG data. Even though univariate solutions have the appealing property of being unique (no arbitrary choice of basis), in this example we see that they do not reveal the full picture. A canonical way to arrive at a unique basis for a non-stationary subspace would be a post-processing in the obtained source-space using deflationary SSA.

The strongest task-locked non-stationary direction

In the last part of this section, we show in an example that the most non-stationary *task-locked* source found by SSA has a meaningful interpretation in terms of the expected neural response of the task. Here, we apply SSA not to the concatenation of all trials of the calibration phase, but use a sliding window *within* the trials of all classes to define our epochs, i.e. our analysis is locked to the motor imager task. The sliding window has length 0.5s and an overlap of 50%. At each position within the epoch, we average the covariance matrix over all trials of the same class (left or right).

In this task-locked setup, one would expect the strongest non-stationary source to correspond to the attenuation of the SMR. Figure 5.14 shows that this is indeed

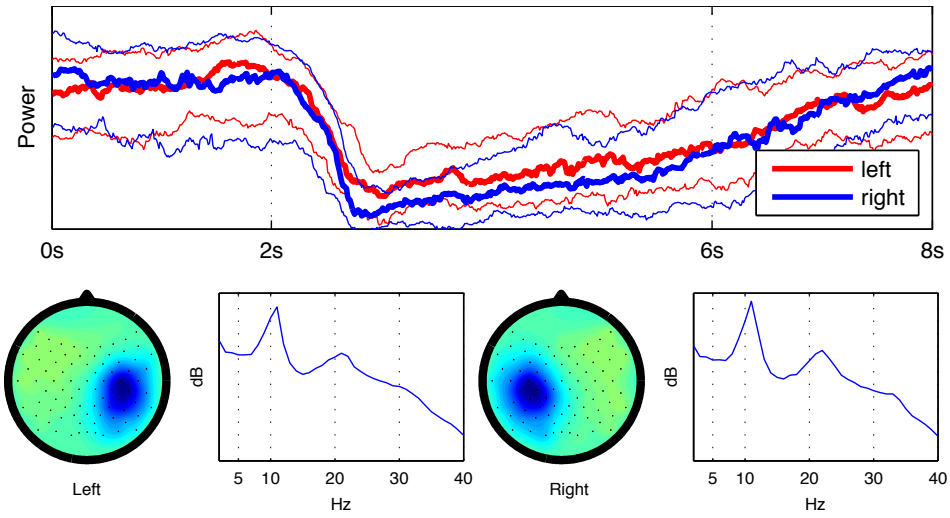

Figure 5.14: Illustration of the most non-stationary source within the trials of class left and right respectively. The top panel shows the median power time course for both sources on the corresponding trials; the bottom panels show the scalp pattern and frequency spectra.

the case, for both classes. The power time series shows the clear response to the stimuli (prompt for imagination) 2s after the beginning of the trial.

While this is an example specifically constructed to allow for comparison against a ground truth, there exist completely unsupervised analysis setups where no labels or other auxiliary information to guide the extraction of useful components are available (e.g. in the analysis of spontaneous activity [Bießmann et al. (2011), Biswal et al. (1995)] or resting state networks). It is in these cases that the most non-stationary components may be precisely the ones that are most interesting, which, as we have seen, PCA or ICA can not, in general, find.

5.3.3 Non-stationarities in brain-computer-interfacing

A major impediment to the more widespread use of EEG-based brain-computer-interfaces is their deteriorating performance over time, for which several adaptation techniques have been proposed [Vidaurre et al. (2011), Shenoy et al. (2006)]. During the calibration phase, the subject is prompted to produce motor imagery which is then used to train a classifier. However, as time passes, these parameters can become suboptimal. This may be due to changes in the mental state (e.g. increased tiredness of the subject) or in response to changing conditions during the application phase, where the environment is less controlled than during the calibration phase.

In this section, we present some first steps towards investigating the role of non-stationary brain sources using SSA. Our hypothesis is that (a) the generalisation

error can suffer from learning along directions where the distribution of the data changes and (b) that we can identify these directions using the training set. That is, we assume that the relevant non-stationary directions on the test set already show non-stationary behaviour during the training condition. We test this hypothesis by adding a pre-processing step to the standard offline analysis in which we remove non-stationary directions. The performance is evaluated on the test set.

Setup and results

The aim of this SSA pre-processing is to remove non-stationary directions, while not removing too much class-discriminative information. Clearly, this is a trade-off. In order to keep the analysis simple, we merely choose an epoch structure which reflects this: each SSA epoch consists of two trials, one from each class, chosen chronologically. Moreover, in order to detect those non-stationarities most likely to affect the CSP+LDA solution, we bandpass filter the data to the most discriminative frequency band, selected by the standard heuristics [Blankertz et al. (2008)]. The most non-stationary directions are found using SSA in deflation mode in order to allow for a step-wise procedure and an interpretation of individual components.

