DISSERTATION

SIGNAL FRACTION ANALYSIS FOR SUBSPACE PROCESSING OF HIGH DIMENSIONAL DATA

Submitted by

Fatemeh Emdad

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2007

COLORADO STATE UNIVERSITY

July 31, 2007

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY FATEMEH EMDAD ENTITLED SIGNAL FRACTION ANALYSIS FOR SUBSPACE PROCESSING OF HIGH DIMENSIONAL DATA BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Wayne Schubert
Committee Member

Committee Member

Committee Member

Adviser

Department Head

ABSTRACT OF DISSERTATION

SIGNAL FRACTION ANALYSIS FOR SUBSPACE PROCESSING
OF HIGH DIMENSIONAL DATA

A general tool for computing subspaces that decomposes data into potentially useful features is proposed. The technique is called Signal Fraction Analysis (SFA). The row-energy and column-energy optimization problems for signal-to-signal ratios are investigated. A generalized singular value problem is presented. This setting is distinguished from the Singular Value Decomposition (SVD).

Preprocessing mappings of the data is used in situations where domain specific knowledge is available as a guide. We suggest an optimization problem where these mapping functions may be adapted using a problem dependent objective function. These ideas are illustrated using Wavelet and Fourier filters applied to EEG data. A self-contained description of the motivating maximum noise fraction method is included and a procedure for estimating the covariance matrix of the noise is described.

We extend SFA by introducing novel constraints and propose two new generalized SVD type problems for computing subspace representations. A connection between SFA and Canonical Correlation Analysis is maintained. We implement and investigate a nonlinear extension to SFA based on a kernel method, i.e., Kernel SFA. Moreover, a second algorithm that uses noise adjustment in the data domain prior to kernelization is suggested. We include a detailed derivation of the methodology using kernel principal component analysis as a prototype. These methods are compared using toy examples and the benefits of KSFA are illustrated.

This work establishes the potential of a SFA beamforming technique via its merger with a wide band MC-CDMA system. The details of non-overlapping window adaptive realization of SFA are introduced. We discuss the relationship between the SFA and DOA estimation via MUSIC. A novel structure for wide band MC-CDMA systems that utilizes the benefits of path diversity (inherent in direct sequence CDMA) and frequency diversity (inherent in MC-CDMA systems) is introduced. Simulations were performed to study the impact of noise perturbations on the performance of SFA. Simulations confirm that SFA enhances the performance and separability of interfering users.

KSFA is applied to the classification of EEG data arising in the Brain Computer Interface Problem. We use Fourier and Wavelet filters to generate signal fractions as well as differencing methods.

Fatemeh Emdad
Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523
Fall 2007

DEDICATION

To my Parents Shamsi Arefadib and Ahmad Emdad,

who helped me find my path to success,

to my husband Dr. Seyed A. Zekavat,

who stayed with me strong in my whole life,

and to my daughters Maryam, Melica, and Mona,

who supported me with love and patience to be who I am now.

I love you all from the bottom of my heart.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my advisor Professor Michael Kirby for his support. It has been my privilege to work under his supervision.

I would like to thank Professors Rick Miranda, Gerhard Dangelmayr, and Wayne H. Schubert for serving on my committee and reviewing my dissertation. I would like to thank Professor Charles W. Anderson for valuable comments and feedback.

I would like to express my high appreciation towards my parents, my brother, my husband, and my daughters. I am truly grateful to my parents Shamsi Arefadib and Ahmad Emdad and my brother Dr. Mohammad Reza Emdad for their love, support, patience, encouragement, kindness and understanding. I can not appreciate them in words or any other way. I have marveled at the courage, strength and resolution demonstrated by my special love my husband Dr. Seyed A. Zekavat. Special thanks to my daughters Maryam, Melica, and Mona for their support, love, patience and understanding. I have to thank Mr. Arta Jamshidi for his support throughout my research process and his kindness. I owe them all that I am and all that I have accomplished. Without their support I would not be able to finish the work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

SUBSPACE SIGNAL PROCESSING

1.1 Introduction

Researchers in a variety of fields collect measurements or observe data, and apply range of techniques to describe, classify, analyze and draw conclusions from the data. Selecting appropriate techniques and understanding their advantages and disadvantages is one of the most important steps in data analysis. In particular, large data sets provide their own distinct challenges given that many of the general techniques for small data sets do not scale to larger problems and often are prohibitively expensive from a computational perspective. The analysis of such data sets has become increasingly important given the dramatic price reduction in mass storage devices that has made the existence of such large data sets commonplace: at the time of writing of this dissertation a 6GB DVD costs about 1 dollar and a 10 Terabyte mass storage device sells for under $10,000.

The ability to collect and analyze high resolution information, both in space and time, provides unique opportunities for the discovery of knowledge. For example, electroencephalographic, or EEG, signals provide a unique, if somewhat noisy, window into the world of brain science and have the potential to provide insight into the relationship between, e.g., patterns of electrical activity and cognitive development. EEG data sets typically involve 32-512 electrodes capturing data at 256Hz-1024Hz and hence provide a formidable challenge for interpretation. Given the fact it is relatively inexpensive, a

typical EEG apparatus may be assembled for under $5,000, the application of EEG is increasingly widespread. The rapid emergence of the field of quantitative EEG (QEEG), see, e.g., [30] reflects a growing need for algorithm development in this area. Additionally, technological developments in the direction of hybrid EEG/MEG as well as EEG/FMRI systems indicate the need for algorithms that perform well on larger data sets.

In addition to scientific discovery, methodologies for handling large amounts of information have the potential to lead to significant technological advances, for example, in the area of mobile communication. In this setting, receivers are confronted with the task of distinguishing a signal of interest from a large number of interfering signals. The mathematics of the signal separation problem are introduced in Chapter 2 and new approaches for this problem are presented and applied in the remaining chapters of this dissertation.

The inherent structure in data may be classified into three main types: spatial, temporal and spatio-temporal. As we shall see in what follows, the distinction between these types of data is important in designing an effective analysis technique. For example, spatial data might consist of a set of digital images of static patterns such as mushrooms. The data classification problem would be to determine if a particular mushroom were poisonous based on a digital image. Temporal data could consist of the value of a stock as a function of time, or the temperature recorded at a weather station. Spatio-temporal data consists of temporally evolving data whose values are known at several, generally nearby, points in space. The electrical activity measured on the scalp by an array of 512 electrodes could be an example for the spatio-temporal data. Each electrode captures temporally evolving information over time, while at a fixed time, the pattern captured by the ensemble of electrodes is spatial. In Chapter 4 we shall see that the formulation of the mathematical subspace approach is directly influenced by the spatial and/or temporal structure of the data.

In general, when large data sets are collected for analysis, data reduction becomes one of the most important objectives for reducing the computational time and cost. In the data reduction problem a large data set is compressed to a smaller set carrying almost all the essential information hidden in the big set. Naturally, the term essential is problem dependent. Furthermore, observed data is often a mixture of the desired data and noise or other undesired signal (e.g., measurement noise or white noise in wireless systems). Isolating the information of interest by removing either noise or an undesired signal from the data is a key step to providing more accurate analysis, conclusions, and indeed performs a better decision making [48, 22]. We are particularly interested in this data extraction problem in the context of geometric pattern analysis [51], beam forming [101], and signal separation [45].

In summary, the economics of mass storage devices coupled with the potential for scientific and technological advances have led to a dramatic increase in interest in new approaches for handling large and potentially noisy data sets. The geometric nature of the data will play a role in the formulation of the subspace methodology.

1.2 The Subspace Approach

The subspace approach to signal and image processing consists of decomposing the data into parts that reveal the essential information, or structure, of interest. It is an implicit assumption in this approach that standard basis in which the data is collected, e.g., an array of pixel values for an image, is generally non-optimal. Consider the case where one is interested in analyzing a collection of signals with common characteristics, i.e., a *family of patterns*. In this situation one seeks to exploit the information across the family to permit the extraction of novel features within each element of the family.

Often observed data arises from the combination of two (or more) sources of interest. For example, we may observe the data matrix X that is actually the superposition[1] of two signals, i.e.,

$$X = P + Q.$$

Alternatively, there is the related problem of observing a set of signals that arise from the linear mixing of the original signals S, i.e.,

$$X = SA.$$

In this situation the challenge is to recover the signal S without knowing the (square) mixing matrix A, a problem also known as Blind Signal Separation (BSS). In these subspace problems, we are concerned with extracting the information associated with the underlying individual signals that has been masked by the interference or the mixing present in the observed signals.

