Approximate Algebraic Estimation of High-Dimensional Stationary Projections in Stationary Subspace Analysis

Diplomarbeit im Fachgebiet Maschinelles Lernen
der Technische Universität Berlin
zur Erlangung des akademischen Grades Dipl. Inf.

Vorgelegt von Jan Saputra Müller

Betreuer:
Prof. Dr. Klaus-Robert Müller
Dr. Dr. Franz J. Király

Eigenständigkeitserklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 24.05.2012

Jan Saputra Müller

Deutsche Zusammenfassung

In der vorliegenden Diplomarbeit wir ein neuer Algorithmus vorgestellt, welcher das *Stationary Subspace Analysis* (SSA) Problem löst. SSA ist ein Modell zur blinden Quellentrennung, in welchem angenommen wird, dass eine beobachtete Zeitreihe $x(t)$ in Wahrheit eine lineare Superposition von statistisch stationären Quellen $s^{\mathtt{s}}(t)$ und nichtstationären Quellen $s^{\mathtt{n}}(t)$ ist. Die Aufgabe eines SSA-Algorithmus ist dabei, nur aufgrund der beobachteten Zeitreihe $x(t)$ eine Projektion auf den Raum der stationären Quellen zu finden.

Um das SSA-Problem zu lösen gibt es zwei etablierte Algorithmen — den klassischen Algorithmus [vBMKM09] [MvBM$^+$11], der SSA als ein Optimierungproblem formuliert, und dann mit Hilfe eines Gradienten-Abstiegsverfahren löst — und den algebraischen Algorithmus [KvBM$^+$12a] [KvBM$^+$12b], welcher SSA als ein algebraisches Problem formuliert, und mit Hilfe von Methoden aus der algebraischen Geometrie löst. Beide Algorithmen haben ihre Stärken und Schwächen. Die Lösungen des klassischen Algorithmus leiden an Genauigkeit, da die zu optimierende Kostenfunktion viele lokale Minima besitzt, was es schwer macht das globale Optimum zu finden. Der algebraische Algorithmus hingegen skaliert nur schlecht für hochdimensionale Zeitreihen.

In dieser Arbeit wird ein weiterer Algorithmus vorgestellt, nämlich Low-Rank SSA, welcher in einem Spezialfall des SSA-Problems angewendet werden kann, nämlich genau dann, wenn die Anzahl stationärer Quellen mehr als die Hälfte aller Quellen ausmacht. In diesem Fall erhält unser algebraisches Problem Zusatzstruktur, die ausgenutzt werden kann. Die Stärken des neuen Algorithmus liegen dabei vor allem darin, dass seine berechneten Lösungen von ihrer Genauigkeit vergleichbar mit dem algebraischen Algorithmus sind, seine Laufzeit aber erheblich besser für hochdimensionale Datensätze skaliert. Der Beitrag dieser Arbeit besteht darin, diese Zusatzstruktur mathematisch aufzuzeigen, und in einer Vielzahl von Experimenten die Leistung des neuen Algorithmus zu analysieren und zu erproben.

Acknowledgments

First, I want to thank all people in the Machine Learning Group at TU Berlin for a great teamwork in the last years. Especially, I want to thank Paul von Bünau, who introduced me in Stationary Subspace Analysis and encouraged me to work on this topic. Moreover, I want to thank him and particularly Franz Király for their valuable discussions and support when I was writing this thesis. Franz Király also had the excellent idea which lead to this thesis, since he discovered the Low-Rank SSA algorithm.

Furthermore, I want to thank Prof. Klaus-Robert Müller for his support and guidance through this thesis and Andrea Gerdes for her support on organizational matters.

Without all of you, this thesis would not have been possible.

Apart from the people who supported me in my work directly, I also want to thank my girlfriend Monika Skolimowska for her loving support and her reassuring words, even when I was stressed or at the end of my tether.

Moreover, I want to thank my best friends Richard Krieger and Jennifer Sturm for their mental support and relaxed evenings.

Last but not least, I want to thank my family for standing at my side and for supporting me in every part of my life.

Contents

Chapter 1

Introduction

1.1 Motivation

In many empirical sciences, understanding changes in distributions is an important problem. Often one deals with signals which have been recorded in experiments over a period of time, resulting in a time series. Those signals are often high-dimensional, as one wants to capture as much information as possible and uses many sensors.

This is for example the case in *brain computer interfacing* (BCI), where an *Electroencephalogram* (EEG) is recorded from a humans brain and one wants to detect change points — for instance to predict to changing sleep states. As we use many electrodes (e.g. up to 128 channels) and each electrode captures a mixture of different sources in the brain, it is a difficult task to extract the relevant information from this high-dimensional time series. The relevant information in this case would intuitively lie on some subspace of the data which is maximally non-stationary — i.e. we want to remove irrelevant stationary directions to improve our change point detection.

In fact, it is possible that there is only one non-stationary source in the whole time series, which would give us all relevant information, but which is overlayed by many stationary sources, such that it is hard to identify. In another situation, we may be interested only in the stationary sources, but one non-stationary source, which is mixed into each channel of our recordings, makes the whole time series appear non-stationary.

In *Stationary Subspace Analysis* (SSA), we assume that our observed signals are linear mixtures of stationary and non-stationary sources and we thus decompose the time series into those stationary and non-stationary components.

For an illustration of what SSA does, consider Figure 1.1, which was taken from [MvBM$^+$11]. In this example, we start with two signals which appear non-stationary. After applying SSA, we observe that there is a direction in which the distribution of our time series does not change over time — and another direction where the distribution changes. This direction explains the non-stationarity of both signals with which we started. By projecting our time series to the stationary and non-stationary directions, we get a stationary and non-stationary signal, respectively.

In the past few years, SSA has been used for many practical applications,

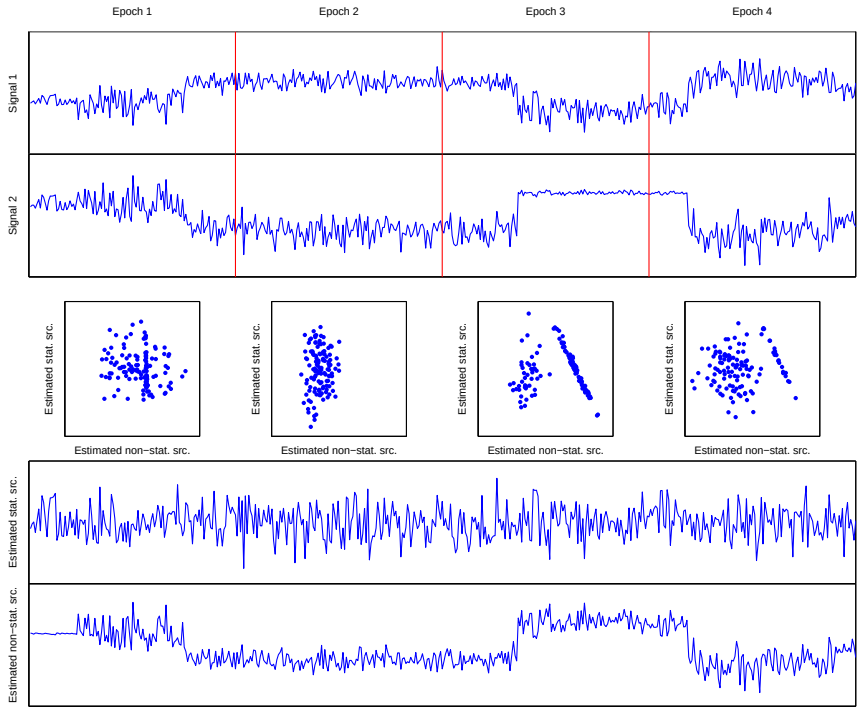

Figure 1.1: SSA example taken from [MvBM$^+$11]. Signal 1 and Signal 2 are non-stationary signals. By applying SSA, we see that both signals are a linear superposition of a stationary and non-stationary signal: the scatter plots in the middle row show that the projection to the vertical axis is identically distributed in each of the four epochs. On the other hand, the projection to the horizontal axis is non-stationary. In the last row we see the estimated stationary and non-stationary source.

for example for finding stationary brain sources in EEG data [vBMSM10], in geomagnetic data analysis [HKW$^+$12], for domain adaption [HKWvB10], for boosting high-dimensional change point detection [vBMM$^+$09] [BvBMM12] and for learning invariances [MvBKM09].

So far, Stationary Subspace Analysis has been formulated as an optimization problem [vBMKM09] which is non-convex and therefore hard to optimize. Recently, SSA has been reformulated as an algebraic problem in [KvBM$^+$12a] and [KvBM$^+$12b]. The key advantage of this formulation was that it does not suffer from convergence to local minima and therefore yields much more accurate results. But there has been also one disadvantage: The resulting algorithm does not scale well for high-dimensional data sets, so that it becomes impractical for most of the real-world problems.

In this thesis, we will introduce a new algorithm called *Low-Rank Stationary Subspace Analysis*, which was invented by Franz Király. We will show that it has the advantage of being accurate and that it scales well on high-dimensional data sets at the same time and is applicable to half of all possible SSA problems.

Additionally, we will introduce a new algorithm for estimating the number of stationary sources, which also was invented by Franz Király. We will give a theoretical derivation of both algorithm and investigate their performance in various simulations.

1.2 Main contributions

The main contributions in this thesis are the following:

- Theoretical backgrounds and derivation of the Low-Rank Stationary Subspace Analysis algorithm
- Investigation of the performance of Low-Rank SSA in various simulations
- Discussion of the results and future work

1.3 Overview

The structure of the thesis is as follows: first we will describe the Stationary Subspace Analysis problem formally and derive its algebraic formulation. Afterwards, we will show that if the number of stationary sources is sufficiently large, our problem has some extra structure which can be used to develop a more efficient algorithm. With this insight, we will provide an algorithm which solves SSA in the exact case, i.e. the algorithm finds the exact solution when there is no noise. This is nearly never the case in real applications. Therefore, we will reformulate the exact algorithm using approximate algebra and obtain an algorithm which will both work in the exact and in the approximate case.

After deriving the Low-Rank SSA algorithm, we will investigate its performance and compare it to the classical and algebraic algorithm with focus on accuracy and runtime.