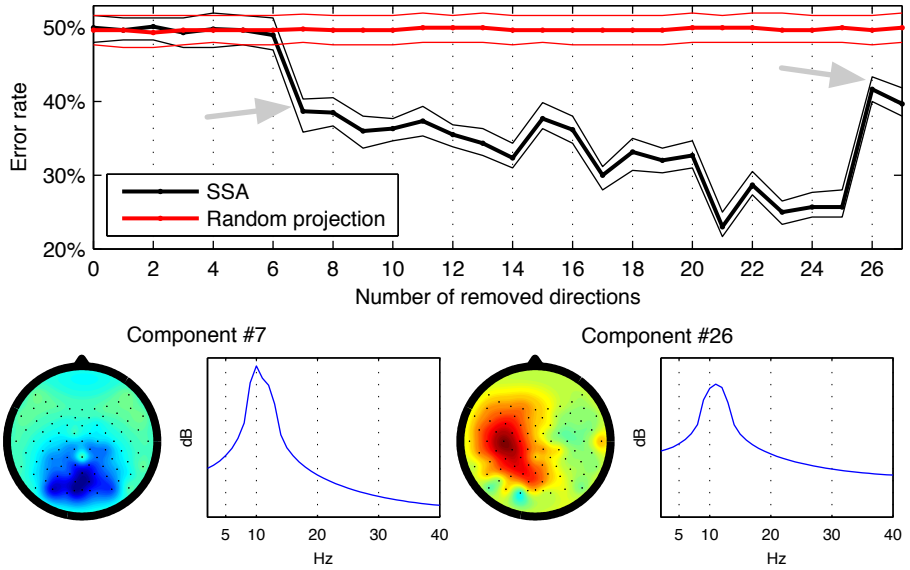

Figure 5.15: Results of the SSA pre-processing for one subject (S7) over varying numbers of removed non-stationary directions (horizontal axis); the vertical axis shows the error rate on the test set. The error tube extends from the 25% to the 75% over resamples of the test data; the two scalp plots correspond to the directions that were removed in the step indicated by the black circle.

From the 40 subjects of the VitalBCI study, we select the bottom 30% subjects in terms of classification accuracy on the test set. It is for this group of subjects

that we conjecture that distribution changes have a role to play: despite strongly discriminative CSP components found on the training set, the performance on the test set is sub-standard.

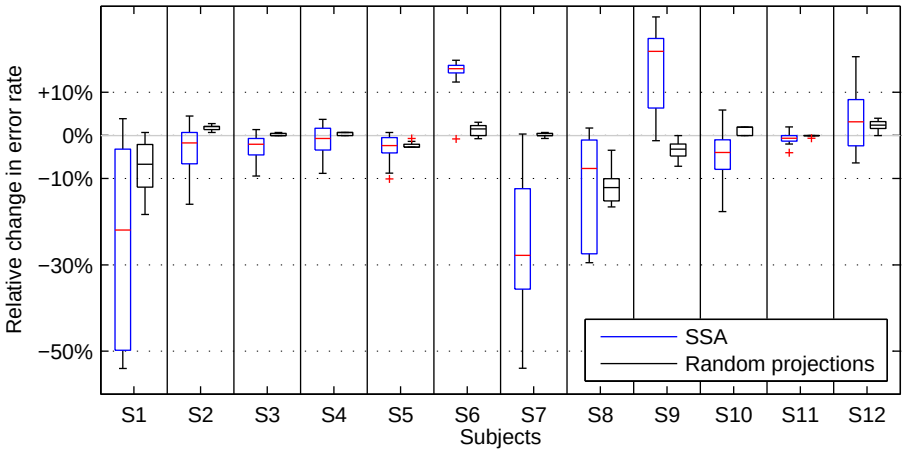

Figure 5.16: Results of the SSA pre-processing over all 12 selected subjects. The vertical axis shows the change in test error rate relative to the baseline; for each subject (horizontal axis) we report its distribution (blue boxplot) over removing the top $k = 1, \ldots, 25$ most non-stationary directions compared to the effect of removing the same number of random directions (black boxplot).

The crucial parameter of the SSA pre-processing (apart from the epoch structure) is how many resp. which non-stationary directions should be removed. Clearly, at some point we will inevitably remove important discriminative information.

The results for one subject (S7) are shown in Figure 5.15. We see that after removing the 7-th most non-stationary direction, the error rate drops from 50% (chance level) to below 40%. The scalp plot of the removed component suggests that we have removed the α-rhythm, whose strength is associated with the level of fatigue or tiredness. This source overlaps with the discriminative SMR, both spatially (in electrode space) and in terms of frequency content. It is therefore likely that the power of the α-rhythm contributes to the feature vectors. If the power of the α-rhythm changes between training and test (e.g. due to increase tiredness), then this may lead to suboptimal performance on the test set.

As we continue to remove non-stationary directions, the error rate eventually drops below 25%. However, with the removal of the 26-th most non-stationary source, the error rate increases strongly. As we can see from the scalp plot, the pattern resembles the characteristic pattern of the μ-rhythm associated with the right motor cortex. By removing it, we remove discriminative information which is reflected in the increased error rate.

The results over all subjects are shown in Figure 5.16 and Figure 5.17. For each subject, we remove the top k most non-stationary components and report the dis-

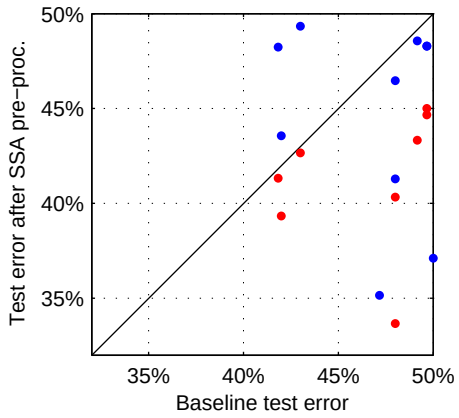

Figure 5.17: Evaluation of the SSA pre-processing for BCI. Each dot corresponds to a subject; the black dots show the *median* test error rate over the top $1, \ldots, 25$ most non-stationary directions. The red dots show the test error rate for an *optimal* choice of this parameter.

tribution of results over all k from 1 to 25. Apart from the three subjects S6, S9 and S12, the median effect of the SSA pre-processing is a reduction in error rate. For the subjects S6 and S9, the most non-stationary direction found in the first step is highly correlated with the label. An expert manually evaluating each direction on the training set would have detected this (characteristic scalp plot and frequency spectrum); however, in our automatic procedure, this leads to a drop in performance.