Yet another situation of significant interest arises when we seek to quantify similarities and differences between two distinct sets of data. For this problem standard subspace methods arising from the consideration of the data as a single set, such as the singular value decomposition, fail to characterize, or exploit, the distinction between the data sets. In this setting it is desirable to construct a single basis for the two data that simultaneously describes each and their differences. In particular, the first few basis vectors might describe features of P absent in Q while the tail of the basis does the reverse. Intermediate basis vectors represent both data sets together.

1.3 Signal Fraction Analysis

The central focus of this dissertation is on a method we refer to as signal fraction analysis (SFA). As described in Chapter 4, SFA is a general framework for the construc-

[1]Note that this includes the point-wise multiplicative case as well using the identity $\log X_{ii} = \log P_{ii} + \log Q_{ii}$.

tion of an orthonormal basis that maximizes a signal-to-signal ratio in general terms. Historically, SFA has its origins in a method known as maximum noise fraction (MNF) that was introduced by Green [35], and was reformulated by Lee [58]. MNF was originally developed to de-noise multi-spectral satellite images (see e.g., [92, 14]). A related approach, referred to as *oriented signal-to-signal ratio*, also proposes to separate signals by optimizing an energy criterion [12]. A similar technique referred to as *Common Spatial Patterns*, has been proposed for separating patterns in data (see e.g., [27, 82]).

Signal fraction analysis provides a unifying perspective and extension to these methods as described in Chapter 4. This method has been applied to the problem of time-series analysis of multivariate time-series (see, e.g., [51, 52, 6, 5]). Furthermore, SFA has been shown to be an effective means to solve the blind signal separation problems [42, 41]. Moreover, SFA is a powerful scheme for data separation in wireless communications [101, 29]. SFA has also proven very useful for the analysis of EEG data in the context of task classification in the brain computer interface (BCI) problem [52], and artifact removal [53].

Details of the SFA procedure will be developed in this dissertation. Here we remark that the signal fraction idea we propose is based on the general notion that a space may be split using any empirical or analytical transformation of the data in the optimization problem. Knowledge of the problems objective determines the manner in which this quotient is defined. Thus, for example, one can employ a signal to noise ratio as in SFA, an empirical signal to signal ratio, or a transformed signal to signal ratio.

1.4 A Brief Overview of Subspace Methods

Here we provide a brief introduction to several well-known subspace methods while deferring the mathematical details of these techniques to Chapter 2.

Some of the most popular data analysis techniques for data compression and feature extraction are Independent Component Analysis (ICA) [43, 13], Principal Component

Analysis (PCA) [48, 22, 51, 101, 33, 106, 71, 95], Canonical Correlation Analysis (CCA) [109], Common Spatial Patterns (CSP) [27, 82], and Maximum Noise Fraction (MNF) [34, 92]. Each of these methods has its own particular advantages and disadvantages.

ICA is a signal processing technique for data sets expressed as a linear transform of statistically independent non-Gaussian components. ICA was introduced in early 1980s by J. Herault, C. Jutten, and B. Ans [39, 40, 8]. In 1989, the first international workshop on higher spectral analysis was organized and Cardoso [21] presented papers on developing ICA. Applying the ICA method serves to separate these linearly mixed components via an optimization problem that maximizes the non-Gaussianity of the linear transform, e.g., in terms of kurtosis [45], entropy [45, 25], etc. As a method for signal separation, ICA has applications in wireless communications, the cocktail-party problem, feature extraction, economics, as well as general signal and image processing problems [45]. ICA is optimized for sources that are non-Gaussian, independent and linearly mixed; for example, it can not completely recover data that is mixed with Gaussian noise. In the literature some work has been done to generalize the basic linear ICA to its nonlinear form, see e.g., [93, 59, 60].

In contrast to ICA, PCA is a signal processing technique that produces a change of basis in which the data is uncorrelated rather than statistically independent. In other words, PCA is a technique to find the main directions over which a cloud of data is stretched. These directions represent the main information (in terms of maximum statistical variance) available in the data. PCA requires the solution of an eigenvalue problem to transform the original correlated data set into a number of uncorrelated linear combinations of the original data set (eigenvectors) called principal components [33, 106]. The associated eigenvalues represent the amount of variance provided by the respective principal component (eigenvector) [48]. Data reduction in PCA is realized via removing principal components (eigenvectors) associated with low eigenvalues.

PCA is a common approach in reducing the dimension of the data; however, in many signal processing applications where the data (signal) is the superposition of signal and noise, the optimal PCA basis is not appropriate for extracting signal from noise or for separating two signals. In other words, PCA does not always generate components of decreasing signal quality, and some components with small eigenvalues may contain relevant information rather than just having noise [95]. Indeed in PCA one principal component related to a bigger eigenvalue may contain less useful information than another principal component associated to a smaller eigenvalue. We call this issue PCA *eigenvalue ambiguity*. Moreover, PCA is sensitive to linear scaling, that is, by changing the scale of the data, eigenvalues would change [49].

When the data are irregularly distributed in space, e.g., in the presence of nonlinear correlations, the data variance often fails to be an adequate measure of relative importance. A nonlinear extension to PCA has been proposed by [86] for such data. Subsequently kernel based methods and nonlinear extensions to kernelizable algorithms have attracted the attention of many researchers. Kernel PCA (KPCA) is generated via mapping the input space (original data set) to feature space (higher dimension space than the input space) using kernel functions. Which means, first the input data is transformed nonlinearly to the new variables and then usual linear PCA will be applied. KPCA is an algorithm that performs nonlinear PCA by carrying out linear PCA after a nonlinear transformation. KPCA allows the extraction of nonlinear features carrying information about the structure of the data.

1.5 Organization of the Dissertation

The organization of this Ph.D. dissertation is as follows: Chapter 1 introduces the nature of the data to be investigated and provides a brief introduction to signal fraction analysis. In Chapter 2 we provide an overview of commonly used approaches to subspace analysis. In Chapter 3, we review Generalized Singular Value Decomposition (GSVD),

and the related CS-Decomposition (CSD). In Chapter 4, we provide a detailed introduction to Signal Fraction Analysis (SFA). We emphasize a general framework that permits the connection of this approach with several other subspace techniques in the literature. In Chapter 5, we review notion of a kernel method including Kernelize Principal Component Analysis (KPCA), and propose two different ways to kernelize SFA. We generate different toy examples to compare the performance of KPCA and Kernel SFA (KSFA). In Chapter 6, we present an application of SFA in signal separation and communication, and we merge SFA and MC-CDMA and form the beam pattern for several degrees of dispersion. Finally in Chapter 7, we work with the real data sets from the EEG brain signals to classify different tasks applying KPCA and KSFA via k-nearest neighbor method, and we compare the performance in classification for KPCA and KSFA.

Chapter 2

MATHEMATICAL BACKGROUND OF SUBSPACE METHODOLOGIES

In this chapter, we provide mathematical background techniques related to signal fraction analysis including the Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Canonical Correlation Analysis (CCA). We also briefly introduce the multiple signal classification (MUSIC) algorithm, and the method of Common Spatial Patterns (CSP).

2.1 The Singular Value Decomposition (SVD)

In this section, we present the well-known SVD, and we show its relationship to the eigenvalue/eigenvector decomposition and we define and compare big and small problems. Then we briefly talk about the range and the null space of a matrix based on its SVD and the geometry of the subspaces and we introduce the notion of constructing projections via optimization criteria.

Singular value decomposition is one of the most traditional and common subspace approaches that obtains appropriate subspace from a noisy data set and it is one of the best decompositions that could be used for a rectangular matrix representation. SVD technique is based on minimizing the least square error, it provides the best low rank approximation of a matrix, and it is sensitive to outliers.

Every matrix (real or complex) has the singular value decomposition [94]. Let A be any $m \times n$ matrix (A can be a complex matrix as well [94]), $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^m$. Here

for simplicity we assume A is real and we use T to denote the transpose of a matrix, simply if A is a complex matrix one should use the Hermitian sign H instead of T.