1.4 Notation

We will use common notation in this thesis. But since there are sometimes multiple common notations, depending on country, language and field of research, we want to fix some of these. If A is a matrix, we will denote its transpose by $A^\top$. The internal sum (i.e. the span) of two vector spaces V and W is denoted as $V + W$. If it is an internal direct sum, i.e. $V \cap W = \{0\}$, we will denote this by writing $V \oplus W$.

Chapter 2

Stationary Subspace Analysis

Stationary Subspace Analysis is a blind source separation problem with a linear mixing model, which factorizes a time series into stationary and non-stationary components. In this chapter, we will describe the Stationary Subspace Analysis problem formally. First, we will explain what *stationary* means. Afterwards, we describe the mixing model of SSA and explain the algebraic structure of the problem, which is followed by the derivation of the additional structure which we have in case of Low-Rank SSA. At the end of this chapter, we will present algorithms which use this structure to solve the SSA problem more efficiently.

2.1 Stationarity

In statistics, there are many different concepts of stationarity. In Stationary Subspace Analysis, we use a quite weak definition of stationarity, which we will have to introduce first.

Definition 2.1.1 (Time series)**.** Let $T \subseteq \mathbb{R}$ and let $\{X_t\}_{t \in T}$ be a family of random vectors in $\mathbb{R}^n$. Then $\{X_t\}_{t \in T}$ is called a (multivariate) *time series*.

Definition 2.1.2 (Stationarity in SSA)**.** Let $\{X_t\}_{t \in T}$ be a time series. In SSA, we will call $\{X_t\}_{t \in T}$ *stationary* if and only if the means and the covariance matrices of the random vectors X_t stay constant over time, i.e.

$$\begin{aligned} \mathrm{E}(X_t) &= \mathrm{E}(X_{t'}), \\ \mathrm{Cov}(X_t) &= \mathrm{Cov}(X_{t'}) \ \forall t, t' \in T. \end{aligned}$$

Otherwise, it is called *non-stationary*.

Our definition of stationarity is even weaker than *weak stationarity* [Pri83], as we do not consider the cross-correlation.

For simplicity reasons, we will only concentrate on the second cumulant, i.e. on the covariance matrices. In principle, it is no problem to additionally include the means as well.

2.2 The mixing model

We will now explain the linear mixing model of Stationary Subspace Analysis.

Definition 2.2.1 (SSA mixing model)**.** Let $\{X_t\}_{t\in T}$ be a time series in $\mathbb{R}^D$, which is a linear superposition of a stationary time series $\{X_t^{\mathfrak{s}}\}_{t\in T}$ in $\mathbb{R}^d$ and a non-stationary time series $\{X_t^{\mathfrak{n}}\}_{t\in T}$ in $\mathbb{R}^{D-d}$, i.e. there is a *mixing matrix* $A \in \mathbb{R}^{D\times D}$ such that

$$X_t = A \begin{bmatrix} X_t^{\mathfrak{s}} \\ X_t^{\mathfrak{n}} \end{bmatrix} = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} X_t^{\mathfrak{s}} \\ X_t^{\mathfrak{n}} \end{bmatrix}, \tag{2.1}$$

where we call $\{X_t^{\mathfrak{s}}\}_{t\in T}$ the *stationary part* and $\{X_t^{\mathfrak{n}}\}_{t\in T}$ the *non-stationary part* of the time series. The columns of $A^{\mathfrak{s}} \in \mathbb{R}^{D\times d}$ and $A^{\mathfrak{n}} \in \mathbb{R}^{D\times(D-d)}$ span the *stationary subspace* and *non-stationary subspace*, respectively. Overall, this model is called the *Stationary Subspace Analysis mixing model*.

Based on this mixing model, we can formulate the Stationary Subspace Analysis Problem, as we will use it in Low-Rank SSA. We assume that our mixing matrix A is invertible. Then it follows that

$$A^{-1}X_t = \begin{bmatrix} X_t^{\mathfrak{s}} \\ X_t^{\mathfrak{n}} \end{bmatrix}.$$

Let $P = A^{-1}|_{d\times D}$ be the first d rows of A^{-1}. Then

$$PX_t = X_t^{\mathfrak{s}}.$$

This means, that by projecting our time series to the rows of P, we get a stationary time series, namely the time series consisting of our original stationary sources. The goal of SSA is to find a projection P', such that the projected time series is stationary, only by knowing the covariance matrices of some time points $t_1, \dots, t_N \subseteq T$. It is not possible to identify the true projection matrix P to the original stationary sources, since any arbitrary linear mixture of the stationary sources is stationary again. However, it is possible to identify the space which is spanned by the rows of P.

Let $\Sigma = \mathrm{Cov}(X_t)$ be the covariance matrix of our time series at time $t \in T$. Then the covariance matrix of the projected time series PX_t is $\mathrm{Cov}(PX_t) = PX_tP^\top$. Our goal is to find a projection matrix P, such that the projected covariance matrices are equal at the time points $t_1, \dots, t_N$.

There are two formulations of the SSA problem: the exact and the approximate one. In the exact case, we assume that we have access to the exact covariance matrices $\Sigma_1, \dots, \Sigma_N$ for some time points $t_1, \dots, t_N \subseteq T$. Under certain assumptions, we will be able to invert the SSA generative model, i.e. we will be able to determine the space which is spanned by the rows of P.

In the approximate case, we will only have access to a realization $x(t)$ of the time series X_t. We have to assume, that we can estimate covariance matrices $\hat{\Sigma}_1, \dots, \hat{\Sigma}_N$, for example by dividing the $x(t)$ into epochs. Since we have only finite data, the estimations are not exact. Therefore, it is not possbile to invert the SSA generative model exactly — instead, we will have to construct an estimator $\hat{P}$, such that the projections of the estimated covariance matrices $\hat{P}\hat{\Sigma}_i\hat{P}^\top$ are as stationary as possible.

First, we will show how to solve the exact problem. Afterwards, we will replace the exact covariance matrices $\Sigma_1, \ldots, \Sigma_N$ by estimations $\hat{\Sigma}_1, \ldots, \hat{\Sigma}_N$ and show how the exact algebraic calculations can be transformed into a consistent estimator $\hat{P}$.

The following formulation is for the exact case.

Problem 2.2.2 (SSA problem in the exact case)**.** Let $t_1, \ldots, t_N \in T$ be time points and let

$$\Sigma_i = \mathrm{Cov}(X_{t_i})$$

be the corresponding covariance matrices of our time series for $i = 1, \ldots, N$. The goal of SSA is to find a projection matrix $P \in \mathbb{R}^{d \times D}$ with $\operatorname{rank} P = d$, such that

$$P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_N P^\top, \tag{2.2}$$

i.e. the projection of the covariance matrices to the rows of P is stationary.

The next formulation is for the approximate case.

Problem 2.2.3 (SSA problem in the approximate case)**.** Let $x(t) \in \mathbb{R}^D$ be a realization of the time series X_t. We assume, that we can estimate covariance matrices $\hat{\Sigma}_1, \ldots, \hat{\Sigma}_N \in \mathbb{R}^{D \times D}$, for example by dividing $x(t)$ into epochs. The goal of SSA in the approximate case is to estimate a projection matrix $\hat{P} \in \mathbb{R}^{d \times D}$ with $\operatorname{rank} \hat{P} = d$, such that

$$\hat{P}\hat{\Sigma}_i \hat{P}^\top \approx \hat{P}\hat{\Sigma}_j \hat{P}^\top \ \forall i, j = 1, \ldots, N, \tag{2.3}$$

i.e. the projection of the covariance matrices to the rows of $\hat{P}$ are nearly stationary, for example by minimizing the quadratic error.

Definition 2.2.4 (Space of stationary projections)**.** The space which is spanned by the rows of P is called the space of stationary projections and will be denoted with S, i.e.

$$S = \operatorname{span}(p_1, \ldots, p_d).$$

We will see later that any other $P' \in \mathbb{R}^{d \times D}$ whose rows are a basis for S will have the property (2.2). This is the reason why we call S the space of stationary projections. Under certain assumptions, one can show that the space S is unique and identifiable. We will also see this later.

The SSA problem as it is formulated here is an algebraic problem [KvBM$^+$12a] [KvBM$^+$12b], in contrast to the classical SSA problem, which is formulated as an optimization problem.

2.3 Genericity in linear algebra

In the derivation of the algebraic structure of Low-Rank SSA, we will need generic linear algebra. At this point, we will only explain the basic ideas of genericity. For further details, we refer to other literature, for example [Par10] and [FH94].

Intuitively, an algebraic object is *generic*, if the parameters which characterize it are drawn independently from a continous and positive probability distribution. This means that those parameters do not fulfill any special equation, except for the ones which are assumed.

If we say that an algebraic object is generic with respect to some property, we mean that the parameters which charaterize the object do not fulfill any special equation except the equations which are implied by this property.

Definition 2.3.1 (genericity)**.** Let $x \in \mathbb{R}^\alpha$ be a vector, whose entries uniquely parameterize an (algebraic) object $M(x)$, for example $M(x)$ could be a real number, vector, matrix or a set of such objects. Let $A(M(x))$ be a statement about the object $M(x)$. Then we say

"a generic object $M(x)$ fulfills the statement $A(M(x))$"

if and only if

"$A(M(X))$ is fulfilled with probability 1 for all $\mathbb{R}^\alpha$-valued random vectors X with a continous and positive density function".

There are several things, which follow from this definition [KvBM+12a]. We will not derive them — instead, we will sum up the results which are relevant for us:

1. Let $\lambda \in \mathbb{R}$ be a generic real number and let $x \in \mathbb{R}$ be an arbitrary real number. Then $\lambda \neq x$.

2. Let $v_1, \ldots, v_m \in \mathbb{R}^n$ be generic vectors. Then

$$\dim \operatorname{span}(v_1, \ldots, v_m) = \min\{m, n\}.$$

   In other words: Generic vectors are linearly independent as long as the dimension of their vector space allows it.

3. Let $A \in \mathbb{R}^{m \times n}$ be a generic matrix. Then $\operatorname{rank} A = \min\{m, n\}$.

4. Let $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times m}$ be generic matrices with $n \geq m$. Then their product $AB \in \mathbb{R}^{m \times m}$ is a generic matrix.

5. Let $v_1, \ldots, v_m \in \mathbb{R}^n$ be generic vectors with $m < n$ and let $U = \operatorname{span}(v_1, \ldots, v_m)$. Then U is a generic vector space with $\dim U = m$.