Whether this pre-processing can be turned into a practical approaching for BCI depends on whether the non-stationary directions that are to be removed can be selected automatically by some criterion. Potential approaches include cross-validation on the training set; a heuristic combining the degree of non-stationarity with a measure of discriminatory power; or an automatic data-driven classification of harmful vs. benign distribution changes based on features of the components.

However, what this analysis has shown is that there *exist* non-stationary components which are highly detrimental to BCI performance for some subjects (subject 7 improved from 50% (chance level) down to almost 20% error rate) and that SSA offers a means of finding them.

5.3.4 Recovering auditory steady-state evoked potentials

In this section, we use SSA to recover a brain source that has been induced by an experimental stimulus, namely the *auditory steady-state evoked potential* (ASSEP). This evoked potential is generated in the brain in response to an auditory stimulus which occurs at a constant frequency. It has been shown that such a steady-state stimulus evokes a response in the auditory system at the stimulation frequency

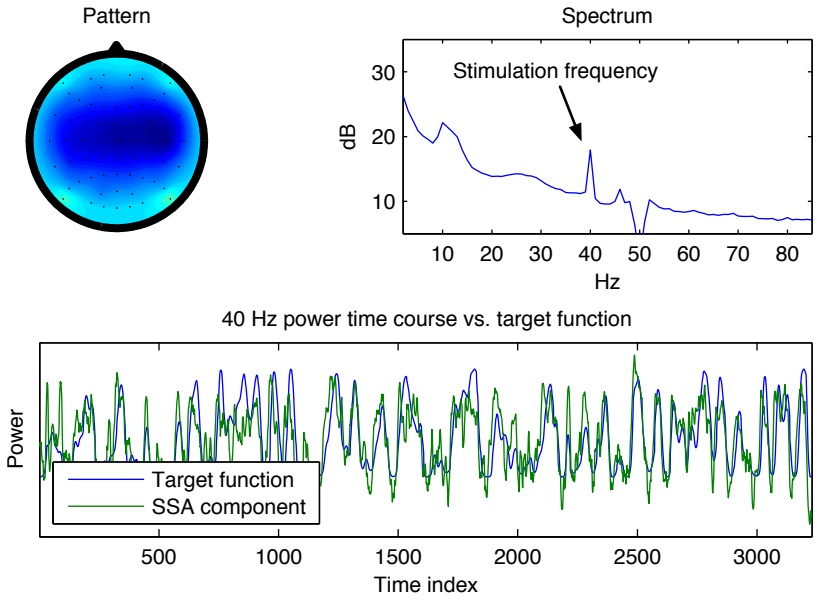

Figure 5.18: Component found by SSA that has the highest correlation with the target (stimulus). The top left panel shows the scalp plot (basis vector) corresponding to the source; the top right panel shows its frequency content. The bottom panel shows the power time course of the SSA component (green) and the target function (blue).

[Plourde et al. (1991), Galambos et al. (1981), Picton et al. (2003)]. Moreover, the amplitude of the ASSEP has been found to correlate to the intensity of the steady-state stimulus [Rodriguez et al. (1986)]. Thus, by modulating the intensity of the stimulus, we can induce a non-stationary brain source.

We analyse a dataset presented by [Dähne et al. (2012)] and follow a similar evaluation strategy. In this experiment, a 500 Hz pure tone is amplitude modulated by a 40 Hz cosine. In order to continuously vary the intensity of the stimulus, it is multiplied by a slowly varying function which modulates the loudness between 10 and 35 dB relative to the subject-specific hearing level. The EEG signals were recorded using 63 wet Ag/AgCl electrodes (Fast'n Easy Cap, EasyCap GmbH) placed according to the 10-20 system. Signals were amplified using two 32-channel amplifiers (Brain Products), sampled at 1 kHz and filtered by an analog bandpass filter between 0.1 and 250 Hz. For the offline analysis, the signals were down-sampled to 250 Hz and a notch filter around 50 Hz was applied to attenuate line noise.

We analyse a block of three minutes of continuous stimulation. SSA is applied to find the most non-stationary sources in deflation mode; epochs are defined by a sliding window of 0.5s length and step 0.25s. In a pre-processing step, the data is bandpass filtered to a window around the expected 40 Hz peak; the width of this window is varied during the analysis. Each obtained component is evaluated in terms of

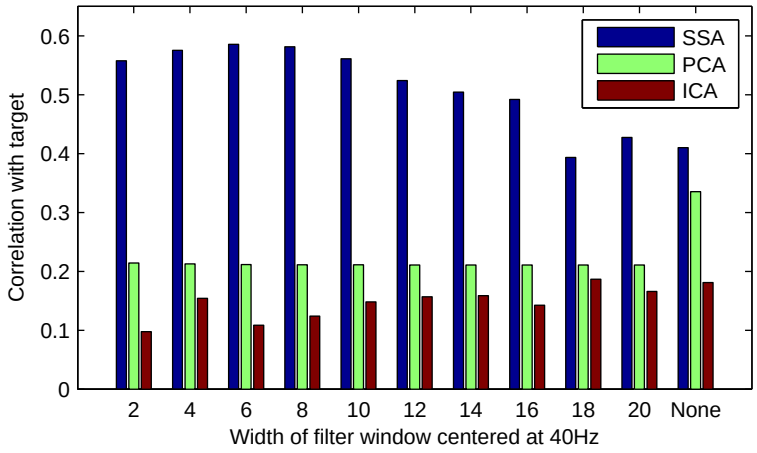

Figure 5.19: Comparison of PCA, FastICA and SSA in recovering the auditory steady-state component. The horizontal axis shows the maximum correlation with the target signal (the stimulus) over all components of each method for varying breadths of bandpass filters around 40 Hz.

its correlation to the intensity modulation time series. We compare the components found by SSA with the PCA and FastICA [Hyvärinen (1999)] solutions.