The SVD optimization problem can be defined as

$$\|A\|_{(m,n)} = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \tag{2.1}$$

where $\|\cdot\|$ defines the 2-norm of a vector [24] which is defined as

$$\|x\| = \sqrt{x^T x} = (\sum_{i=1}^{n} |x_i|^2)^{1/2} \tag{2.2}$$

and $\|A\|_{(m,n)}$ denotes the induced matrix norm which is the smallest number C such that the inequality (2.3) holds for all vectors $x \in \mathbf{R}^n$ [94],

$$\|Ax\| \leq C\|x\| \tag{2.3}$$

Solving Equation (2.1) provides the right singular vectors of A and solving Equation (2.4) leads to the solutions called the left singular vectors

$$\|A\|_{(n,m)} = \max_{y \neq 0} \frac{\|y^T A\|}{\|y\|} \tag{2.4}$$

In the SVD we work with a single matrix (e.g., A) and we use

$$A = U\Sigma V^T \tag{2.5}$$

where

$$\Sigma = \begin{bmatrix} S_A & 0_1 \\ 0_2 & 0_3 \end{bmatrix} \tag{2.6}$$

as the SVD of matrix $A \in \mathbf{R}^{m \times n}$. Here matrices $U \in \mathbf{R}^{m \times m}$ and $V \in \mathbf{R}^{n \times n}$ are orthonormal matrices of left and right singular vectors respectively and $\Sigma \in \mathbf{R}^{m \times n}$ is a diagonal matrix of singular values in descending order. If we suppose that $rank(A) = r$ then the diagonal entries of matrix Σ has r nonzero, and nonnegative elements. SVD provides the low rank approximation of a matrix. For a general case, when we don't know if matrix A is a long matrix or a wide matrix, we can use Equation (2.6) for the singular

value matrix where $S_A = diag(\sigma_i)$, $\sigma_i > 0$, $i = 1, \ldots, r$, $S_A \in \mathbf{R}^{r \times r}$, $0_1 \in \mathbf{R}^{r \times (n-r)}$, $0_2 \in \mathbf{R}^{(m-r) \times r}$, and $0_3 \in \mathbf{R}^{(m-r) \times (n-r)}$, and ($0_1, 0_2$, and 0_3 are zero matrices) [73].

The Singular value decomposition and the eigenvalue/eigenvector decompositions of A^TA and AA^T have a very close relationship. The singular values of matrix A are the square roots of the eigenvalues of the matrix A^TA and AA^T. The right singular vectors $v_k \in V$ are the same as the eigenvectors of matrix A^TA and the left singular vectors $u_k \in U$ are the same as the eigenvectors of matrix AA^T.

Using

$$U^TAV = \Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n) \tag{2.7}$$

we see that the eigenvalue/eigenvector decomposition for $A^TA \in \mathbf{R}^{n \times n}$ can be represented by

$$A^TA = V\Sigma^2V^T \tag{2.8}$$

or

$$A^TAV = V\Sigma^2 \tag{2.9}$$

The existence of the SVD has been proved in [94, 51]. In [51] a constructive proof for the existence of right and left singular vectors is provided. Here, we should add that since A^TA and AA^T are symmetric matrices, then their eigenvectors exist. From the point that eigenvectors and singular vectors have a close relationship which has been discussed above, the singular vectors of matrix A exist as well [51].

Big vs. Small problem

In Section 2.1, we established a brief foundation for understanding the relationship between SVD and eigenvalue eigenvector problem. In this section we introduce big and small problem. In this dissertation we call Equation (2.9) as small problem based on the

size of matrix A^TA which is $n \times n$. Now we find eigenvalue eigenvector decomposition for $AA^T \in R^{m \times m}$, and since we assumed ($m \geq n$) we refer to

$$AA^TU = U\Sigma^2 \tag{2.10}$$

as the big problem.

Range and Null spaces

In terms of the range and null space of matrix A, we may write these spaces as

$$R(A) = \{y : y = Ax = U\Sigma V^T x = U_1\phi\} = R(U_1) \tag{2.11}$$

and

$$N(A) = \{x : y = Ax = U\Sigma V^T x = 0\} = \{x : V_1^T x = 0\} = N(V_1^T) \tag{2.12}$$

Here, we have partitioned matrices U and V like $U = [U_1, U_2]$ and $V = [V_1, V_2]$. Assume that the $rank(A) = r \leq n$, then we partition U and V such that U_1 is $m \times r$ and V_1 is $r \times n$, and $\phi = S_A V_1^T x$ (see Equation (2.6) for S_A). Therefore, the dimension of the range of A is r and the dimension of the null space of A is $m - r$.

2.2 Principal Component Analysis (PCA)

Intimately related to the SVD matrix decomposition described above, Principal component analysis (PCA) is a powerful classic technique for statistical data analysis, feature extraction and data compression. Pearson was the first to introduce PCA technique [81]. Tufts and Kumaresan developed this technique in [96]. Then, Vaccaro, Tufts, and Boudreaux-Bartels reviewed it in [97]. PCA transforms the original correlated data set into a number of uncorrelated data called principal components [106, 48] which contains almost all the information that exists in the original data. These components are uncorrelated linear combinations of the original data set. In fact we can say PCA linearly de-correlates the data.

PCA technique is realized via generating the eigenvalues and respective eigenvectors of the covariance matrix of the original correlated data set. The eigenvectors represent a set of uncorrelated variables that determines the directions of maximum variability in the data which are called principle components, and the eigenvalues show the amount of information provided by the respective principal component [48, 44]. Data reduction is realized via removing principal components (eigenvectors) associated to low eigenvalues. Sometimes, a meaningful specification can be defined for the new (reduced dimension) set of the data [22].

Given a set of points X, PCA finds the best line that approximates it. PCA and SVD are two important approaches in graphics, statistics, computer vision and much more. Principal components are uncorrelated linear combinations of the random variables whose variances are as large as possible. Then, if $X_1, X_2, \ldots, X_n$ are n random variables of p dimension and $X^T = [X_1, X_2, \ldots X_n] \in \mathbf{R}^{p \times n}$, we can consider the linear combinations of them as [48]:

$$Y_i = X^T l_i^T = l_{1i}X_1 + l_{2i}X_2 + \ldots + l_{ni}X_n \quad i = 1, \ldots, n \tag{2.13}$$

If we denote the covariance matrix of the random variables by $\Sigma = E(X^T X)$, then

$$var(Y_i) = l_i^T \Sigma l_i \quad i = 1, \ldots, n \tag{2.14}$$

and

$$cov(Y_i, Y_k) = l_i^T \Sigma l_k \quad k, i = 1, \ldots, n \tag{2.15}$$

Principal components are those uncorrelated linear combinations with maximum variance, hence maximizing Equation (2.14) subject to unit length for the coefficients (l_i) leads to have the following principal components (for the proof please see [48])

$$Y_i = e_i^T X^T = e_{li}X_1 + e_{2i}X_2 + \ldots e_{ni}X_n \quad i = 1, \ldots, n \tag{2.16}$$

with its variance equal

$$var(Y_i) = e_i^T \Sigma e_i = \lambda_i \quad i = 1, \ldots, n \tag{2.17}$$

where, (λ_i, e_i), $i = 1, \ldots, n$ is the eigenvalue eigenvector pair of $\Sigma = X^T X$. As it is clear, the principal components are uncorrelated and have variances equal to the eigenvalues of $\Sigma = X^T X$.

Therefore, we can summarize the PCA process as follows: first we find the covariance matrix of the data, then we compute the eigenvalue/eigenvector decomposition of it and we order the eigenvalues in a descending order and based on that we order the eigenvectors associated to them to find the principal components. PCA finds the largest eigenvector of the signal, and the projections onto these eigenvectors will be used as a new representation of the signal and it is important to know that PCA makes no distinction between dependent and independent signals.

2.3 Independent Component Analysis (ICA)

ICA is based on the assumption of having linearly independent non-Gaussian signals [39, 40, 8, 21, 45]. It is one of the best methods for blind source separation (BSS). Blind source separation separates original signals (sources) that have been mixed via an unknown (blind) mixing matrix.

Let $S^T = [s_1, s_2, \ldots, s_n]$, be the source vector and $X^T = [x_1, x_2, \ldots, x_n]$ be the source mixture random vector, with mixing matrix A then using the vector- matrix notation the mixing model is

$$X = AS \tag{2.18}$$

In this model just X is known; S can not directly be observed and the mixing matrix A is also unknown. Elements of S are statistically independent and non-Gaussian and for simplicity we suppose A is a square matrix.

Let us denote the linear combination of elements of X by y as:

$$y = W^T X \tag{2.19}$$

Therefore, from Equation (2.18) we have:

$$y = W^T X = W^T AS = Z^T S \tag{2.20}$$

where,

$$W^T A = Z^T \tag{2.21}$$

Here, W needs to be determined. From Equation (2.20) we see that y is a linear combination of the original independent non-Gaussian signals. Using the central limit theorem and Equation (2.20), we know that the distribution of the sum of more than two independent variables is closer to Gaussian than the original signals. Hence, we are looking for a measure of non-Gaussianity to apply on y to find the independent sources.