6. Let $U, V \subseteq \mathbb{R}^n$ be generic linear subspaces with $\dim U + \dim V \leq n$. Then $U + V = U \oplus V$.

2.4 Algebraic Structure of Low-Rank SSA

We will now derive the algebraic structure of Low-Rank SSA, if we know the covariance matrices $\Sigma_1, \ldots, \Sigma_N$ exactly. In particular, we will see that our algebraic Problem 2.2.2 has some additional structure if the number of stationary dimensions d is larger than half of the number of total dimensions D, i.e. if $d > \frac{D}{2}$. This additional structure can be used to solve the Stationary Subspace Analysis problem more efficiently.

In the following two lemmas we will show, that the SSA problem of finding a projection matrix P to the space of stationary projections S can be reduced to the problem of finding the common zero set of quadratic forms $p^\top Q_1 p, \ldots, p^\top Q_{N-1} p$.

Lemma 2.4.1. Let

$$\overline{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}\Sigma_i$$

be the mean of the covariance matrices and let $Q_i := \Sigma_i - \overline{\Sigma}$ for all $i = 1, \ldots, N$. Then

$$P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_N P^\top$$

holds if and only if

$$PQ_1 P^\top = PQ_2 P^\top = \cdots = PQ_{N-1} P^\top = 0.$$

Proof. We first show, that

$$\begin{aligned} &P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_N P^\top \\ \Leftrightarrow &P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_{N-1} P^\top = P\overline{\Sigma} P^\top \end{aligned} \tag{2.4}$$

"$\Rightarrow$": Since all $P\Sigma_i P^\top$ are equal, they are equal to their mean, which is

$$\frac{1}{N}\sum_{i=1}^{N} P\Sigma_i P^\top = P\left(\frac{1}{N}\sum_{i=1}^{N}\Sigma_i\right)P^\top = P\overline{\Sigma}P^\top.$$

"$\Leftarrow$": $P\Sigma_N P^\top$ can be written as

$$\begin{aligned} P\Sigma_N P^\top &= P\left(\sum_{i=1}^{N}\Sigma_i - \sum_{i=1}^{N-1}\Sigma_i\right)P^\top \\ &= P\left(N\overline{\Sigma} - \sum_{i=1}^{N-1}\Sigma_i\right)P^\top \\ &= NP\overline{\Sigma}P^\top - \sum_{i=1}^{N-1} P\Sigma_i P^\top \\ &= NP\overline{\Sigma}P^\top - (N-1)P\overline{\Sigma}P^\top \\ &= P\overline{\Sigma}P^\top \end{aligned}$$

Therefore, $P\overline{\Sigma}P^\top$ can be replaced by $P\Sigma_N P^\top$.

Now we show the equivalence asserted in the Lemma:

$$\begin{aligned} &P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_N P^\top \\ \Leftrightarrow &P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_{N-1} P^\top = P\overline{\Sigma} P^\top \\ \Leftrightarrow &P(\Sigma_1 - \overline{\Sigma})P^\top = P(\Sigma_2 - \overline{\Sigma})P^\top = \cdots = P(\Sigma_{N-1} - \overline{\Sigma})P^\top = P(\overline{\Sigma} - \overline{\Sigma})P^\top \\ \Leftrightarrow &PQ_1 P^\top = PQ_2 P^\top = \cdots = PQ_{N-1} P^\top = 0 \end{aligned}$$

□

Lemma 2.4.2. Let $Q \in \mathbb{R}^{D\times D}$ be a symmetric matrix. Let

$$P = (p_1, \ldots, p_d)^\top \in \mathbb{R}^{d\times D}$$

be an arbitrary matrix, where $p_1, \ldots, p_d \in \mathbb{R}^D$ are the rows of P. Then the following is equivalent:

(i) $PQP^\top = 0$

(ii) $\forall p \in \text{span}(p_1, \ldots, p_d) : \; p^\top Qp = 0$

Proof.
"(i) $\Rightarrow$ (ii)": Let $PQP^\top = 0$. Let $p \in \text{span}(p_1, \ldots, p_d)$ be an arbitrary vector in the row span of P. Then there is a vector $w \in \mathbb{R}^d$ such that $p = P^\top w$. It follows

$$p^\top Qp = (P^\top w)^\top Q(P^\top w) = w^\top (PQP^\top)w = 0$$

where the last equality follows from $PQP^\top = 0$.

"(i) $\Leftarrow$ (ii)": Let $p^\top Qp = 0$ for all $p \in \text{span}(p_1, \ldots, p_d)$. Let $i, j \in \{1, \ldots, d\}$ be an arbitrary row and column index of $PQP^\top$, respectively. It holds:

$$(PQP^\top)_{ij} = p_i^\top Qp_j$$

Since $p_i, p_j, (p_i + p_j) \in \text{span}(p_1, \ldots, p_d)$ we know that

$$p_i^\top Qp_i = p_j^\top Qp_j = (p_i + p_j)^\top Q(p_i + p_j) = 0.$$

As Q is symmetric, it follows

$$0 = (p_i + p_j)^\top Q(p_i + p_j) = p_i^\top Qp_i + 2p_i^\top Qp_j + p_j^\top Qp_j = 2p_i^\top Qp_j$$

and therefore $(PQP^\top)_{ij} = p_i^\top Qp_j = 0$. Since i and j have been arbitrarily chosen, it follows that $PQP^\top = 0$. □

A direct consequence of this lemma is the following corollary, which we will need in our first Theorem.

Corollary 2.4.3. Let $Q \in \mathbb{R}^{D\times D}$ be a symmetric matrix. Let

$$P = (p_1, \ldots, p_d)^\top \in \mathbb{R}^{d\times D}$$

be an arbitrary matrix, where $p_1, \ldots, p_d \in \mathbb{R}^D$ are the rows of P. Let

$$P' = (p'_1, \ldots, p'_d)^\top \in \mathbb{R}^{d\times D}$$

be another matrix whose rows span the same space as the rows of P, i.e. $\text{span}(p_1, \ldots, p_d) = \text{span}(p'_1, \ldots, p'_d)$. Then it holds:

$$PQP^\top = 0 \Leftrightarrow P'Q(P')^\top = 0$$

In particular, the rows of P' can be chosen orthogonal.

Proof. Using Lemma 2.4.2 it holds:

$$\begin{aligned} & PQP^\top = 0 \\ \Leftrightarrow & \forall p \in \operatorname{span}(p_1, \ldots, p_d) : \ p^\top Q p = 0 \\ \Leftrightarrow & \forall p \in \operatorname{span}(p'_1, \ldots, p'_d) : \ p^\top Q p = 0 \\ \Leftrightarrow & P' Q (P')^\top = 0 \end{aligned}$$

□

Corollary 2.4.3 implies, that the set of stationary projections is a vector space. Or in other words: Knowing a projection matrix $P \in \mathbb{R}^{d \times D}$ of rank d fulfilling equation (2.2) is equivalent to know its row space, which is the space of stationary projections.

Assumption 2.4.4. We will assume that $\Sigma_1, \ldots, \Sigma_N$ are generic covariance matrices. Let $Q \in \mathbb{R}^{D \times D}$ be one of our matrices Q_i. It follows that Q has the properties of being

(i) symmetric and

(ii) fulfilling $PQP^\top = 0$.

Additionally, we will assume that

(iii) apart from fulfilling (i) and (ii), Q is generic.

For the next Theorem, we will need a technical Lemma, which we will prove first.

Lemma 2.4.5. Let $A \in \mathbb{R}^{n \times n}$ be a matrix with $\operatorname{rank} A = n$. Let $B \in \mathbb{R}^{n \times n}$ be an arbitrary matrix and let $\lambda \in \mathbb{R}$ be generic. Then

$$\operatorname{rank}(A + \lambda B) = n.$$

Proof. The determinant of $A + \lambda B$ is a polynomial in λ which we denote by

$$p(\lambda) = \det(A + \lambda B).$$

By definition of p, it holds that

$$p(0) = \det(A + 0B) = \det(A) \neq 0$$

so p is not the zero polynomial. This implies that p has only a finite number of zeros. Thus, for generic choice of λ, $\det(A + \lambda B) \neq 0$, i.e., $A + \lambda B$ has full rank for generic λ. □

In the following Theorem we will show that our matrices Q_i are rank deficient, if the number of stationary sources is large. Actually, $\operatorname{rank} Q_i < D$ if $d > \frac{D}{2}$. Furthermore, we will see that $S^\perp$ is a linear subspace of $\operatorname{span} Q_i$ for all i. This additional structure is the main observation and will be used to construct the Low-Rank SSA algorithm.

Theorem 2.4.6. Let $Q \in \mathbb{R}^{D\times D}$ be a matrix with properties (i) - (iii) from Assumption 2.4.4. Then it holds that

$$\operatorname{rank} Q = \min\left(2(D-d), D\right).$$

Furthermore, let $S = \operatorname{span}(p_1, \ldots, p_d)$ be the vector space of stationary projections. Then

$$\operatorname{span} Q = S^{\perp} + R$$

where R is a generic $(D-d)$-dimensional subspace of $\mathbb{R}^D$.

Proof. Let $P \in \mathbb{R}^{d\times D}$ be a projection matrix as in property (ii) whose rows are orthonormal (such a P exists because of Corollary 2.4.3). We can complete the rows of P to an orthonormal basis of $\mathbb{R}^{1\times D}$ and write it into the rows of a matrix P_0:

$$P_0 = \begin{pmatrix} P \\ P_{\perp} \end{pmatrix} \in \mathbb{R}^{D\times D}$$

Without loss of generality we can prove the assertion in P_0-coordinates. This is possible, since our assertion is basis independent. Transforming the rows of P into our new basis leads to

$$\tilde{P} := PP_0^{-1} = PP_0^{\top} = P\begin{pmatrix} P^{\top} & P_{\perp}^{\top} \end{pmatrix} = \begin{pmatrix} PP^{\top} & PP_{\perp}^{\top} \end{pmatrix} = \begin{pmatrix} I & 0 \end{pmatrix}$$

and Q written in our new basis is

$$\begin{aligned} \tilde{Q} &:= P_0 Q P_0^{-1} \\ &= P_0 Q P_0^{\top} \\ &= \begin{pmatrix} P \\ P_{\perp} \end{pmatrix} Q \begin{pmatrix} P^{\top} & P_{\perp}^{\top} \end{pmatrix} \\ &= \begin{pmatrix} PQP^{\top} & PQP_{\perp}^{\top} \\ P_{\perp}QP^{\top} & P_{\perp}QP_{\perp}^{\top} \end{pmatrix} \\ &= \begin{pmatrix} 0 & A \\ A^{\top} & B \end{pmatrix} \end{aligned}$$

where $A := PQP_{\perp}^{\top} \in \mathbb{R}^{d\times(D-d)}$ and $B := P_{\perp}QP_{\perp}^{\top} \in \mathbb{R}^{(D-d)\times(D-d)}$. We will first determine the rank of $\tilde{Q}$ and then its span.