Figure 5.18 shows the SSA direction with the highest correlation to the target found without bandpass filtering (corresponding to the right most bars in Figure 5.19). The scalp plot shows the characteristic broad central pattern. The power spectrum (top right panel) exhibits a clear peak at 40 Hz, the stimulation frequency; and in the bottom panel, we see that the intensity modulation is correlated with the power of the SSA component in a small window around 40 Hz.

Figure 5.19 shows the results of the comparison of SSA against PCA and FastICA. The bars show the maximum correlation coefficient over all components for each method. Overall, SSA significantly outperforms PCA and FastICA, probably because it is the only method that maximises a criterion which is related to the characteristics of this source.

5.4 Summary and Discussion

The simulations on synthetic data revealed the fundamental relationship between the performance of SSA and basic parameters of the data. One result stands out: estimation errors in individual epochs average out *across* epochs. This is good news for the practitioner, who is often keen to increase the temporal resolution and capture as much variation as possible, which means making the epochs smaller.

Another important outcome is the invalidation of the likelihood ratio test for real, i.e. non-gaussian, data. On the other hand, a test based on resampling per-

formed remarkably well. Even so, more robust analytic test procedures would be highly desirable to alleviate the computational burden of resampling. A straightforward approach might be to insert a more heavy-tailed distribution into the likelihood ratio formalism or to expand the objective function around higher-order moments.

A comparison of the standard and the algebraic SSA algorithm confirmed that the latter is advantageous in a niche setting: large number of epochs with a high number of samples in low dimension. Even though this is probably exactly the opposite of what one expects in practice, the results do highlight the attractive features of the algebraic approach. Several possible approaches towards reducing its computational cost were discussed in Chapter 4.

The results on EEG analysis have shown clearly that PCA and ICA are not useful for understanding non-stationarities in neural activity. The non-stationary directions that ICA finds are artefacts, not brain sources. Secondly, a multivariate approach is necessary to find the most non-stationary subspace which captures the non-stationarities in the dependency structure. In this respect, SSA differs from most linear latent variable methods in that it finds a subspace and not a set of individual components. While we do not offer an interpretation of the non-stationary *subspace*, this clearly is an exciting direction for future research.

In two settings we have demonstrated that SSA finds solutions corresponding to a ground truth: the most non-stationary sources in task-locked setting from BCI clearly corresponds to the intentionally modulated μ-rhythm; and SSA successfully finds the auditory steady state evoked potential, which PCA and ICA do not. These findings highlight the ability of SSA to extract meaningful components in completely unsupervised settings where the relevant components are those that exhibit distribution changes.

The application to BCI analysis showed that in some subjects, non-stationary brain sources are highly detrimental to performance. Removing them in a pre-processing step prior to calibration yielded reductions in error rates of almost 30 percentage points on some subjects. While it is unclear whether SSA pre-processing can be turned into a practical automatic step that is beneficial to a sufficiently large number of subjects, the results show that (a) distribution changes can have a strong influence and (b) that SSA is the right tool for analysing them.

Chapter 6

Conclusion

Analysing change in observed data in response to the change in a controlled variable is a quintessential task in empirical research. As such, there exists a wide range of statistical methods which address this problem. The common approach in the multivariate case is to find a direction in the data space associated with the change in the controlled variable. For instance, linear discriminant analysis [Fisher (1936)] finds a direction along which the distribution of the data is maximally different between two sets of samples. In EEG analysis, common spatial patterns [Koles (1991)] find linear projections that allow us to distinguish between two experimental conditions. However, the applicability of these families of methods crucially depend on the availability of an externally controlled variable, or label. This is not always available. In this respect, SSA closes a gap because it allows for the *unsupervised* analysis of change in data.

Summary of the main results

A simple example in Chapter 1 shows that in order to analyse changes in the joint distribution of a multivariate data set, a univariate perspective is, in principle, insufficient. In essence, this is because changes in the dependencies between variables are not necessarily visible in the marginals. The two-variable example also showed the existence of another coordinate system, better suited to our task: it separates the stationary from the non-stationary components. This observation sets the theme for this thesis: finding such advantageous coordinate systems.

In the next Chapter 2 we equip ourselves with the necessary background for this task. We review fundamentals from mathematics, the basic framework of coordinate transformations in machine learning and statistics, and the two most prominent methods: principal component analysis (PCA) and independent component analysis (ICA).

Chapter 3 contains our main contribution, the formulation of the stationary subspace analysis problem and the first algorithm to solve it. We start by introducing a generative model, discuss its identifiability, and define our notion of stationarity: time-constant mean and covariance matrix. Corresponding to this definition,

we derive a measure of stationarity based on epochs of the given time series, which leads to an efficient algorithm that minimises or maximises this objective to find the stationary and non-stationary components respectively. We discuss the problem of spurious solutions and present a theoretical result which tells us precisely how much data is needed to avoid them in practice.

In Chapter 4, we develop an alternative formulation of the SSA problem in terms of an approximate solution to a system of polynomial equations. In contrast to the SSA algorithm developed in the previous chapter, we do not *search* for the solution guided by the gradient of an objective function but directly *solve* the problem algebraically. This leads to more accurate solutions in certain scenarios; and allows us to choose the number of stationary directions from data by computing a certain eigenvalue spectrum.