One of the classical measures of non-Gaussianity is Kurtosis. The Kurtosis of y is defined as:

$$kurt(y) = E(y^4) - 3[E(y^2)]^2 \tag{2.22}$$

Here, $E(.)$ denotes the expectation. Assuming y with unit variance, the right hand side simplifies to

$$kurt(y) = E(y^4) - 3 \tag{2.23}$$

For Gaussian random variables (bell shape) Kurtosis equals zero, for Sub-Gaussian random variables (flatter than Gaussian) Kurtosis is negative, and for Sup-Gaussian random variables (sharper peak, longer tail) Kurtosis is positive.

Applying the fixed point algorithm to optimize the Kurtosis leads to: (for more detail see [45, 44])

$$W_{new} \propto E(X(W_{old}^T X)^3) - 3W_{old} \tag{2.24}$$

Here, we need to find the estimation of the expectation in Equation (2.24) using sample mean.

2.4 Canonical Correlation Analysis (CCA)

There are many ways to reduce the dimensionality and perform dimensionality reduction. CCA is one of those methods and the objective of CCA is mostly two things: 1) dimension reduction, and 2) extracting similarity. It reduces the dimension of the data, because, the dimension of the two bases is equal to or less than the smallest dimension of the two data sets. It extracts the similarity since it maximizes the correlation between the two data sets and correlation is a measure for similarity [109].

Suppose we have two sets of data, X and N. CCA finds two linear transformations, that maximize the correlation of X and N in the new coordinates. Let us call two sets of data $X_{n\times p}$ and $N_{n\times p}$, then consider the linear projections $y_1 = Xa$ and $y_2 = Nb$. The correlation between y_1 and y_2 is given by

$$\rho(y_1, y_2) = \frac{\mathrm{E}y_1 y_2}{\sqrt{\mathrm{E}{y_1}^2 \mathrm{E}{y_2}^2}} = \frac{a^T X^T N b}{\sqrt{a^T X^T X a}\sqrt{b^T N^T N b}}.$$

Here, a and b are two basis vectors. The canonical correlations can be extracted by maximizing $\rho(y_1, y_2)$

$$\max_{a,b\neq 0} \frac{a^T X^T N b}{\sqrt{a^T X^T X a}\sqrt{b^T N^T N b}}. \tag{2.25}$$

Taking the derivative of the correlation function with respect to a and b and setting them to zero respectively leads to

$$X^T N b = \frac{a^T X^T N b}{a^T X^T X a} X^T X a \tag{2.26}$$

$$N^T X a = \frac{b^T N^T X a}{b^T N^T N b} N^T N b \tag{2.27}$$

If we first pre-multiply Equation (2.27) by $(N^T N)^{-1}$ and then by $X^T N$ we get:

$$X^T N (N^T N)^{-1} N^T X a = \frac{b^T N^T X a}{b^T N^T N b} \frac{a^T X^T N b}{a^T X^T X a} X^T X a \tag{2.28}$$

Based on the definition for $\rho(y_1, y_2)$ and Equation (2.28) we get

$$X^T N(N^T N)^{-1} N^T X a = (\rho(y_1, y_2))^2 X^T X a \tag{2.29}$$

If we first pre-multiply Equation (2.26) by $(X^T X)^{-1}$ and then by $N^T X$ then

$$N^T X(X^T X)^{-1} X^T N b = \frac{b^T N^T X a}{b^T N^T N b} \frac{a^T X^T N b}{a^T X^T X a} N^T N b \tag{2.30}$$

Based on the definition for $\rho(y_1, y_2)$ and Equation (2.30) we get

$$N^T X(X^T X)^{-1} X^T N b = (\rho(y_1, y_2))^2 N^T N b \tag{2.31}$$

Therefore, derivation of CCA leads to Equations (2.29) and (2.31).

2.5 Multiple Signal Classification (MUSIC)

In general, besides recovering the desired signal another parameter of interest in array signal processing extracted from the raw received data is the direction of arrival. Many methods have been proposed and analyzed for DOA finding. One of the popular methods for finding arrival directions is based on multiple signal classification (MUSIC). MUSIC algorithm is a signal subspace approach that mainly provides estimates of the number of users and the direction of arrival (DOA). Finding DOA via MUSIC algorithm leads to solve a GSVD problem (the same GSVD problem for SFA method), and to compute the maximum of a function called the DOA spectrum. In many applications since the DOA is unknown, one can apply MUSIC method to estimate those directions and then apply SFA to estimate the desired signal. Therefore, the study of the merger of SFA and MUSIC might open another window to problems that arise in array signal processing [89].

The MUSIC algorithm is one of the best methods to find the direction of arrival signals. It relies on the property that the signal subspace is orthogonal to the noise

subspace. Let the data model be

$$X(t) = A(\theta)S(t) + n(t) \tag{2.32}$$

where $S(t)$ is the original signal vector including signals from different sources at time t, $X(t)$ represents the signal output vector, $n(t)$ is the additive noise vector at time t and $A(\theta)$ is the direction of arrival matrix. If we call the $i-th$ column of $A(\theta)$ by $a(\theta_i)$ then $a\vec{\theta}_i = [1, e^{jk\cos(\theta_i)}, \ldots, e^{u(M-1)k\cos(\theta_i)}]^T$ where, M is the number of antenna elements (or receivers), $k = \frac{2\pi d}{\nu}$ with d corresponds to the distance between the antenna elements, and ν corresponds to the signal wavelength. The vector $a\vec{\theta}_i$ specifies the position of the $i-th$ source with respect to the receiver. Here, the problem can be summarized as follows: given the measurements $X(t)$, estimate the direction of the arrival vector θ.

For any vector e_i in the noise subspace, $a\vec{\theta}_i$ is orthogonal to e_i . If we denote the noise subspace by $\Re(e_i)$ we have $\Re(e_i) \perp a\vec{\theta}_i$ which is equivalent to

$$a\vec{\theta}_i^{\,T}\Re(e_i) = 0 \quad i = l, \ldots, d \tag{2.33}$$

Here d is the dimension of the noise subspace. Therefore, if we define

$$M(a) = \sum |(\theta - i)^T \Re(e_i)|^2 \tag{2.34}$$

Equation (2.34) equals zero and its reciprocal is infinite. Hence the plot of $\frac{1}{M(a)}$ should have tall peaks at direction of arrival points. This fact leads to define the MUSIC spectrum as the following:

$$P(\theta) = \frac{1}{M(a)} \tag{2.35}$$

Hence maximizing Equation (2.35) provides the peaks that show the direction of arrivals.

2.6 Common Spatial Pattern (CSP)

Another method related to SFA proposed independently is known as common spatial patterns (CSP) and addresses the special case of observing two data sets and constructing

a subspace representation for splitting them [27, 82, 103]. The CSP method is based on the joint diagonalization of two covariance matrices and was first used on EEG data set to extract abnormal components from the clinical EEG [55]. This method is based on statistical pattern recognition, in which we maximize the variance of the signal, and at the same time, we minimize the variance of the noise (or the other signal) to determine the desired signal [27, 83, 55, 54, 32, 36, 107, 82, 75, 104].

The SFA and CSP techniques are related because there is a connection between the common spatial patterns (CSP) and the generalized singular value decomposition (GSVD). They both use the signal to noise ratio (directly or indirectly) to find the desired signal. The main difference between these two methods is the requirements for the desired signal extraction. In SFA we find the signal which has the maximum signal to noise ratio and we almost look for the noise free signal, however in CSP method we look for the strongest signal with a reasonable SNR by looking at the signal and noise difference. In this case the extracted signal might not have the greatest possible signal to noise ratio but provide a good approximation of the desired signal with the minimum acceptable SNR [107].