Case 1: $2(D-d) < D$. This is equivalent to $d > (D-d)$. From assumption (iii) we know, that our matrix A is generic and since $d > (D-d)$ it follows $\operatorname{rank} A = (D-d)$, which means that the rows of A span the whole row space of the last $(D-d)$ columns of $\tilde{Q}$. Therefore, we can use the upper part $\begin{pmatrix} 0 & A \end{pmatrix}$ of $\tilde{Q}$ to eliminate B completely using Gaussian elimination. The resulting matrix is

$$\tilde{Q}' = \begin{pmatrix} 0 & A \\ A^{\top} & 0 \end{pmatrix}$$

and has the same span as Q. The upper part $\begin{pmatrix} 0 & A \end{pmatrix}$ and the lower part $\begin{pmatrix} A^{\top} & 0 \end{pmatrix}$ of $\tilde{Q}'$ are linearly independent and

$$\operatorname{rank}\begin{pmatrix} 0 & A \end{pmatrix} = \operatorname{rank}\begin{pmatrix} A^{\top} & 0 \end{pmatrix} = D-d$$

because of the genericity of A. Thus, the rank of $\tilde{Q}'$ is $2(D-d)$ and

$$\operatorname{rank}\tilde{Q} = \operatorname{rank}\tilde{Q}' = 2(D-d) = \min\left(2(D-d), D\right).$$

Now we will determine the span of $\tilde{Q}'$. Let $M \in \mathbb{R}^{(D-d)\times d}$ be an arbitrary generic matrix and let

$$N := \begin{pmatrix} I & 0 \\ M & I \end{pmatrix} \in \mathbb{R}^{D\times D}.$$

Multiplying N with $\tilde{Q}'$ from the left side yields:

$$\tilde{Q}'' := N\tilde{Q}' = \begin{pmatrix} I & 0 \\ M & I \end{pmatrix}\begin{pmatrix} 0 & A \\ A^\top & 0 \end{pmatrix} = \begin{pmatrix} 0 & A \\ A^\top & MA \end{pmatrix}$$

Notice, that $\operatorname{rowspan}\tilde{Q} = \operatorname{rowspan}\tilde{Q}' = \operatorname{rowspan}\tilde{Q}''$ as we only performed invertible row operations.

The upper part $\begin{pmatrix} 0 & A \end{pmatrix}$ of $\tilde{Q}''$ has dimension $(D-d)$ and is orthogonal to the space $\tilde{S} := \operatorname{rowspan}\tilde{P}$:

$$\begin{pmatrix} 0 & A \end{pmatrix}\tilde{P}^\top = \begin{pmatrix} 0 & A \end{pmatrix}\begin{pmatrix} I \\ 0 \end{pmatrix} = 0\cdot I + A\cdot 0 = 0$$

Therefore, the upper part of $\tilde{Q}''$ spans $\tilde{S}^\perp$. The lower part $\begin{pmatrix} A^\top & MA \end{pmatrix}$ of $\tilde{Q}''$ is generic: $A^\top$ is generic by assumption and MA is generic, as the $(D-d)$ rows of MA are generic linear combinations of the rows of A and the row space of A is $(D-d)$-dimensional. Let

$$\tilde{R} := \operatorname{rowspan}\begin{pmatrix} A^\top & MA \end{pmatrix}$$

The dimension of $\tilde{R}$ is $(D-d)$, as it is spanned by $(D-d)$ generic vectors in $\mathbb{R}^D$. Summarizing the last results it follows

$$\operatorname{span}\tilde{Q} = \operatorname{rowspan}\tilde{Q} = \operatorname{rowspan}\tilde{Q}'' = \tilde{S}^\perp + \tilde{R},$$

which proves the assertion.

Case 2: $2(D-d) \geq D$. This is equivalent to $d \leq (D-d)$. Let $\tilde{Q}|_{(2d)\times(2d)}$ be the matrix $\tilde{Q}$ restricted to the first $2d$ rows and columns. It has the structure

$$\tilde{Q}|_{(2d)\times(2d)} = \begin{pmatrix} 0 & C \\ C^\top & D \end{pmatrix},$$

where $C, D \in \mathbb{R}^{d\times d}$ and C is a generic matrix. In particular, $\det C \neq 0$ and $\det\tilde{Q}|_{(2d)\times(2d)} = 2\det C \neq 0$ from which follows $\operatorname{rank}\tilde{Q}|_{(2d)\times(2d)} = 2d$.

We will now show by induction over k, that the matrices $\tilde{Q}|_{k\times k}$ for $k = 2d, \ldots, D$ also have full rank. We just showed this for $k = 2d$. Now assume that the assertion holds for k, i.e. $\operatorname{rank}\tilde{Q}|_{k\times k} = k$. The entry $\tilde{Q}_{(k+1)\times(k+1)} \neq 0$, since it is generic. Therefore, the matrix $\operatorname{diag}(\tilde{Q}|_{k\times k}, \tilde{Q}_{(k+1)\times(k+1)})$ has the determinant

$$\det\begin{pmatrix} \tilde{Q}|_{k\times k} & 0 \\ 0 & \tilde{Q}_{(k+1)\times(k+1)} \end{pmatrix} = \det\tilde{Q}|_{k\times k}\det\tilde{Q}_{(k+1)\times(k+1)} \neq 0,$$

from which follows that $\operatorname{rank}\left(\operatorname{diag}(\tilde{Q}|_{k\times k}, \tilde{Q}_{(k+1)\times(k+1)})\right) = k+1$. We can now complete the matrix $\operatorname{diag}(\tilde{Q}|_{k\times k}, \tilde{Q}_{(k+1)\times(k+1)})$ to the matrix $Q|_{(k+1)\times(k+1)}$:

Let $B_i := e_{k+1}e_i^\top + e_i e_{k+1}^\top \in \mathbb{R}^{D\times D}$ be the matrix which has a 1 at the indices $(k+1, i)$ and $(i, k+1)$ and zeros elsewhere. Then

$$Q|_{(k+1)\times(k+1)} = \operatorname{diag}(\tilde{Q}|_{k\times k}, \tilde{Q}_{(k+1)\times(k+1)}) + \sum_{i=1}^{k} Q_{k+1,i} B_i.$$

Since the $Q_{k+1,i} \in \mathbb{R}$ are generic, we can subsequently apply Lemma 2.4.5 and it follows that $\operatorname{rank} Q|_{(k+1)\times(k+1)} = k+1$, which proves the assertion for $k+1$.

In particular, we now have proven that $\operatorname{rank}\tilde{Q} = D = \min(2(D-d), D)$. The assertion about the span of $\tilde{Q}$ follows immediately, since $\operatorname{span}\tilde{Q} = \mathbb{R}^D$ which can be written as

$$\mathbb{R}^D = S^\perp + R$$

for an arbitrary generic vector space R of dimension $(D-d)$. □

Now that we know that the subspaces spanned by the Q_i contain the orthogonal space of S, it is self-evident to try to calculate $S^\perp$ by finding the intersection of those subspaces. The question arises how many of the Q_i we need to determine $S^\perp$ uniquely. The following proposition and corollary will help us to find an answer for this question.

Proposition 2.4.7. Let $V \subseteq \mathbb{R}^D$ be a vector space and let $G_1, G_2 \subseteq \mathbb{R}^D$ be generic vector spaces such that $V \subseteq G_1$ and $V \subseteq G_2$. Then

$$G := G_1 \cap G_2 = V \oplus R$$

where $R \subseteq \mathbb{R}^D$ is a generic sub-vector space of $V^\perp$ with

$$\dim R = \max(\dim G_1 + \dim G_2 - \dim V - D, 0)$$

In particular, G is a generic vector space with $V \subseteq G$ and

$$\dim G = \dim V + \dim R$$

Proof. We can write G_1 and G_2 as the direct sums

$$G_1 = V \oplus \tilde{G}_1$$
$$G_2 = V \oplus \tilde{G}_2$$

where $\tilde{G}_1$ and $\tilde{G}_2$ are generic sub-vector spaces with $\tilde{G}_1 \subseteq V^\perp$ and $\tilde{G}_2 \subseteq V^\perp$. Define $R := \tilde{G}_1 \cap \tilde{G}_2$. Then it holds

$$G = G_1 \cap G_2 = V \oplus (\tilde{G}_1 \cap \tilde{G}_2) = V \oplus R.$$

In the following, we will regard $\tilde{G}_1$ and $\tilde{G}_2$ as subspaces of $V^\perp$. In particular, the orthogonalization operator $(.)^\perp$ applied to $\tilde{G}_1, \tilde{G}_2$ or $\tilde{G}_1 \cap G_2$ dualizes in $V^\perp$ instead of in $\mathbb{R}^D$.

$$\begin{aligned}
\dim R &= \dim(\tilde{G}_1 \cap \tilde{G}_2) \\
&= \operatorname{codim}\left((\tilde{G}_1 \cap \tilde{G}_2)^\perp \subseteq V^\perp\right) \\
&= \operatorname{codim}\left(\tilde{G}_1^\perp + \tilde{G}_2^\perp \subseteq V^\perp\right) \\
&= \dim V^\perp - \dim\left(\tilde{G}_1^\perp + \tilde{G}_2^\perp\right)
\end{aligned} \tag{2.5}$$

Since $\tilde{G}_1^\perp$ and $\tilde{G}_2^\perp$ are generic subspaces of $V^\perp$, they are linearly independent as long as $\dim \tilde{G}_1^\perp + \dim \tilde{G}_2^\perp \leq \dim V^\perp$. In this case, it follows

$$\dim\left(\tilde{G}_1^\perp + \tilde{G}_2^\perp\right) = \dim \tilde{G}_1^\perp + \dim \tilde{G}_2^\perp.$$