Chapter 5 is concerned with the validation of the SSA algorithm on synthetic data and in applications to EEG data analysis. In controlled simulations, we analyse the influence of key parameters on the performance of the SSA algorithm, compare two approaches for stationarity testing, and demonstrate the relative merits of the algebraic SSA algorithm.

In three applications to EEG analysis, we show that SSA provides us with genuinely new and useful results. First of all, we demonstrate that neither PCA nor ICA allow us to uncover the non-stationarity directions in EEG data. In particular, we also show that non-stationarities are, loosely speaking, multivariate in EEG data: the most non-stationary *subspace* is not the combination of the individually maximally non-stationary directions, but can only be found by joint optimisation of several projections. Moreover, we show that the most non-stationary direction in a brain-computer-interfacing context has a meaningful interpretation, in that it corresponds to the intentional attenuation of the SMR.

Secondly, we investigate the role of non-stationarities in brain-computer-interfacing. It has often been hypothesised that distribution changes are one of the factors behind deteriorating performance over time as parameters calibrated during the training sessions may be suboptimal in the long term, should the distribution of the EEG data change due to environmental influences or changes in the brain state. To this end, we investigate whether removing non-stationary directions (estimated during calibration) can be beneficial for performance. Our results show that for some subjects, this indeed leads to a dramatic increase in performance. While it is unclear whether this pre-processing can be turned into an automated practical method (due to difficulties in parameter choice), this analysis demonstrates that SSA is a useful new tool in this context.

In the third application, we show that SSA can recover a neural component from EEG signals which PCA and ICA do not find. The auditory steady state evoked potential is the response to an auditory stimulus which occurs at a steady frequency (beeps). This power of this response has been found to correlate with the volume (intensity) of the stimulus. By varying the stimulus intensity, we can induce a brain

response (at the stimulus frequency) that has non-stationary variance. Therefore, SSA should be able to find it which is indeed the case. We compare the performance of SSA, PCA and ICA in terms of the correlation with the stimulus intensity: as it turns out, neither PCA nor ICA can find components that are significantly correlated.

Limitations and future directions

The most relevant limitations of the presented SSA algorithm are related to its underlying definition of stationarity: changes in the higher order moments go unnoticed and the time structure within the epochs is discarded. While an extension in the former direction probably comes at a high price (in terms of sample and computational complexity), integrating time structure should be possible along the lines of TDSEP. As many real sources are endowed with temporal structure, this should bring about significant improvements.

Another important question is: how to choose the number of stationary resp. non-stationary sources? We have proposed two answers: post-hoc stationarity testing using resampling, or inspecting the PCA eigenvalue spectrum in coefficient vector space It would be interesting to investigate an approach based on a model selection framework [Akaike (1974), Schwarz (1978), Rissanen (1978)] or a Bayesian formulation that allows us to compute the posterior over possible numbers of stationary directions.

The applications to EEG analysis presented in this chapter is a very first step. Future research will investigate the types and potential causes of non-stationary neural components, both in task-locked and paradigmless settings, possibly in combination with source reconstruction techniques [Scherg and Von Cramon (1986), Herrmann et al. (2004), Haufe et al. (2008)]. In the EEG context, it may be possible to interpret the non-stationary *subspace* (as opposed to individual components) using physiological constraints that reduce the number of plausible bases [Schmidt (1986), Mosher and Leahy (1999)]. Moreover, our results showed that the degree of non-stationarity is related to artefact-likelihood. Whether this can be turned into a practical artefact rejection method needs to be evaluated.

Where else can the algebraic approach be advantageous? In short, whenever the desired solution is a set of polynomials of a certain type (in the SSA case: linear projections) whose approximate vanishing set is given in terms of another set of input polynomials, usually of higher degree. That is, both input and output are polynomials and the objective of the learning task needs to be describable in terms of equations. This is, for example, the case in the simultaneous diagonalisation of several matrices [Cardoso and Souloumiac (1996), Bunse-Gerstner et al. (1993)], where one aims to make the off-diagonal elements equal to zero by applying a linear basis transformation.

Bibliography

[Abdi (2003)] H. Abdi. Factor rotations in factor analyses. *Encyclopedia for Research Methods for the Social Sciences. Sage: Thousand Oaks, CA*, pages 792–795, 2003.

[Akaike (1974)] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[Allen et al. (2011)] E.A Allen, E.B Erhardt, E Damaraju, W Gruner, J.M Segall, R.F Silva, M Havlicek, S Rachakonda, J Fries, and R Kalyanam. A baseline for the multivariate comparison of resting-state networks. *Front Syst Neurosci*, 5, 2011.

[Bell and Sejnowski (1995)] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, November 1995.

[Belouchrani et al. (1997)] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, and E. Moulines. A blind source separation technique using second order statistics. *IEEE Trans. on Signal Processing*, 45(2):434–444, 1997.

[Bengio et al. (2004)] Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16: 177–184, 2004.

[Berger (1929)] Hans Berger. Über das elektroencephalogramm des menschen. *Archiv fur Psychiatrie und Nervenkrankheiten*, 87:527–570, 1929.

[Bießmann et al. (2011)] Felix Bießmann, Sergey M Plis, Frank C Meinecke, Tom Eichele, and Klaus-Robert Müller. Analysis of multimodal neuroimaging data. *Biomedical Engineering, IEEE Reviews in*, 4:26 – 58, 2011.

[Biswal et al. (1995)] B Biswal, F Z Yetkin, V M Haughton, and J S Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magn Reson Med*, 34(4):537–41, Oct 1995.

[Blanchard et al. (2006)] G. Blanchard, M. Kawanabe, M. Sugiyama, V. Spokoiny, and K.R. Müller. In search of non-gaussian components of a high-dimensional distribution. *The Journal of Machine Learning Research*, 7: 247–282, 2006.