We apply the CSP method on the data set $X \in R^{n \times p}$ where $X = S + N$ (the desired signal is called S, and undesired signal or noise is N). Here we define the CSP function as:

$$CSP = var(X) - \tilde{\alpha} var(N) \tag{2.36}$$

Here, $\tilde{\alpha}$ is the scalar weighting parameter. The CSP transformation is a linear transformation defined by

$$Y = Xa = Sa + Na \tag{2.37}$$

where, a is a $p \times 1$ vector. Here, the problem is to determine a in order to maximize the CSP for Y when satisfying the constraint on the length of a ($\|a\|^2 = 1$). The CSP for

Y is:

$$CSP = var(Sa) - \tilde{\alpha} var(Na) \tag{2.38}$$

Therefore,

$$\begin{aligned} CSP = (E(a^T S^T Sa) - E(a^T S^T)E(Sa)) - \\ \tilde{\alpha}(E(a^T N^T Na) - E(a^T N^T)E(Na)) \end{aligned} \tag{2.39}$$

Here, we assume a data set with zero mean (i.e.,$E(S) = 0$ and $E(N) = 0$). Hence since expectation is a linear operator we have $E(Sa) = E(S)a = 0$. Therefore, Equation (2.39) corresponds to

$$CSP = E(a^T S^T Sa) - \tilde{\alpha} E(a^T N^T Na) \tag{2.40}$$

From $X = S + N$ we have:

$$a^T X^T Xa = a^T S^T Sa + a^T S^T Na + a^T N^T Sa + a^T N^T Na \tag{2.41}$$

Computing $a^T S^T Sa$ from Equation (2.41) and substituting it into Equation (2.40) results in

$$\begin{aligned} CSP = E(a^T X^T Xa) - E(a^T S^T Na) \\ -E(a^T N^T Sa) - E(a^T N^T Na) \\ -\tilde{\alpha} E(a^T N^T Na) \end{aligned} \tag{2.42}$$

Assuming the signals independent from each other and from the noise, we get

$$E(a^T S^T Na) = a^T E(S^T N)a = a^T E(S^T)E(N)a = 0 \tag{2.43}$$

and Equation (2.42) simplifies to

$$CSP = E(a^T X^T Xa) - (1 + \tilde{\alpha}) E(a^T N^T Na) \tag{2.44}$$

Since we don't have access to the statistical information, we must use the estimation of the covariance matrices to compute the CSP. Assuming n large, we can estimate the statistical means in Equation (2.44) via the sample mean. In this case, Equation (2.44) corresponds to

$$CSP = a^T X^T X a - (1 + \tilde{\alpha}) a^T N^T N a \tag{2.45}$$

The CSP technique is based on maximizing the signal-noise power difference with the unit length constraint on the a [27, 82]. Therefore, to maximize the CSP subject to the constraint $\|a\|^2 = 1$ we have to maximize the criterion function

$$C(a, \gamma) = a^T X^T X a - (1 + \tilde{\alpha}) a^T N^T N a - \gamma(\|a\|^2 - 1) \tag{2.46}$$

where $\tilde{\beta} = \tilde{\alpha} + 1$ and γ is the Lagrange multiplier. This optimization problem is solved by taking the derivative of $C(\alpha, \gamma)$ with respect to a and γ, and setting them to zero

$$\frac{\partial C(a, \gamma)}{\partial a} = 2(X^T X a - \tilde{\beta} N^T N a - \gamma a) = 0 \tag{2.47}$$

$$\frac{\partial C(a, \gamma)}{\partial \gamma} = \|a\|^2 - 1 = 0 \tag{2.48}$$

Equation (2.47) results in,

$$(X^T X - \tilde{\beta} N^T N) a = \gamma a \tag{2.49}$$

From $SNR = \frac{a^T X^T X a}{a^T N^T N a} - 1$, and (2.45) we get

$$SNR = \frac{CSP}{a^T N^T N a} + \tilde{\beta} > \tilde{\beta} \quad for\ all\ \ CSP > 0 \tag{2.50}$$

Equation (2.50) shows that SFA and CSP method are related in a way that if we choose $\tilde{\beta}$ equal to the minimum acceptable signal to noise ratio, then the eigenvectors associated to positive eigenvalues guarantee to have the desirable signal to noise ratio. Thus CSP is a method that finds signals with maximum signal power (variance) while having sufficiently large signal to noise ratio.

For example in [107] the authors have presented a novel multivariate analysis technique to see the different activity patterns under specific conditions. They define "condition-specific response" as the difference between the means of two observed signals under two different conditions. In addition they present, "generalized indicator functions method", which results in an orthogonal set of signals indicating the presence of condition-specific signals.

Their work was in optical imaging which is a challenging problem because of the presence of large background noise. In that paper they define $f_m(t, X)$ as the gray level value (single image) at the cortical position $X = (x, y)$ in the t^{th} frame under experimental condition m, where $t = 1, \ldots, T$, assuming $m = 1, \ldots, M$ different conditions. They use standard Euclidean dot product as the similarity measurement and define $\rho_m(t)$ and call it "response amplitude function". Then, they decompose $\rho_m(t)$ to background amplitude, signal amplitude and noise amplitude, and maximize the variance of signal amplitude while minimizing the residual variance in noise amplitude to find the desired signal.

Chapter 3

THE GENERALIZED SINGULAR VALUE DECOMPOSITION

In this chapter, we present the Generalized Singular Value Decomposition (GSVD) originally proposed by Van Loan in his Ph.D. [31]. In his work, he shows that GSVD can be obtained by factoring two matrices into the products of an orthogonal, a diagonal and a nonsingular matrix, respectively. As evidenced by the algorithms presented in Chapter 2, the GSVD plays a fundamental role in signal processing. In order to keep the presentation self-contained, we present a proof of the GSVD that follows naturally from the lesser known CS decomposition (CSD) due to [80]. In analogy with the SVD, we also make a preliminary distinction between the big problem and the small problem for the GSVD, a topic that will be further discussed in Chapter 4.

The GSVD and cosine sine decomposition (CSD) are useful tools in numerical linear algebra. The cosine sine decomposition (CSD) technique, is introduced and discussed in [91, 80]. Van Loan and Stewart combined the QR factorization and CSD to give an algorithm for GSVD computation [91, 64]. Sun, Paige, Demmel, Verselic and Li in [26, 79] have shown that the CSD is well conditioned [11]. Stewart first developed a method for computing the CSD and the GSVD [91], and later on Van Loan based on the SVD and QR decomposition presented their more efficient method. One of the problems associated with GSVD is the computation of the cross product and/or inversion of a matrix which causes the reduction in accuracy in final results. This can be avoided by CSD technique. As we shall see, CSD is a special case of the GSVD. In CSD we

assume that the matrix has orthonormal columns when in GSVD we don't consider this condition.

For the sake of completeness, we now present several proofs of the GSVD.

3.1 Van Loan's Proof of the GSVD

In this section, we present Van Loan's original proof of the GSVD [64].

Theorem 3.1.1. *Suppose that* $A \in \boldsymbol{R}^{m\times n}$, $B \in \boldsymbol{R}^{p\times n}$ $(m \geq n)$, *then there exist orthogonal matrices* $U \in \boldsymbol{R}^{m\times m}$ *and* $V \in \boldsymbol{R}^{p\times p}$ *and an invertible matrix* $X \in \boldsymbol{R}^{n\times n}$ *such that*

$$\begin{cases} U^T A X = C = diag(c_1, c_2, \ldots, c_n) \\ V^T B X = S = diag(s_1, s_2, \ldots, s_q) \end{cases} \tag{3.1}$$

Here $q = min(p, n)$, $C \in \boldsymbol{R}^{m\times n}$ *and* $S \in \boldsymbol{R}^{p\times n}$ *are diagonal matrices. Applying some mathematical manipulation, Equation (3.1) can be expressed as (3.2).*

We observe that solving the GSVD problem in (3.1) is the same as solving

$$s_i{}^2 A^T A X = c_i{}^2 B^T B X \tag{3.2}$$

to find the generalized singular values and corresponding singular vectors. In matrix form we have

$$A^T A X = B^T B X (S^T S)^{-1} C^T C = B^T B X \Lambda \tag{3.3}$$

Where, $\Lambda = (S^T S)^{-1} C^T C$.

Proof:

From Equation (3.1) we get

$$A^T = X^{-T} C^T U^T \tag{3.4}$$

and

$$B^T = X^{-T} S^T V^T \tag{3.5}$$

Here, for simplicity we use X^{-T} notation for X^{-1^T}. Now if we calculate A^TAX and B^TBX from Equations (3.4) and (3.5) we have

$$A^TAX = X^{-T}C^TC \tag{3.6}$$

and

$$B^TBX = X^{-T}S^TS \tag{3.7}$$

Now we find X^{-T} in Equation (3.7) and substitute it into Equation (3.6) giving

$$A^TAXS^TS = B^TBXC^TC \tag{3.8}$$

which completes the proof.

3.2 A Variational Proof of the GSVD

Consider the optimization problem

$$D(x) = \max_{Bx \neq 0} \frac{\|Ax\|}{\|Bx\|} \tag{3.9}$$

where $X \in \mathbf{R}^{n \times n}$ of all vectors $x \in \mathbf{R}^n$ and $\|\cdot\|$ is the 2-norm defined in the previous chapter. Here, $A \in \mathbf{R}^{m \times n}$, and $B \in \mathbf{R}^{p \times n}$. Then the GSVD of (A, B) are the values of the objective function in Equation (3.9). To provide the variational proof we differentiate the objective function $D(x)$ with respect to x and set it to zero.