On the other hand, if $\dim \tilde{G}_1^\perp + \dim \tilde{G}_2^\perp > \dim V^\perp$, $\tilde{G}_1^\perp + \tilde{G}_2^\perp$ will span the whole space $V^\perp$ and therefore

$$\dim\left(\tilde{G}_1^\perp + \tilde{G}_2^\perp\right) = \dim V^\perp.$$

Putting both together we get the case-independent equation

$$\dim\left(\tilde{G}_1^\perp + \tilde{G}_2^\perp\right) = \min\left(\dim \tilde{G}_1^\perp + \dim \tilde{G}_2^\perp, \dim V^\perp\right).$$

Now we can continue with calculation (2.5):

$$\begin{aligned}
\dim R &= \dim(\tilde{G}_1 \cap \tilde{G}_2) \\
&= \dim V^\perp - \dim\left(\tilde{G}_1^\perp + \tilde{G}_2^\perp\right) \\
&= \dim V^\perp - \min\left(\dim \tilde{G}_1^\perp + \dim \tilde{G}_2^\perp, \dim V^\perp\right) \\
&= \max\left(\dim V^\perp - (\dim \tilde{G}_1^\perp + \dim \tilde{G}_2^\perp), \dim V^\perp - \dim V^\perp\right) \\
&= \max\left(\dim V^\perp - \dim \tilde{G}_1^\perp - \dim \tilde{G}_2^\perp, 0\right) \\
&= \max\left(\dim V^\perp - (\dim V^\perp - \dim \tilde{G}_1) - (\dim V^\perp - \dim \tilde{G}_2), 0\right) \\
&= \max\left(\dim \tilde{G}_1 + \dim \tilde{G}_2 - \dim V^\perp, 0\right) \\
&= \max\left(\dim \tilde{G}_1 + \dim \tilde{G}_2 - (D - \dim V), 0\right) \\
&= \max\left((\dim G_1 - \dim V) + (\dim G_2 - \dim V) - (D - \dim V), 0\right) \\
&= \max\left(\dim G_1 + \dim G_2 - \dim V - D, 0\right)
\end{aligned}$$

□

The result of Proposition 2.4.7 is generalized to arbitrary finite intersections by Corollary 2.4.8 using induction.

Corollary 2.4.8. Let $V \subseteq \mathbb{R}^D$ be a vector space and let $G_1, \ldots, G_n$ be generic vector spaces such that $V \subseteq G_i$ and $\dim G_i = k$ for all $i = 1, \ldots, n$. Then

$$\bigcap_{i=1}^{n} G_i = V \oplus R,$$

where $R \subseteq \mathbb{R}^D$ is a generic sub-vector space of $V^\perp$ with

$$\dim R = \max\left(D - \dim V - n(D-k), 0\right)$$

Proof. We will show the assertion by an induction over n.

Let $n = 1$. Then by assumption

$$\bigcap_{i=1}^{n} G_i = G_1 = V \oplus R$$

and

$$\begin{aligned}\dim R &= \dim G_1 - \dim V \\ &= k - \dim V \\ &= D - \dim V - n(D-k) \\ &= \max\left(D - \dim V - n(D-k), 0\right)\end{aligned}$$

Now assume that the assertion holds for n. Then

$$G := \bigcap_{i=1}^{n} G_i = V \oplus R_n$$

with

$$\dim R_n = \max\left(D - \dim V - n(D-k), 0\right).$$

Since

$$\bigcap_{i=1}^{n+1} G_i = \left(\bigcap_{i=1}^{n} G_i\right) \cap G_{n+1} = G \cap G_{n+1}$$

we can use Proposition 2.4.7 and it follows:

$$\begin{aligned}\dim R &= \max\left(\dim G + \dim G_{n+1} - \dim V - D, 0\right) \\ &= \max\left((\dim V + \dim R_n) + k - \dim V - D, 0\right) \\ &= \max\left(\dim R_n + k - D, 0\right) \\ &= \max\left(\max\left(D - \dim V - n(D-k), 0\right) - (D-k), 0\right) \\ &= \max\left(D - \dim V - (n+1)(D-k), 0\right)\end{aligned}$$

This proves the assertion for $n + 1$. □

From Theorem 2.4.6 we know that the orthogonal space of S can be calculated by intersecting a *sufficient number* of subspaces spanned by the Q_i. Together with Corollary 2.4.8 we can determine this number exactly, which leads to our main theorem.

Theorem 2.4.9. Let $Q_1, \ldots, Q_M \in \mathbb{R}^{D \times D}$ be matrices with properties (i) - (iii) from Assumption 2.4.4. Then

$$\bigcap_{i=1}^{M} \operatorname{span}(Q_i) = S^{\perp} \oplus R$$

where R is a generic sub-vector space of S with dimension

$$\dim R = \begin{cases} \max\left(d - M(2d - D), 0\right), & d > \frac{D}{2} \\ d, & d \leq \frac{D}{2} \end{cases}$$

In particular, if $d > \frac{D}{2}$ and

$$M \geq \frac{d}{2d - D}$$

then one has

$$S^{\perp} = \bigcap_{i=1}^{M} \operatorname{span}(Q_i).$$

Proof. From Theorem 2.4.6 we know that

$$\operatorname{span} Q_i = S^{\perp} + R_i$$

where the R_i are generic vector spaces of dimension $(D-d)$ for all $i = 1, \ldots, M$. In particular, $\operatorname{span} Q_i$ are generic vector spaces with $S^{\perp} \subseteq Q_i$ for all $i = 1, \ldots, M$. Applying Corollary 2.4.8 we get

$$\bigcap_{i=1}^{M} \operatorname{span} Q_i = S^{\perp} \oplus R$$

where R is a generic sub-vector space of $(S^{\perp})^{\perp} = S$ having dimension

$$\begin{aligned} \dim R &= \max\left(D - \dim S^{\perp} - M(D - \dim \operatorname{span} Q_i), 0\right) \\ &= \max\left(D - (D-d) - M(D - \operatorname{rank} Q_i), 0\right) \\ &= \max\left(d - M(D - \operatorname{rank} Q_i), 0\right) \end{aligned} \tag{2.6}$$

Another result of Theorem 2.4.6 was that

$$\operatorname{rank} Q_i = \min\left(2(D-d), D\right)$$

which we can reformulate as

$$\operatorname{rank} Q_i = \begin{cases} 2(D-d), & d > \frac{D}{2} \\ D, & d \leq \frac{D}{2} \end{cases}.$$

Continuing calculation (2.6) we get

$$\begin{aligned} \dim R &= \max\left(d - M(D - \operatorname{rank} Q_i), 0\right) \\ &= \begin{cases} \max\left(d - M(D - 2(D-d)), 0\right), & d > \frac{D}{2} \\ \max\left(d - M(D-D), 0\right), & d \leq \frac{D}{2} \end{cases} \\ &= \begin{cases} \max\left(d - M(2d - D), 0\right), & d > \frac{D}{2} \\ d, & d \leq \frac{D}{2} \end{cases} \end{aligned}$$

In particular, if $d > \frac{D}{2}$ and $d - M(2d - D) \leq 0$ which is equivalent to

$$M \geq \frac{d}{2d - D}$$

we get $\dim R = 0$ from which follows

$$\bigcap_{i=1}^{M} \operatorname{span}(Q_i) = S^{\perp} \oplus R = S^{\perp} \oplus 0 = S^{\perp}.$$

□

Corollary 2.4.10 reformulates the results of Theorem 2.4.9 in a way, that it can be directly transformed into the Low-Rank SSA algorithm.

Corollary 2.4.10. Let $\Sigma_1, \ldots, \Sigma_N \in \mathbb{R}^D$ be generic covariance matrices, such that there is a projection matrix $P \in \mathbb{R}^{d \times D}$ with $\operatorname{rank} P = d$ fulfilling

$$P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_N P^\top.$$

Let $\overline{\Sigma} = \sum_{i=1}^N \Sigma_i$ be the mean of the covariance matrices and let $Q_i = \Sigma_i - \overline{\Sigma}$, $i = 1, \ldots, N-1$. Furthermore, let

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

Then it holds:

$$\sum_{i=1}^{N-1} (\operatorname{span}(Q_i))^\perp = S$$

Proof. From Lemma 2.4.1 we have, that a projection matrix $P \in \mathbb{R}^{d \times D}$ fulfills

$$P\Sigma_1 P^\top = P\Sigma_2 P^\top = \cdots = P\Sigma_N P^\top$$

if and only if

$$PQ_1 P^\top = PQ_2 P^\top = \cdots = PQ_{N-1} P^\top = 0.$$

From the assumption we know that $N \geq \frac{d}{2d-D} + 1$ which is equivalent to $N - 1 \geq \frac{d}{2d-D}$. Applying Theorem 2.4.9 with $M = N - 1$ we get

$$S^\perp = \bigcap_{i=1}^{N-1} \operatorname{span}(Q_i)$$

and it follows

$$S = (S^\perp)^\perp = \left(\bigcap_{i=1}^{N-1} \operatorname{span} Q_i \right)^\perp = \sum_{i=1}^{N-1} (\operatorname{span} Q_i)^\perp.$$

□

Corollary 2.4.11 (Identifiability of the space of stationary projections)**.** Corollary 2.4.10 shows, that the space of stationary projections is identifiable if the number of epochs N is $N \geq \frac{d}{2d-D} + 1$.

2.5 Low-Rank SSA: the exact case

Using Corollary 2.4.10 we can now construct an algorithm to calculate the space of stationary projections S. This algorithm only works for the noiseless case, where we can invert the SSA model exactly. It is the basis for the approximate algorithm, which will be introduced in the next section.

Algorithm 1 Low-Rank SSA (exact case). *Input*: Covariance matrices $\Sigma_1, \ldots, \Sigma_N \in \mathbb{R}^{D\times D}$, $N \geq \frac{d}{2d-D} + 1$, dimension d of S. *Output*: The space of stationary projections S.

1: Calculate $\overline{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \Sigma_i$.
2: Initialize $B \leftarrow \emptyset$.
3: **for** $i = 1, \ldots, N-1$ **do**
4: $\quad Q_i \leftarrow \Sigma_i - \overline{\Sigma}$.
5: $\quad$ Calculate basis B_i for $Q_i^{\perp}$.
6: $\quad B \leftarrow B \cup B_i$.
7: **end for**
8: $S \leftarrow \operatorname{span} B$.
9: **return** S.