[Blankertz et al. (2008)] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.R. Muller. Optimizing spatial filters for robust eeg single-trial analysis. *Signal Processing Magazine, IEEE*, 25(1):41–56, 2008.

[Blankertz et al. (2010)] Benjamin Blankertz, Claudia Sannelli, Sebastian Halder, Eva M. Hammer, Andrea Kübler, Klaus-Robert Müller, Gabriel Curio, and Thorsten Dickhaus. Neurophysiological predictor of SMR-based BCI performance. *NeuroImage*, 51(4):1303–1309, 2010.

[Blythe et al. (2012)] Duncan A. J. Blythe, Paul von Bünau, Frank C. Meinecke, and Klaus-Robert Müller. Feature extraction for change-point detection using stationary subspace analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):631–643, 2012.

[Bunse-Gerstner et al. (1993)] A. Bunse-Gerstner, R. Byers, and V. Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 14:927–927, 1993.

[Caboara et al. (1997)] Massimo Caboara, Pasqualina Conti, and Carlo Traverse. Yet another ideal decomposition algorithm. In Teo Mora and Harold Mattson, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1255 of *Lecture Notes in Computer Science*, pages 39–54. Springer Berlin / Heidelberg, 1997.

[Cardoso and Souloumiac (1993)] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non gaussian signals. *IEE Proceedings-F*, 140:362–370, 1993.

[Cardoso (1999)] J.F. Cardoso. High-order contrasts for independent component analysis. *Neural computation*, 11(1):157–192, 1999.

[Cardoso and Souloumiac (1996)] J.F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.

[Caton (1875)] Richard Caton. The electric currents of the brain. *British Medical Journal*, 2(765):278, 1875.

[Corless et al. (1995)] R.M. Corless, Patrizia M. Gianni, Barry M. Trager, and S.M. Watt. The singular value decomposition for polynomial systems. *Proc. ISSAC '95*, pages 195–207, 1995.

[Cover et al. (1991)] T.M. Cover, J.A. Thomas, J. Wiley, et al. *Elements of information theory*, volume 6. Wiley Online Library, 1991.

[Cox et al. (2007)] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 0387356509.

[Croft and Barry (2000)] R.J. Croft and RJ Barry. Removal of ocular artifact from the eeg: a review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 30(1): 5–19, 2000.

[Dähne et al. (2012)] Sven Dähne, Vadim Nikulin, Frank Meinecke, Stefan Haufe, Johannes Höhne, Michael Tangermann, and Klaus-Robert Müller. Optimal spatial filters for correlating band power with cognitive function, 2012.

[Dornhege (2007)] G. Dornhege. *Toward brain-computer interfacing*. The MIT Press, 2007.

[Eisenbud et al. (1992)] David Eisenbud, Craig Huneke, and Wolmer Vasconcelos. Direct methods for primary decomposition. *Inventiones Mathematicae*, 110: 207–235, 1992. ISSN 0020-9910.

[Fatourechi et al. (2007)] M. Fatourechi, A. Bashashati, R.K. Ward, and G.E. Birch. Emg and eog artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, 118(3):480–494, 2007.

[Fisher (1936)] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

[Friedman and Tukey (1974)] J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *Computers, IEEE Transactions on*, 100(9):881–890, 1974.

[Froeberg (1985)] Ralf Froeberg. An inequality for hilbert series of graded algebras. *Math. Scand.*, 56:117 – 144, 1985.

[Froeberg and Hollman (1994)] Ralf Froeberg and Joachim Hollman. Hilbert series for ideals generated by generic forms. *Journal of Symbolic Computation*, 17(2):149 – 157, 1994. ISSN 0747-7171.

[Galambos et al. (1981)] R. Galambos, S. Makeig, and P.J. Talmachoff. A 40-hz auditory potential recorded from the human scalp. *Proceedings of the national academy of sciences*, 78(4):2643, 1981.

[Gauß (1809)] Carl Friedrich Gauß. Theoria motus corporum coelestium in sectionibus conicis solem ambientium. *Göttingen*, 1809.

[Gianni et al. (1988)] Patrizia Gianni, Barry Trager, and Gail Zacharias. Groebner bases and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6(2-3):149 – 167, 1988. ISSN 0747-7171.

[Golub and Van Loan (1996)] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins Univ Pr, 1996.

[Gretton et al. (2012)] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13:723–773, 2012.

[Guyon and Elisseeff (2003)] Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3: 1157–1182, Mar 2003.

[Hara et al. (2010)] Satoshi Hara, Yoshinobu Kawahara, Takashi Washio, and Paul von Bünau. Stationary subspace analysis as a generalized eigenvalue problem. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I*, ICONIP'10, pages 422–429, Berlin, Heidelberg, 2010. Springer-Verlag.

[Hara et al. (2012)] Satoshi Hara, Yoshinobu Kawahara, Takashi Washio, Paul von Bünau, Terumasa Tokunaga, and Kiyohumi Yumoto. Separation of stationary and non-stationary sources with a generalized eigenvalue problem. *Neural Networks*, 33:7–20, 2012. ISSN 0893-6080.

[Harman (1976)] H.H. Harman. *Modern factor analysis*. University of Chicago Press, 1976.

[Hastie and Stuetzle (1989)] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, pages 502–516, 1989.

[Haufe et al. (2008)] S. Haufe, V.V. Nikulin, A. Ziehe, K.R. Müller, and G. Nolte. Combining sparsity and rotational invariance in eeg/meg source reconstruction. *NeuroImage*, 42(2):726–738, 2008.

[Heldt et al. (2009)] Daniel Heldt, Martin Kreuzer, Sebastian Pokutta, and Hennie Poulisse. Approximate computation of zero-dimensional polynomial ideals. *Journal of Symbolic Computation*, 44(11):1566 – 1591, 2009. ISSN 0747-7171. In Memoriam Karin Gatermann.