To this end, we differentiate

$$D(x) = \frac{x^TA^TAx}{x^TB^TBx} \tag{3.10}$$

with respect to x as

$$\frac{\partial D(x)}{\partial x} = \frac{2A^TAx(x^TB^TBx) - 2B^TBx(x^TA^TAx)}{(x^TB^TBx)^2} = 0 \tag{3.11}$$

Equation (3.11) in matrix form results in

$$A^TAX(X^TB^TBX) = B^TBX(X^TA^TAX). \tag{3.12}$$

For simplicity we define scalars α and β as

$$\alpha = X^T A^T A X \tag{3.13}$$

$$\beta = X^T B^T B X \tag{3.14}$$

Substituting (3.13) and (3.14) into (3.12), we have,

$$\beta A^T A X = \alpha B^T B X \tag{3.15}$$

Equation (3.15) defines the generalized singular value decomposition of the matrix pair (A, B). In Equation (3.15), the columns of the matrix X, consist of the generalized singular vectors and $\lambda = \alpha/\beta$ where $\beta \neq 0$ is the generalized singular value associated to X. Clearly the GSVD is a more general form of the SVD. If one of the matrices A or B in Equation (3.15) equal to identity matrix, then GSVD turns out to be an ordinary SVD.

3.3 A CS Decomposition Proof

In this section, we introduce the CS Decomposition (CSD) and show how the GSVD may be derived from the CSD. The exposition here primarily follows that in [11]. The Cosine Sine, or CS Decomposition (CSD), is a decomposition of an orthonormal matrix, and as we shall see is a special case of GSVD [11]. The CSD has been proposed for finding the distance between subspaces in the sense that the CSD finds the angles between subspaces [91].

Theorem 3.3.1. *(Theorem CS Decomposition))[33] Suppose that matrix* $Q \in \boldsymbol{R}^{(m_1+m_2)\times n}$ *has orthonormal columns partitioned into two parts* $Q_1 \in \boldsymbol{R}^{m_1\times n}$, *and* $Q_2 \in \boldsymbol{R}^{m_2\times n}$, *such that*

$$Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \tag{3.16}$$

where, $m_1, m_2 \geq n$ *and* $Q^TQ = Q_1^TQ_1 + Q_2^TQ_2 = I$. *Then, there exist orthogonal matrices* $U \in \boldsymbol{R}^{m_1 \times m_1}$, $V \in \boldsymbol{R}^{m_2 \times m_2}$, *and* $W \in \boldsymbol{R}^{n \times n}$, *and diagonal matrices* $C \in \boldsymbol{R}^{m_1 \times n}$ *and* $S \in \boldsymbol{R}^{m_2 \times n}$ *such that*

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}^T \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} W = \begin{bmatrix} C \\ S \end{bmatrix} \tag{3.17}$$

where, in Equation (3.17) matrices C *and* S *are the diagonal matrices [33] satisfying* $C^TC + S^TS = I$.

Proof:

To show that $C^TC + S^TS = I$ we use Equation (3.17) which leads to

$$Q_1^TQ_1 + Q_2^TQ_2 = W(C^TC + S^TS)W^T \tag{3.18}$$

and since, $Q_1^TQ_1 + Q_2^TQ_2 = I$ and matrix W is an orthogonal matrix therefore if we pre and post multiply the right hand side of Equation (3.18) by W^{-1} and W^{-T} simultaneously then

$$C^TC + S^TS = I \tag{3.19}$$

The main part of the proof of the CSD begins by first calculating the QR decomposition of the matrix $(A^T, B^T)^T$, i.e.,

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R \tag{3.20}$$

Now because in Equation (3.20) matrix $\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$ is an orthonormal matrix then we can apply CSD. Then, using Theorem (3.3.1) and Equation (3.17), we get

$$A = Q_1R = UCW^TR \tag{3.21}$$

and

$$B = Q_2R = VSW^TR. \tag{3.22}$$

In Equations (3.21), and (3.22) if we consider

$$X^{-1} = W^T R \tag{3.23}$$

then we have proved that

$$\begin{cases} U^T A X = C = diag(c_1, c_2, \ldots, c_n) \\ V^T B X = S = diag(s_1, s_2, \ldots, s_q) \end{cases} \tag{3.24}$$

3.4 Theorem by Paige and Saunders

Paige and Saunders extended Van Loan's GSVD to treat all possible cases [80].Of particular importance is the case where Van Loan's matrix X fails to have an inverse.

Theorem 3.4.1. *(Paige and Saunders) If $A \in \boldsymbol{R}^{m\times p}$, $B \in \boldsymbol{R}^{n\times p}$, there exist orthonormal matrices $U \in \boldsymbol{R}^{m\times p}$, $V \in \boldsymbol{R}^{m\times p}$, $X \in \boldsymbol{R}^{n\times n}$ and diagonal matrices $S \in \boldsymbol{R}^{p\times p}$, and $C \in \boldsymbol{R}^{p\times p}$ such that:*

$$\begin{cases} A = UCY^T \\ B = VSY^T \end{cases} \tag{3.25}$$

where, $X^{-1} = Y^T$ and $Y \in \boldsymbol{R}^{n\times n}$ in Equation (3.28).[1]

Suppose that $rank(A) = r_a$, $rank(B) = r_b$, and $rank[A \quad B]^T = r_{ab}$ then in Equations (3.26) and (3.27), $I_A \in \mathbf{R}^{(r_{ab}-r_b)\times(r_{ab}-r_b)}$, and $I_B \in \mathbf{R}^{(r_{ab}-r_a)\times(r_{ab}-r_a)}$ are unit matrices, $0_A \in \mathbf{R}^{(p-r_a)\times(p-r_a)}$, and $0_B \in \mathbf{R}^{(p-r_b)\times(p-r_b)}$ are zero matrices, and $C_A \in \mathbf{R}^{(r_a+r_b-r_{ab})\times(r_a+r_b-r_{ab})}$, and $S_B \in \mathbf{R}^{(r_a+r_b-r_{ab})\times(r_a+r_b-r_{ab})}$ are real diagonal matrices [73].

---

[1] In Matlab command,

$$C = \begin{pmatrix} 0_A & & \\ & C_A & \\ & & I_A \end{pmatrix} \tag{3.26}$$

and

$$S = \begin{pmatrix} I_B & & \\ & S_B & \\ & & 0_B \end{pmatrix} \tag{3.27}$$

3.5 Big versus Small problem

Here, we define and introduce the notion of big problem and the small problem. Suppose we have two matrices $A \in \mathbf{R}^{m\times n}$, and $B \in \mathbf{R}^{p\times n}$, where $m \gg n$, then in generalized singular value form we have

$$\begin{cases} A = UCX^{-1} \\ B = VSX^{-1} \end{cases} \Longrightarrow A^TAX = \lambda B^TBX \tag{3.28}$$

Here, λ is the diagonal element in the diagonal matrix Λ in Equation (3.3). Let us refer to this problem as small problem because of the size of matrix A^TA which is $n \times n$. Here, matrices U and V are orthogonal and X is nonsingular. In terms of big problem, we have

$$\begin{cases} XC^{-1}U^TAA^TUC^{-T}X^T = I \\ XS^{-1}V^TBB^TVS^{-T}X^T = I \end{cases} \tag{3.29}$$

or

$$AA^T = Z^TBB^TZ \tag{3.30}$$

Here, $Z = VS^{-T}C^TU^T$ and we refer to Equation (3.30) as the big problem given it is of size $m \times m$.

We shall see in Chapter 4 that the big and small problems arise naturally in different contexts of data analysis.

3.6 Geometry of Subspaces

Following are several facts concerning subspaces associated with the GSVD. For additional properties of subspaces associated with the GSVD please see [105]. In what follows we assume the Matlab GSVD form, i.e.,

```
[U,V,X,C,S]=gsvd(A,B,0)
```

where

$$A = UCX^T$$

and

$$B = USX^T.$$

3.6.1 Theorem for the row space in general

Theorem 3.6.1. *Suppose that we have all conditions in Theorem 3.4.1; here we intend to work on the row spaces of matrices A and B. Therefore we work with A^T and B^T. Using the previous theorem we get:*

$$\begin{cases} A = UCX^{-1} \\ B = VSX^{-1} \end{cases} \tag{3.31}$$

[2] *Which results in*

$$\begin{pmatrix} A \\ B \end{pmatrix} X = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} C \\ S \end{pmatrix} \tag{3.32}$$

and

$$\begin{pmatrix} A^T \\ B^T \end{pmatrix} X_p = \begin{pmatrix} U_p & 0 \\ 0 & V_p \end{pmatrix} \begin{pmatrix} C_p \\ S_p \end{pmatrix}. \tag{3.33}$$

Where $X_p{}^{-1} = Y_p{}^T$, and

$$C_p = \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_A \end{pmatrix} \tag{3.34}$$

and

$$S_p = \begin{pmatrix} \Sigma_B & 0 \\ 0 & 0 \end{pmatrix} \tag{3.35}$$

Here $\Sigma_A \in \mathbf{R}^{r_a \times r_a}$, and $\Sigma_B \in \mathbf{R}^{r_b \times r_b}$ are diagonal matrices of size $rank(A) = r_a$ and $rank(B) = r_b$ [80].