First, Algorithm 1 calculates the mean of our covariance matrices $\overline{\Sigma}$. Then, for each i, it calculates the difference matrix Q_i and a basis for the orthogonal space of the column span of Q_i. The matrix B is the union of all those basis vectors. From Corollary 2.4.10 we know that the space of stationary projections S is the span of the vectors in B. The calculation of $\operatorname{span} B$ can be done for example using Gaussian elimination or the singular value decomposition.

Apart from calculating the space of stationary projections S, we can also estimate the dimensionality of S, if it is unknown. This is even simpler: if $d > \frac{D}{2}$, we know from Theorem 2.4.6 that for any of our Q_i it holds $\operatorname{rank} Q_i = 2(D-d)$, which is equivalent to

$$d = D - \frac{\operatorname{rank} Q_i}{2}.$$

Using this, we can formulate an algorithm for estimating d:

Algorithm 2 Estimation of d (exact case). *Input*: Covariance matrices $\Sigma_1, \ldots, \Sigma_N \in \mathbb{R}^{D\times D}$, $N \geq D$. *Output*: An estimate for the stationary dimension d.

1: Calculate $\overline{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \Sigma_i$.
2: $Q_1 \leftarrow \Sigma_1 - \overline{\Sigma}$.
3: $k \leftarrow \operatorname{rank} Q_1$.
4: **if** $k = D$ **then**
5: $\quad$ **return** $0 \leq d \leq \frac{D}{2}$.
6: **else**
7: $\quad$ **return** $d = D - \frac{k}{2}$.
8: **end if**

2.6 Low-Rank SSA: the approximate case

In real-world applications, where the Q_i are estimated from noisy data, we need approximative versions of the previous algorithms. A canonical way to approximate vector spaces is to use the *Singular Value Decomposition* (SVD) [YE36] [CGTW95] [HKPP09], which has been already used in Algebraic SSA [KvBM$^+$12b]. In particular, the SVD can be used to calculate approximative spans of a given dimension and the approximation is optimal w.r.t. the Frobenius norm.

If $d > \frac{D}{2}$ and the Q_i are estimated from noisy data, they will not have rank $2(D-d)$, as we would expect from Theorem 2.4.6. Their rank will be larger, in most cases they will be full rank matrices. Using SVD, we can calculate the approximate span of Q_i having dimension $2(D-d)$.

Let $A \in \mathbb{R}^{m \times n}$ be an arbitrary matrix. We want to find the approximate column span of A having dimension $k < D$. At first, we calculate the singular value decomposition (SVD) of A:

$$A = U \Sigma V^\top,$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m \times n}$ is a matrix having non-zero entries only on its diagonal. The values $\sigma_1, \ldots, \sigma_r$ on the diagonal of Σ are called singular values and should be ordered descendingly.

To get an approximation of the column-span of A having dimension k, we could just take the span of the first k columns of U. But in our case it is useful to keep the variance information, as we want to do further approximations on our rank reduced A, which we will write as $\hat{A}$. Let

$$\hat{\Sigma} = \Sigma|_{m \times k}$$

be the matrix Σ restricted to the first k columns. Then we define

$$\hat{A} = U \hat{\Sigma}$$

as our rank-k approximation of A. We will note the approximation process as

$$\hat{A} = \text{appspan}(A, k).$$

On the other hand, it is also possible to define an approximate rank of A. Let $\theta > 0$ be a threshold, where all singular values smaller than θ are regarded as zero. Then the approximate rank of A is the number of singular values greater than θ:

$$\operatorname*{rank}_\theta(A) = \#\{\sigma_i | \sigma_i > \theta\}$$

Now we can formulate an approximate version of Algorithm 1, by replacing calculations of spans by the calculation of approximate spans. In particular, we have to replace the calculation of the orthogonal complement of the Q_i by the calculation of the approximate orthogonal complement — this is done by calculating the approximate span of the Q_i previous to the complement calculation.

Algorithm 3 Low-Rank SSA (approximate case). *Input*: Covariance matrices $\Sigma_1, \ldots, \Sigma_N \in \mathbb{R}^{D \times D}$, $N \geq \frac{d}{2d-D} + 1$, dimension d of S. *Output*: The estimated space of stationary projections S.

1: Calculate $\overline{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} \Sigma_i$.
2: Initialize $B \leftarrow \emptyset$.
3: **for** $i = 1, \ldots, N-1$ **do**
4: $\quad Q_i \leftarrow \Sigma_i - \overline{\Sigma}$.
5: $\quad \hat{Q}_i \leftarrow \text{appspan}\,(Q_i, 2(D-d))$
6: $\quad$ Calculate basis B_i for $\hat{Q}_i^\perp$.
7: $\quad B \leftarrow B \cup B_i$
8: **end for**
9: $S \leftarrow \text{appspan}(B, d)$.
10: **return** S.

The algorithm for estimating the dimensionality of S can also be formulated in an approximate version. In this case, we replace the rank calculation of Q_1 by the mean of the approximate ranks of the Q_i, since this will give us a more reliable solution.

Algorithm 4 Estimation of d (approximate case). *Input*: Covariance matrices $\Sigma_1, \ldots, \Sigma_N \in \mathbb{R}^{D\times D}$, $N \geq D$, threshold $\theta > 0$ for the singular values. *Output*: An estimate for the stationary dimension d.

Calculate $\overline{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \Sigma_i$.
Initialize $R \leftarrow \text{zeros}(N)$.
for $i = 1, \ldots, N$ **do**
    $Q_i \leftarrow \Sigma_i - \overline{\Sigma}$.
    $R_i \leftarrow \text{rank}_\theta(Q_i)$
end for
$k \leftarrow \text{round}\left(\frac{\text{mean}(R)}{2}\right)$.
if $k = D$ **then**
    return $0 \leq d \leq \frac{D}{2}$.
else
    return $d = D - k$.
end if

Chapter 3

Simulations

In this section, we will investigate the accuracy and runtime of Low-Rank SSA experimentally and compare it to the classical SSA algorithm [vBMKM09] and Algebraic SSA [KvBM$^+$12b]. In the experiments, we will investigate the accuracy in relation to

- the dimensionality of the time series,
- the number of stationary dimensions and
- the noise in the data.

Afterwards, in the runtime experiment, we will investigate the runtime of the three algorithms in relation to the dimensionality of the time series and compare it to each other.

In the last experiment we will investigate the accuracy of our stationary dimension estimation algorithm in relation to the noise level.

3.1 Generation of the data

To measure the performance of the different SSA algorithms, we generated synthetic data. This was done as follows: first, we draw a random mixing matrix $A = (a_{ij}) \in \mathbb{R}^{D\times D}$, where each entry of A is i.i.d. and uniformly drawn from the interval $[-0.5, 0.5]$, i.e.

$$a_{ij} \sim \mathcal{U}_{[-0.5,0.5]}.$$

In the next step, we draw random covariance matrices $\tilde{\Sigma}_1, \ldots, \tilde{\Sigma}_N \in \mathbb{R}^{D\times D}$ in original coordinates, i.e. each coordinate corresponds to a source. The first d rows and columns of Σ_i correspond to the stationary sources and the last $(D-d)$ rows and columns to the non-stationary sources. This is done in the following way: first we draw a random matrix $B^{\mathfrak{s}} \in \mathbb{R}^{d\times D}$, which models the correlation of our stationary sources and stays constant over time. Each entry of $B^{\mathfrak{s}}$ is drawn i.i.d from $\mathcal{U}_{[-0.5,0.5]}$ again. Then, for each point in time $i = 1, \ldots, N$, we draw a random matrix $B_i^{\mathfrak{n}} \in \mathbb{R}^{(D-d)\times D}$, which models the correlation of our non-stationary sources at time i. With

$$B_i = \begin{bmatrix} B^{\mathfrak{s}} \\ B_i^{\mathfrak{n}} \end{bmatrix}$$

we calculate the covariance matrix in original coordinates at time i as

$$\tilde{\Sigma}_i = B_i B_i^\top = \begin{bmatrix} B^{\mathfrak{s}}(B^{\mathfrak{s}})^\top & B^{\mathfrak{s}}(B_i^{\mathfrak{n}})^\top \\ B_i^{\mathfrak{n}}(B^{\mathfrak{s}})^\top & B_i^{\mathfrak{n}}(B_i^{\mathfrak{n}})^\top \end{bmatrix}$$

It can be seen, that the correlation between the stationary sources stays constant over time, while the correlation between the non-stationary sources and the correlation between the stationary and non-stationary sources changes.

To add noise of level σ, we draw a random noise vector $\epsilon_i \in \mathbb{R}^D$, with entries i.i.d. in $\mathcal{U}_{[0,2\sigma]}$, such that the expected value of each entry in ϵ_i is σ. Then we draw a random rotation matrix $R_i \in \mathbb{R}^{D\times D}$ and calculate the noise matrix for time i as

$$N_i = R_i \operatorname{diag}(\epsilon_i) R_i^\top.$$

When we add this matrix to $\tilde{\Sigma}_i$, it will simulate positive definite noise with an average noise level of σ in all directions of $\mathbb{R}^D$.

Our final covariance matrices are calculated as

$$\Sigma_i = A(\tilde{\Sigma}_i + N_i)A^\top$$

for $i = 1, \ldots, N$.

3.2 Error measures

In an experimental setting where the true space of stationary projections S is known, we want to measure the accuracy of the solutions found by the different SSA algorithms. Let $S = \operatorname{span}(p_1, \ldots, p_d)$ be the true space of stationary projections and let $\hat{S} = \operatorname{span}(\hat{p}_1, \ldots, \hat{p}_d)$ be the estimated space of stationary projections, where $\hat{P} = (\hat{p}_1, \ldots, \hat{p}_d)^\top$ is an estimation for the projection matrix which has been found by an SSA algorithm.