[Hermann (1926)] Grete Hermann. Die Frage der endlich vielen Schritte in der Theorie der Polynomideale - unter Benutzung nachgelassener S□tze von K. Hentzelt. *Mathematische Annalen*, 95(1):736 – 788, 1926. ISSN 0747-7171.

[Herrmann et al. (2004)] M.J. Herrmann, J. Römmler, A.C. Ehlis, A. Heidrich, and A.J. Fallgatter. Source localization (loreta) of the error-related-negativity (ern/ne) and positivity (pe). *Cognitive Brain Research*, 20(2):294–299, 2004.

[Hinton and Roweis (2002)] G. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15:833–840, 2002.

[Huber (1985)] P.J. Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985.

[Hyvarinen (1999)] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, 10 (3):626–634, 1999.

[Hyvärinen et al. (2001)] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 26. Wiley-interscience, 2001.

[Hyvärinen (1999)] Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10 (3):626–634, May 1999. ISSN 10459227.

[Jaynes (1957)] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 160(620–630), 1957.

[Jaynes and Bretthorst (2003)] E.T. Jaynes and G.L. Bretthorst. *Probability theory: the logic of science*. Cambridge Univ Pr, 2003.

[Jung et al. (2000)] T.P. Jung, S. Makeig, C. Humphries, T.W. Lee, M.J. Mckeown, V. Iragui, and T.J. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(02):163–178, 2000.

[Jung et al. (2001)] T.P. Jung, S. Makeig, M.J. McKeown, A.J. Bell, T.W. Lee, and T.J. Sejnowski. Imaging brain dynamics using independent component analysis. *Proceedings of the IEEE*, 89(7):1107–1122, 2001.

[Kaiser (1958)] H.F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

[Kaltofen et al. (2007)] E. Kaltofen, Z. Yang, and L. Zhi. Structured low rank approximation of a sylvester matrix. *Symbolic-numeric computation*, pages 69–83, 2007.

[Kandel et al. (2000)] E.R. Kandel, J.H. Schwartz, T.M. Jessell, et al. *Principles of neural science*, volume 4. McGraw-Hill New York, 2000.

[Kawanabe et al. (2011)] Motoaki Kawanabe, Wojciech Samek, Paul von B¸nau, and Frank Meinecke. An information geometrical view of stationary subspace analysis. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 397–404. Springer Berlin / Heidelberg, 2011.

[Kiraly et al. (12)] Franz J. Kiraly, Paul von Bünau, Frank C. Meinecke, Duncan A. J. Blythe, and Klaus-Robert Müller. Algebraic geometric comparison of probability distributions. *Journal of Machine Learning Research*, 13:855–903, 12.

[Kiraly et al. (2012)] Franz J. Kiraly, Paul von Bünau, Jan Saputra Müller, Duncan A. J. Blythe, Frank C. Meinecke, and Klaus-Robert Müller. Regression for sets of polynomial equations. In *JMLR Workshop and Conference Proc. Vol. 22*, pages 628–637, 2012.

[Kohonen (1990)] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[Koles (1991)] Z.J. Koles. The quantitative extraction and topographic mapping of the abnormal components in the clinical eeg. *Electroencephalography and Clinical Neurophysiology*, 79(6):440–447, 1991.

[Kreuzer et al. (2009)] Martin Kreuzer, Hennie Poulisse, and Lorenzo Robbiano. *Approximate Commutative Algebra*, chapter From Oil Fields to Hilbert Schemes. Springer-Verlag Berlin Heidelberg, 2009.

[Krick and Logar (1991)] Teresa Krick and Alessandro Logar. An algorithm for the computation of the radical of an ideal in the ring of polynomials. In Harold Mattson, Teo Mora, and T. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 539 of *Lecture Notes in Computer Science*, pages 195–205. Springer Berlin / Heidelberg, 1991.

[Laplagne (2006)] Santiago Laplagne. An algorithm for the computation of the radical of an ideal. In *Proceedings of the 2006 international symposium on Symbolic and algebraic computation*. ACM, 2006. URL `http://dx.doi.org/10.1145/1145768.1145802`.

[Legendre (1805)] Adrien-Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*, chapter Sur la methode des moindres quarres. Firmin Didot, http://imgbase-scd-ulp.u-strasbg.fr/displayimage.php?pos=-141297, 1805.

[Makeig et al. (1996)] S. Makeig, A.J. Bell, T.P. Jung, T.J. Sejnowski, et al. Independent component analysis of electroencephalographic data. *Advances in neural information processing systems*, pages 145–151, 1996.

[McMenamin et al. (2010)] B.W. McMenamin, A.J. Shackman, J.S. Maxwell, D.R.W. Bachhuber, A.M. Koppenhaver, L.L. Greischar, and R.J. Davidson. Validation of ica-based myogenic artifact correction for scalp and source-localized eeg. *Neuroimage*, 49(3):2416–2432, 2010.

[Meinecke et al. (2009)] Frank C. Meinecke, Paul von Bünau, Motoaki Kawanabe, and Klaus-Robert Müller. Learning invariances with stationary subspace analysis. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 87 –92, 2009.

[Meinecke (2011)] Frank Carsten Meinecke. *Synchronized? Identifying Interactions from Superimposed Signals*. PhD thesis, Berlin Institute of Technology (TU Berlin), 2011. URL `http://opus.kobv.de/tuberlin/volltexte/2012/3386/`.

[Molgedey and Schuster (1994)] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72(23):3634–3637, Jun 1994.