---

[2]We should add that in Matlab command we have: $[U, V, Y, C, S] = gsvd(A, B, 0)$ that we use for the column space and we use $[U_p, V_p, Y_p, C_p, S_p] = gsvd(A^T, B^T, 0)$ for the row space.

3.6.2 Theorem for the row space with three block partitioning

Theorem 3.6.2. *Let* $U = [U_1, U_2, U_3]$, *and* $V = [V_1, V_2, V_3]$, *and* $Y = [Y_1, Y_2, Y_3]$ *be compatibly partitioned with the block partitioning of* C *and* S *demonstrated in Theorem 3.4.1, then:*

- *a)* $span(Y_3, Y_2 S_A) = R(A^T)$
- *b)* $span(Y_1, Y_2 S_B) = R(B^T)$
- *c)* $span(U_2 S_A, U_3) = R(A)$
- *d)* $span(V_1, V_2 S_B) = R(B)$
- *e)* $span(U_1) = R(A)^{\perp}$
- *f)* $span(V_3) = R(B)^{\perp}$

Proof:

a) From Equation (3.32) or $A^T U = Y C^T$, and with the partitioning of $Y = [Y_1, Y_2, Y_3]$ we conclude that:

$$R(A^T) = [Y_1, Y_2, Y_3] \begin{pmatrix} 0_A & & \\ & S_A & \\ & & I_A \end{pmatrix} = [0, Y_2 S_A, Y_3] \tag{3.36}$$

b) From Equation (3.32) or $B^T V = Y S^T$, and the partitioning $Y = [Y_1, Y_2, Y_3]$ we conclude that:

$$R(B^T) = [Y_1, Y_2, Y_3] \begin{pmatrix} I_B & & \\ & S_B & \\ & & 0_A \end{pmatrix} = [Y_1, Y_2 S_B, 0] \tag{3.37}$$

c) From $AX = UC$, and with the partitioning of $U = [U_1, U_2, U_3]$ we conclude that:

$$R(A) = [U_1, U_2, U_3] \begin{pmatrix} 0_A & & \\ & S_A & \\ & & I_A \end{pmatrix} = [0, U_2 S_A, U_3] \tag{3.38}$$

d) From $BX = VS$, and the partitioning $V = [V_1, V_2, V_3]$ we conclude that:

$$R(B) = [V_1, V_2, V_3] \begin{pmatrix} I_B & & \\ & S_B & \\ & & 0_A \end{pmatrix} = [V_1, V_2 S_B, 0] \tag{3.39}$$

To prove parts e) and f) we use the following definition:

$$R(A)^{\perp} = \{Z_1 \in R(A)^{\perp}; Z_1^T Y = 0; Y = AX\} \tag{3.40}$$

and

$$R(B)^{\perp} = \{Z_2 \in R(B)^{\perp}; Z_2^T Y = 0; Y = BX\} \tag{3.41}$$

Therefore, span$(U_1) = R(A)^{\perp}$, and span$(V_3) = R(B)^{\perp}$.

3.6.3 Theorem for the row space with two block partitioning

In the previous section, we partitioned the matrices into three block matrices and in this section we will partition them into two block matrices. Now if we compatibly partition U_p and V_p with the block partitioning of C_p and S_p using Section 3.6.1, we could create the following Theorem.

Theorem 3.6.3. *Let $U_p = [U_{p_1}, U_{p_2}]$, and $V_p = [V_{p_1}, V_{p_2}]$ be compatibly partitioned with the block partitioning of C_p and S_p in section 3.6.1, then*

- *a) span$(U_{p_2}) = R(A^T)$*
- *b) span$(V_{p_1}) = R(B^T)$*
- *c) span$(U_{p_1}) = R(A^T)^{\perp} = Null(A)$*

- *d) span*$(V_{p_2}) = R(B^T)^{\perp} = Null(B)$

Proof

a) From

$$\begin{pmatrix} A^T \\ B^T \end{pmatrix} Xp = \begin{pmatrix} U_p & 0 \\ 0 & V_p \end{pmatrix} \begin{pmatrix} C_p \\ S_p \end{pmatrix} \tag{3.42}$$

$A^T X_p = U_p C_p$, and $U_p = [U_{p_1}, U_{p_2}]$ we conclude that

$$R(A^T) = [U_{p_1}, U_{p_2}] \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_A \end{pmatrix} = [0, U_{p_2}\Sigma_A] \tag{3.43}$$

b)From

$$\begin{pmatrix} A^T \\ B^T \end{pmatrix} X_p = \begin{pmatrix} U_p & 0 \\ 0 & V_p \end{pmatrix} \begin{pmatrix} C_p \\ S_p \end{pmatrix} \tag{3.44}$$

$B^T X_p = V_p S_p$, and $V_p = [V_{p_1}, V_{p_2}]$ we conclude that

$$R(B^T) = [V_{p_1}, V_{p_2}] \begin{pmatrix} \Sigma_B & 0 \\ 0 & 0 \end{pmatrix} = [V_{p_1}\Sigma_B, 0] \tag{3.45}$$

To prove parts c) and d) we use, $\begin{pmatrix} A^T \\ B^T \end{pmatrix} X_p = \begin{pmatrix} U_p & 0 \\ 0 & V_p \end{pmatrix} \begin{pmatrix} C_p \\ S_p \end{pmatrix}$, where, $U_p = [U_{p_1}, U_{p_2}]$, $V_p = [V_{p_1}, V_{p_2}]$, and

$$R(A^T)^{\perp} = \{Z_1 \in R(A^T)^{\perp}; Z_1^T Y = 0; Y = A^T X\} \tag{3.46}$$

and

$$R(B^T)^{\perp} = \{Z_2 \in R(B^T)^{\perp}; Z_2^T Y = 0; Y = B^T X\} \tag{3.47}$$

Therefore; $Z_1 = U_{p_1}$ and $Z_2 = V_{p_2}$.

Chapter 4

SIGNAL FRACTION ANALYSIS (SFA)

In this chapter, we present Signal Fraction Analysis (SFA) and we establish a foundation for Chapter 5 which introduces a nonlinear extension of SFA, i.e., kernel SFA. As described in Chapter 1, SFA is a framework for the construction of bases that maximize a signal-to-signal ratio. This work represents an extension of the maximum noise fraction (MNF) approach that was introduced for the de-noising of multi-spectral satellite images, see e.g., [92, 35]. Related approaches include *oriented signal-to-signal ratio* [12] and *Common Spatial Patterns*, see, e.g., [27, 82].

Signal fraction analysis provides a unifying perspective that addresses the decomposing of a data set into constituent features. In part, its strength lies in the generality of the approach, the manner in which features can be computed is essentially unlimited. In particular, we consider a data matrix X and two mappings of the data matrix

$$f : X \rightarrow Y = f(X)$$

and

$$g : X \rightarrow Z = g(X)$$

The purpose of the mapping f is to identify characteristics of the data distinct from the mapping g. As described below, we are then able to compute a subspace representation for the data by solving a GSVD problem. We suggest that such transformations may possess energy in either the row or columns of the images and seek bases that reflect this.

We also propose side constraints to the optimization problem to allow greater control over the nature of the subspace representations.

One important characteristics of the SFA transformation, which is not shared with the PCA transformation, is the invariability to linear scaling [51, 48]. A disadvantage of SFA transformation for noise removal is its need for estimation of noise covariance matrix [22, 51]. Different methods of estimating noise covariance have been introduced in the literature [35, 61]. In communications, the noise covariance is estimated via transmitting a known (pilot) signal [61, 101, 90].