One possible error measure would be the the largest *principal angle* [Jor75] of S and $\hat{S}$. This measure has been already used for example in [KvBM+12b]. Let $U, V \subseteq \mathbb{R}^D$ be linear subspaces of $\mathbb{R}^D$. Then the first principal angle of U and V is defined as

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

where $u_1 \in U$ and $v_1 \in V$ are called *principal vectors*. The rest of the principal angles and vectors are then defined recursively as

$$\begin{aligned} \theta_i &= \min\left\{ \arccos\left(\frac{u^\top v}{\|u\|\|v\|} \right) \middle| \, u \in U, v \in V, u \perp u_j, v \perp v_j \ \forall j \in \{1, \ldots, i-1\} \right\} \\ &= \angle(u_i, v_i) \end{aligned}$$

for $i = 1, \ldots, \min\{\dim U, \dim V\}$.

Nevertheless, taking the largest principal angle as an error measure has some disadvantages. For example, most of the information contained in the principal angles is lost: two different solutions having the same largest principal angle to the true non-stationary subspace would be regarded as equally good. But in fact, it is possible that the rest of the principal angles of the first solution are all

close to zero, while the rest of the principal angles of the second solution may be all close to the largest principal angle.

Therefore we used a more appropriate error measure based on the principal angles which has been already proposed in [Mei11] and which is called the *subspace error*. Let $\theta_1, \ldots, \theta_d$ be the principal angles of S and $\hat{S}$ and $k = \min\{d, D-d\}$ the maximum possible number of non-zero principal angles. Then the subspace error is defined as

$$\mathcal{E}(S, \hat{S}) := \frac{1}{k} \sum_{i=1}^{d} \sin^2(\theta_i).$$

Apart from regarding all principal angles instead of only the largest one, the subspace error has the advantage of having a good interpretation: it can be considered as the percentage of non-stationary signal power in the estimated stationary sources, if we project our time series to the estimated stationary subspace $\hat{S}$.

3.3 Influence of the number of stationary dimensions

In the first experiment, we investigated the accuracy of the three different SSA algorithms in relation to the number of stationary sources d for four different noise levels. For this purpose, we fixed the total number of dimensions to $D = 10$. Note, that we had to start with a number of stationary sources of $d = 6$, since Low-Rank SSA requires $d > \frac{D}{2}$ to work. The results can be seen in Figure 3.1.

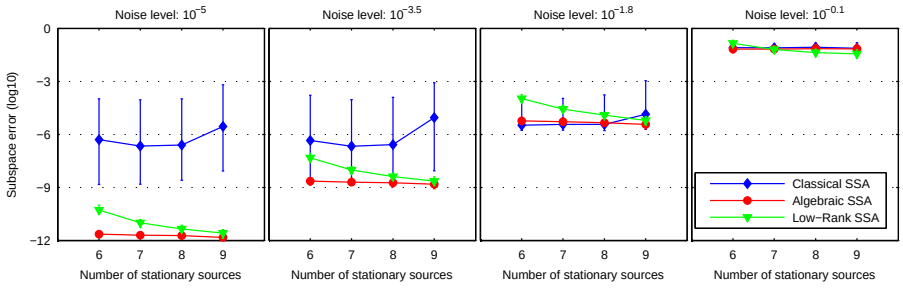

Figure 3.1: Comparison of three different SSA algorithms. Each panel shows the median error of the algorithms (vertical axis) for varying numbers of stationary sources from $d = 6$ to $d = 9$ (horizontal axis). The noise level in each panel is fixed and increases from the left to the right panel. The error bars extend from the 25% to the 75% quantile estimated over 500 random realizations of the data set.

The blue line shows the median error of the classical, optimization based SSA algorithm working only on the covariance matrices, the red line the Algebraic SSA algorithm and the green line shows the performance of Low-Rank SSA.

It can be seen, that for small noise levels, Algebraic SSA and Low-Rank SSA outperform the classical SSA algorithm. The accuracy of Low-Rank SSA seems

to improve with increasing number of stationary sources. This is something we expected, since the Low-Rank method estimates the orthogonal space $S^\perp$ and $\dim S^\perp = D - d$, which means that the dimension of the space to estimate gets smaller with increasing d. For the classical algorithm, there seems to be a minimum error below which it cannot fall. This is maybe due to the flatness of the SSA objective function close to the optimal solution [HKW+12], which makes it hard to optimize.

The performance of the Low-Rank SSA is not significantly worse than the performance of Algebraic SSA and the error which both algorithms make is quite stable, in contrast to the classical algorithm, whose error has a high variance. This is also not surprising, since the optimization based approach of the classical algorithm suffers from convergence to local minima, which is not the case for the algebraic and low-rank algorithm.

At the highest noise level, all algorithms generate almost the same error. The reason for this is, that for such a high noise level, the stationary subspace is hard to identify and the solutions found by the algorithms are more or less random.

3.4 Influence of the noise level

In the second experiment, we investigated the accuracy of the three different SSA algorithms in relation to the noise level. We fixed the total number of dimensions to $D = 10$ and the number of stationary sources to $d = 8$ and increased the noise level from 10^{-5} to $10^{0.2}$. The results can be seen in Figure 3.2.

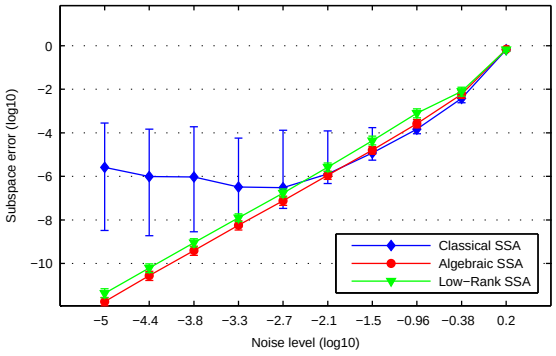

Figure 3.2: Median error of the three SSA algorithms, when the noise level is increased. The total number of dimensions is fixed to $D = 10$ and the number of stationary sources is $d = 8$. The error bars extend from the 25% to the 75% quantile estimated over 500 random realizations of the data set.

The error of the algebraic algorithm and the low-rank algorithm is nearly the same again. Also, it is interesting to see that in the log-log scale the noise-error dependence is close to a straight line, which means that the error grows like a power function when the noise level is linearly increased. For noise levels $\geq 10^{-2.7}$ all algorithms perform nearly equally well. An explanation for this would be, that for noise levels $\geq 10^{-2.7}$ the true space of stationary projections

S is so hard to identify, that there is a minimum error baseline, which can be only surpassed by chance.

3.5 Influence of the total number of dimensions

In the third experiment, we investigated the influence of the total number of dimensions to the accuracy of the algorithms, when the number of stationary sources d is only one smaller than the total number of dimensions D. We increased the total number of dimensions from $D = 3$ to $D = 12$ and set the number of stationary sources to $d = D - 1$ at a constant noise level of $10^{-2.7}$. The results can be seen in Figure 3.3.

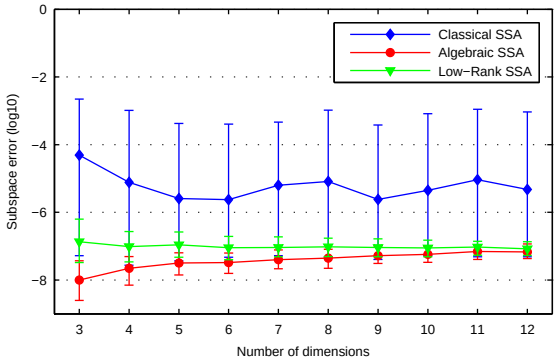

Figure 3.3: Median error of the three SSA algorithms, when the total number of dimensions D is increased and the number of stationary sources is fixed to $d = D - 1$. The error bars extend from the 25% to the 75% quantile estimated over 500 random realizations of the data set.

It is visible, that the total number of dimensions has only little influcence on the error which the algorithms make. The accuracy of the algebraic algorithm decreases with increasing number of dimensions D, which is what we expected, since the number of stationary dimensions $d = D - 1$ increases as well and the algebraic algorithm calculates the space of stationary projections S directly. Thus, with increasing dimension D, S becomes harder to estimate.

At first, it is surprising that the accuracy of Low-Rank SSA stays constant with increasing D. The reason for this is, that Low-Rank SSA calculates $S^\perp$ instead of S, whose dimension is $\dim S^\perp = D - \dim S = D - (D-1) = 1$. Hence, the dimensionality of the space which Low-Rank SSA estimates does not grow at all.

The error of the classical algorithm has such a high variance, that it is not possible to see if its accuracy decreases. Nevertheless, its accuracy is outperformed by both other algorithms again.

3.6 Runtime Experiment

In the experiments in Section 3.3, 3.4 and 3.5, Low-Rank SSA was shown to be nearly as accurate as Algebraic SSA, while those both algorithms outperform

the classical algorithm. But the advantage of our new algorithm is not primarily its accuracy, but its runtime. SSA is computationally expensive to solve, especially if the number of dimensions gets large. While the classical algorithm is feasible up to dimensions $D \approx 100$ on today's normal computers, the algebraic algorithm becomes too slow if the total number of dimensions exceeds $D \approx 30$. This is problematic, since in real-world applications, one has often very high-dimensional data sets, for example in Functional Magnetic Resonance Imaging (fMRI) or in text classification problems.

We expect that Low-Rank SSA is much faster than the algebraic and classical algorithm and therefore, if the requirement $d > \frac{D}{2}$ is fulfilled, makes SSA solveable for very high dimensions D.

The following runtime experiment was run on an Intel®Core ™2 Quad CPU Q6600 with 2.40 Ghz. The setting was the following: we increased the total number of dimensions from $D = 3$ to $D = 12$ and set the number of stationary sources to $d = \lfloor \frac{D}{2} \rfloor + 1$. The noise level was fixed to $10^{-2.7}$, where all algorithms performed nearly equally well in the second experiment. The results can be seen in Figure 3.4.

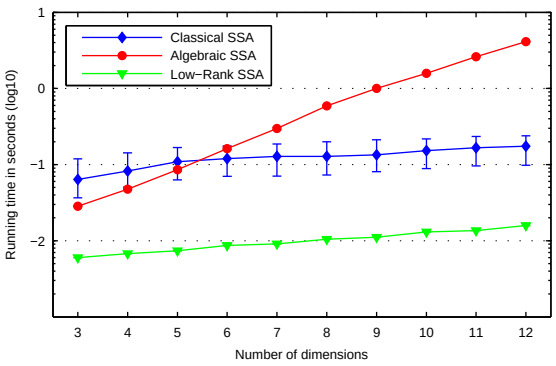

Figure 3.4: Median runtime of the three SSA algorithms, when the total number of dimensions D is increased and the number of stationary sources is set to $d = \lfloor \frac{D}{2} \rfloor + 1$. The error bars extend from the 25% to the 75% quantile estimated over 500 random realizations of the data set.