[Mood et al. (1974)] A.M. Mood, F.A. Graybill, and D.C. Boes. *Introduction to the theory of statistics*. McGraw-Hill Book Company, 1974.

[Mosher and Leahy (1999)] J.C. Mosher and R.M. Leahy. Source localization using recursively applied and projected (rap) music. *Signal Processing, IEEE Transactions on*, 47(2):332–340, 1999.

[Neyman and Pearson (1933)] J. Neyman and E.S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

[Park et al. (1999)] H. Park, L. Zhang, and J.B. Rosen. Low rank approximation of a hankel matrix by structured total least norm. *BIT Numerical Mathematics*, 39(4):757–779, 1999.

[Parra et al. (2005)] L.C. Parra, C.D. Spence, A.D. Gerson, and P. Sajda. Recipes for the linear analysis of eeg. *Neuroimage*, 28(2):326–341, 2005.

[Pearson (1901)] K Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[Pfurtscheller and Lopes da Silva (1999)] G. Pfurtscheller and FH Lopes da Silva. Event-related eeg/meg synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, 1999.

[Picton et al. (2003)] T.W. Picton, M.S. John, A. Dimitrijevic, and D. Purcell. Human auditory steady-state responses: respuestas auditivas de estado estable en humanos. *International journal of audiology*, 42(4):177–219, 2003.

[Plourde et al. (1991)] G. Plourde, D.R. Stapells, and T.W. Picton. The human auditory steady-state evoked potentials. *Acta Oto-Laryngologica*, 111(S491): 153–160, 1991.

[Plumbley (2005)] Mark D. Plumbley. Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. *Neurocomputing*, 67(161–197), 2005.

[Priestley (1983)] M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, 1983.

[Quionero-Candela et al. (2009)] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[Rahimi and Recht (2008)] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[Raichle et al. (2001)] M E Raichle, A M MacLeod, A Z Snyder, W J Powers, D A Gusnard, and G L Shulman. A default mode of brain function. *Proc Natl Acad Sci USA*, 98(2):676–82, Jan 2001.

[Rissanen (1978)] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[Rodriguez et al. (1986)] R. Rodriguez, T. Picton, D. Linden, G. Hamel, and G. Laframboise. Human auditory steady state responses: effects of intensity and frequency. *Ear and hearing*, 7(5):300, 1986.

[Roweis and Saul (2000)] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Samek et al. (2012)] Wojciech Samek, Klaus-Robert Müller, Motoaki Kawanabe, and Carmen Vidaurre. Brain-computer interfacing in discriminative and stationary subspaces. In *Conf Proc IEEE Eng Med Biol Soc*. IEEE EMBS, 2012.

[Scherg and Von Cramon (1986)] M. Scherg and D. Von Cramon. Evoked dipole source potentials of the human auditory cortex. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 65(5):344–360, 1986.

[Schmidt (1986)] R. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on*, 34(3):276–280, 1986.

[Schölkopf et al. (1998)] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[Schwarz (1978)] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[Schweikert et al. (2008)] Gabriele Schweikert, Christian Widmer, Bernhard Schölkopf, and Gunnar Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*, pages 1433–1440. Curran Associates, Inc., 2008.

[Shenoy et al. (2006)] P. Shenoy, M. Krauledat, B. Blankertz, R.P.N. Rao, and K.R. Müller. Towards adaptive classification for bci. *Journal of Neural Engineering*, 3:R13, 2006.

[Shimodaira (2000)] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[Smirnoff (1939)] N. Smirnoff. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bulletin de l'Universite de Moscow, Serie internationale (Mathematiques)*, 2:3–14, 1939.

[Stetter (2004)] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics, 2004. ISBN 0898715571.

[Sugiyama (2009)] M. Sugiyama. Density ratio estimation: A new versatile tool for machine learning. *Advances in Machine Learning*, pages 6–9, 2009.

[Sugiyama et al. (2007)] M. Sugiyama, M. Krauledat, and K.R. Müller. Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, 8:985–1005, 2007.

[Sugiyama et al. (2008)] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[Sugiyama et al. (2011)] Masashi Sugiyama, Makoto Yamada, Paul von Bünau, Taiji Suzuki, Takafumi Kanamori, and Motoaki Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24(2):183–198, 2011. ISSN 0893-6080.

[Tenenbaum et al. (2000)] J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290 (5500):2319–2323, 2000.

[Vidal (1973)] J.J. Vidal. Toward direct brain-computer communication. *Annual review of Biophysics and Bioengineering*, 2(1):157–180, 1973.

[Vidaurre et al. (2011)] C. Vidaurre, M. Kawanabe, P. Bunau, B. Blankertz, and K.R. Muller. Toward an unsupervised adaptation of lda for brain-computer interfaces. *Biomedical Engineering, IEEE Transactions on*, (99):1–1, 2011.

[von Bünau et al. (2010)] Paul von Bünau, Frank C. Meinecke, Simon Scholler, and Klaus-Robert Müller. Finding stationary brain sources in EEG data. In *Proceedings of the 32nd Annual Conference of the IEEE EMBS*, pages 2810–2813, 2010.

[Wald and Wolfowitz (1940)] A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *The Annals of Mathematical Statistics*, 11(2):147–162, 1940.

[Ziehe and Müller (1998a)] A. Ziehe and K.-R. Müller. TDSEP – an efficient algorithm for blind separation using time structure. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks, ICANN'98*, Perspectives in Neural Computing, pages 675 – 680, Berlin, 1998a. Springer Verlag.

[Ziehe and Müller (1998b)] A. Ziehe and K.-R. Müller. TDSEP – an efficient algorithm for blind separation using time structure. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proc. ICANN '98*, pages 675 – 680. Springer Verlag, 1998b.