The resulting figure confirms what we expected: even for a small number of dimensions, the low-rank algorithm is significantly (approx. 10 times) faster than both other algorithms. For $D = 12$, Low-Rank SSA is approximately 160 times faster than Algebraic SSA. Since all curves are approximately straight lines in our log-lin plot, the runtime grows exponentially when the number of dimensions is linearly increased. Nevertheless, the runtime of Low-Rank SSA grows at a much smaller rate than the one of Algebraic SSA.

3.7 Performance for high-dimensional data sets

In the fifth experiment we investigated the performance of our new algorithm for high-dimensional data sets up to $D = 100$ and compared it to the classical algorithm. We set the number of stationary sources to $d = \frac{3}{4}D$ and the noise

level to $10^{-2.7}$. The results for the runtime can be seen in Figure 3.5.

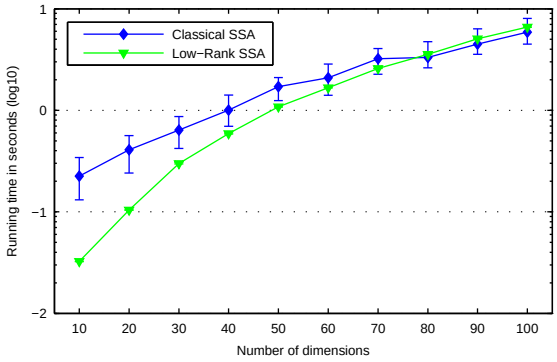

Figure 3.5: Median runtime of Low-Rank SSA and classical SSA, when the total number of dimensions D is increased from $D = 10$ to $D = 100$ and the number of stationary sources is set to $d = \frac{3}{4}D$. The error bars extend from the 25% to the 75% quantile estimated over 200 random realizations of the data set.

In this experiment, Low-Rank SSA was faster than classical SSA for dimensionality up to $D = 70$. For higher dimensions, both algorithms had nearly the same runtime. The most time-consuming operations in the Low-Rank SSA algorithm are the singular value decompositions, which becomes important when the number of dimensions gets large. This is the reason, why Low-Rank SSA is caught up by the classical algorithm at some point.

Apart from the runtime, we also measured the accuracy of both algorithms. The results can be seen in Figure 3.6. The accuracy of Low-Rank SSA is very

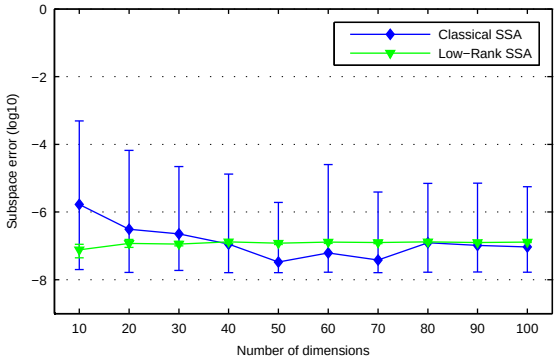

Figure 3.6: Median error of Low-Rank SSA and classical SSA, when the total number of dimensions D is increased from $D = 10$ to $D = 100$ and the number of stationary sources is set to $d = \frac{3}{4}D$. The error bars extend from the 25% to the 75% quantile estimated over 200 random realizations of the data set.

constant, while the accuracy of classical SSA has a high variance. However, the median performance of both algorithms is more or less equal for the noise level we have chosen.

3.8 Accuracy of the stationary dimension estimation

In the sixth experiment we investigated the accuracy of our stationary dimension estimation algorithm in dependence of noise. We fixed the total number of dimensions to $D = 20$ and drew the number of stationary sources d randomly from $\{11, \ldots, 19\}$. Then we calculated the difference between the estimated number of stationary sources $\hat{d}$ and the true number of stationary sources d, i.e. $\Delta(\hat{d}, d) = \hat{d} - d$. The difference is positive if the estimated number of stationary sources is larger than the true number of stationary sources. The tolerance level θ for the singular values is chosen equally to the noise level. The results can be seen in Figure 3.7.

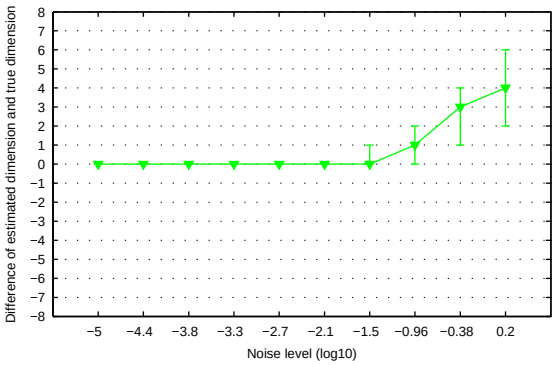

Figure 3.7: Median error when estimating the number of stationary sources. The noise level increases from left to right. On the vertical axis is the difference between the estimated number of stationary sources $\hat{d}$ and the true number of stationary sources d. The error bars extend from the 25% to the 75% quantile estimated over 500 random realizations of the data set.

It is visible that the algorithm always estimates the number of stationary sources correctly up to a noise level of $10^{-2.1}$. For higher noise, the algorithm estimates a larger number of stationary sources as present. Also, the estimate of the number of stationary sources gets unstable when the noise level is more and more increased.

Chapter 4

Conclusion

In this thesis, we have introduced a new algorithm for inverting the SSA model. This algorithm can be applied in the situation, when the number of stationary sources is larger than half of the total number of dimensions. In this case, we observed that our algebraic problem has some extra structure, which can be used to solve SSA more efficiently.

Based on this observation, we formulated the Low-Rank SSA algorithm for the exact case as well as an algorithm for estimating the number of stationary sources.

Since our data sets are finite and noisy, SSA is not solvable exactly in real applications. Therefore, we derived an approximate formulation of Low-Rank SSA with methods from approximate algebra.

In the last chapter, we investigated the performance of our new SSA algorithm with respect to accuracy and runtime and compared it with two other SSA algorithms, namely the classical and algebraic one. It turned out, that Low-Rank SSA is indeed much faster than the other both algorithms, while having an accuracy which is comparable with that of Algebraic SSA.

4.1 Discussion

Our new Low-Rank SSA algorithm has advantages over the former algorithms. In particular, it is nearly as accurate as algebraic SSA, which is the most accurate SSA algorithm yet. At the same time, Low-Rank SSA is significantly faster than algebraic SSA, and also faster than classical SSA, if the number of dimensions is not too large. But in contrast to classical SSA, the results of our new algorithm are stable and accurate.

But our new algorithm has also downsides. The first one is obvious: if the number of stationary sources is smaller than half of the total number of dimensions, Low-Rank SSA is not applicable. The second disadvantage is, that Low-Rank SSA regards only the covariances. It is the same problem as in algebraic SSA: we would be able to incorporate other moments, since our whole theory can be built-up in a more general way using tensors and higher order cumulants, see [KvBM$^+$12a] and [KvBM$^+$12b]. However, we have no justification how to weight the different moments at the current time. Moreover, the way in which we approximate the true solution does not have a probabilistic

interpretation — it is unclear how the least squares approximation using the singular value decomposition corresponds to a noise model on the data.

4.2 Future Work

There are many things which have to be investigated in future works. For example, one interesting question would be whether the Low-Rank SSA algorithm can be modified in a way that it is applicable if the number of stationary sources is small, i.e. if $d \leq \frac{D}{2}$. Maybe it is possible to add artificial stationary sources and to increase the total number of dimensions simultaneously, so that we can reduce the problem to the case $d > \frac{D}{2}$.

Another interesting research question would be, how to weight different moments, if we incorporate more than only the covariances. This is something which is unclear for both Low-Rank SSA and Algebraic SSA.

In the approximate version of Low-Rank SSA, we used the singular value decomposition to calculate approximate spans. Those calculations consume a lot of time — maybe it is possible to replace the SVD by an approximation method which is faster.

Also, it would be interesting to apply Low-Rank SSA on real-world data. For example, there are many high-dimensional data sets from the neurosciences, where assuming a large number of stationary sources may be justified.

Bibliography

[BvBMM12] 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.

[CGTW95] Robert M. Corless, Patrizia M. Gianni, Barry M. Trager, and Steven M. Watt. The singular value decomposition for polynomial systems. *Proc. ISSAC '95*, pages 195–207, 1995.

[FH94] Ralf Fröberg and Joachim Hollman. Hilbert series for ideals generated by generic forms. *Journal of Symbolic Computation*, 17(2):149 – 157, 1994.

[HKPP09] 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. In Memoriam Karin Gatermann.

[HKW+12] 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(0):7 – 20, 2012.

[HKWvB10] 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.

[Jor75] C. Jordan. Essai sur la Géometrie à n dimensions. *Bull. Soc. Math. France*, 3:103–174, 1875.

[KvBM+12a] 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, 2012.

[KvBM+12b] 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.

[Mei11] Frank C. Meinecke. *Synchronized? Identifying interactions from superimposed signals*. PhD thesis, TU Berlin, 2011.

[MvBKM09] 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.

[MvBM+11] Jan S. Müller, Paul von Bünau, Frank C. Meinecke, Frank J. Király, and Klaus-Robert Müller. The stationary subspace analysis toolbox. *Journal of Machine Learning Research*, 12:3065–3069, 2011.

[Par10] Keith Pardue. Generic sequences of polynomials. *Journal of Algebra*, 324(4):579 – 590, 2010.

[Pri83] M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, 1983.

[vBMKM09] Paul von Bünau, Frank C. Meinecke, Franz J. Király, and Klaus-Robert Müller. Finding stationary subspaces in multivariate time series. *Phys. Rev. Lett.*, 103(21):214101, Nov 2009.

[vBMM+09] Paul von Bünau, Frank C. Meinecke, Jan S. Müller, Steven Lemm, and Klaus-Robert Müller. Boosting high-dimensional change point detection with stationary subspace analysis. In *Workshop on Temporal Segmentation at NIPS 2009*. 2009.

[vBMSM10] 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.

[YE36] G. Young and C. Eckart. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.