In Section 4.1, we revisit the SVD and motivate the notion of row and column energy optimization. In Section 4.2 we derive Signal Fraction Analysis (SFA) and we show its connection to generalized singular value decomposition (GSVD). In Section 4.3 we present the special case of Maximum Noise Fraction and discuss the estimation of the noise covariance. In Section 4.4 we consider SFA with constraints. Finally, we show a connection between SFA and canonical correlation analysis (CCA) in Section 4.5. We conclude with a summary of contributions in Section 4.6

4.1 The Singular Value Decomposition Revisited

If we let ϕ be a basis vector for $\mathcal{R}(X)$ we can define the quantity

$$\alpha_c(X, \phi) = \frac{1}{n} \sum_{j=1}^{n} (\phi^T x^{(j)})^2$$

which is just the mean squared projection of the columns of X onto the basis vector ϕ. We may refer to this term as the *column energy* of X associated with ϕ and rewrite it as

$$\alpha_c(X, \phi) = \phi^T X X^T \phi$$

where we have omitted the scalar $1/n$.

Similarly, if we let ψ be a basis vector for the row space $\mathcal{R}(X^T)$ we can define the quantity

$$\alpha_r(X, \psi) = \frac{1}{m} \sum_{j=1}^{m} (\psi^T y^{(j)})^2$$

which is just the mean squared projection of the columns of $X^T = [y^{(1)}|\ldots|y^{(m)}]$ onto the basis vector ψ. We may rewrite this *row energy* of X associated with ψ as

$$\alpha_r(X, \psi) = \psi^T X^T X \psi$$

where we have now omitted the scalar $1/m$.

The singular value decomposition of an $m \times n$ matrix X is special in that it simultaneously determines optimal bases for both the row and column spaces. This is achieved by maximizing the quadratic form

$$\alpha(\phi, \psi) = \max \phi^T X \psi$$

subject to the side constraints that both ϕ and ψ have unit length. Alternatively, the solutions to this problem may be computed by maximizing either the row energy $\alpha_r(X, \psi)$ over ψ or the column energy $\alpha_c(X, \phi)$ over ϕ.

Differentiating the column energy $\alpha_c(X, \phi)$ with respect to ϕ produces the necessary condition

$$XX^T\phi = \lambda\phi$$

Alternatively, differentiating the row energy $\alpha_r(X, \psi)$ with respect to ψ produces the necessary condition

$$X^T X\psi = \lambda\psi.$$

We recognize these equations as the eigenvector equations for the left singular vectors Φ and right singular vectors Ψ of X. As is well-known, Ψ and Φ are linked by the SVD via

$$\Phi = X\Psi\Sigma^\dagger$$

where the dagger denotes the pseudo-inverse. Thus, we see that determining Φ to maximize the column energy also produces the Ψ that maximizes the row energy. While these facts are well known to be true in the case of the SVD, similar considerations are actually false in the generalized SVD.

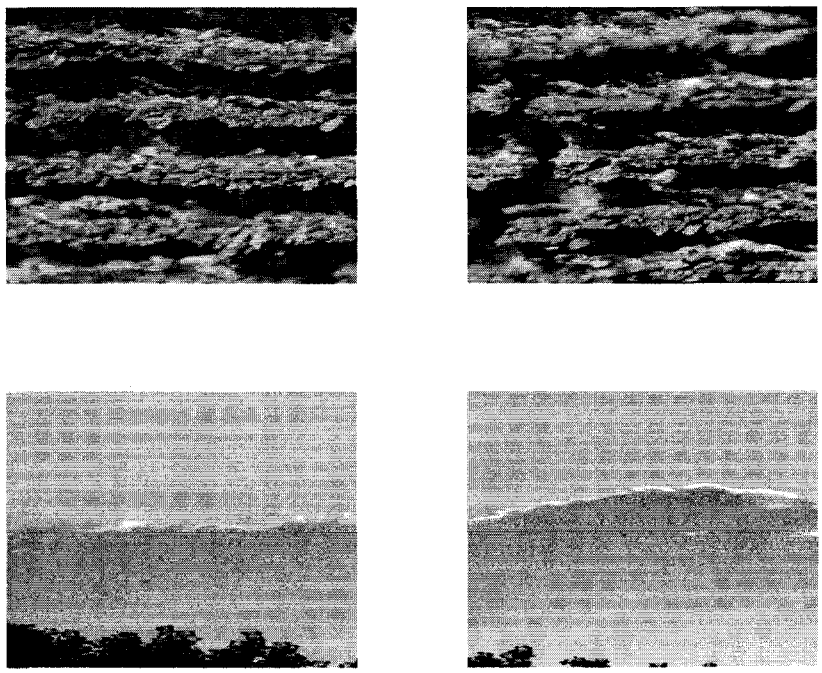


Figure 4.1: The top two 1000×500 image matrices were obtained by dividing a 1000×1000 magnified image of wood in half. Similarly, we obtained two images of a sunset in the bottom row.

4.2 Signal Fraction Analysis

In order to identify a *best* basis associated with two data matrices X and Y we may propose to determine a basis such that when the column energy of X is large, then the column energy of Y is small and visa versa. Such a basis is produced by solving optimization problem

$$\lambda = \max_{\phi} \frac{\alpha_c(X, \phi)}{\alpha_c(Y, \phi)} \tag{4.1}$$

Or in other words, computing the solutions to

$$\alpha_c(Y, \phi^{(i)}) X X^T \phi^{(i)} = \alpha_c(X, \phi^{(i)}) Y Y^T \phi^{(i)} \tag{4.2}$$

that is to say, the transpose of the standard formulation of the generalized singular vector problem. This problem has been proposed by [12] where Equation (4.1) is referred to as the *signal-to-signal ratio.*

For spatio-temporal patterns there is potential structure along both the columns and rows suggesting the alternative optimization problem

$$\gamma = \max_{\psi} \frac{\alpha_r(X, \psi)}{\alpha_r(Y, \psi)} \tag{4.3}$$

is important to separate X and Y according to row energy. We will see below that the optimization problem given by Equation (4.3) is related to a technique proposed for removing noise from multi-spectral satellite imagery known as *maximum noise fraction* [35]. Now the solutions to Equation (4.3) must satisfy

$$\alpha_r(Y, \psi^{(i)}) X^T X \psi^{(i)} = \alpha_r(X, \psi^{(i)}) Y^T Y \psi^{(i)} \tag{4.4}$$

Now, unlike the SVD, the optimization problems given by Equations (4.1) and (4.3) are not linked.

As a preliminary experiment in investigating the properties of signal fraction analysis we compute the solutions to Equation (4.4) in two different ways and plot the associated

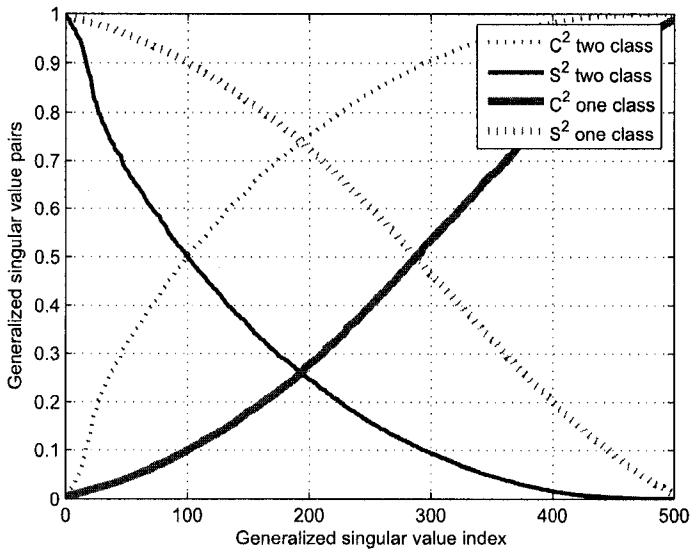


Figure 4.2: This is a plot of the 500 generalized singular values where we take A and B in the GSVD problem to be the two different wood images (one class) and where we take A to be wood and B to be sunset.

generalized singular values in Figure 4.2. In the two class problem the matrix A consists of the wood image (top left Figure 4.1) and the matrix B consists of the sun image (bottom right of Figure 4.1). In the one class problem the matrix A consists of the wood image (top left Figure 4.1) and the matrix B consists of the wood image (top right of Figure 4.1).

4.2.1 Examples of SFA Filtering

Here we briefly consider the more general optimization problem

$$\max \frac{\psi f(X)^T f(X)\psi}{\psi g(X)^T g(X)\psi} \tag{4.5}$$

where X is a given data set to be investigated. In many cases, linear transformations such as wavelet or Fourier projections onto interesting scales or frequencies will be sufficient and that is what we explore in these examples. For example, in Figure 4.3 we have mapped an EEG signal via a Fourier transformation to the spectra associated with

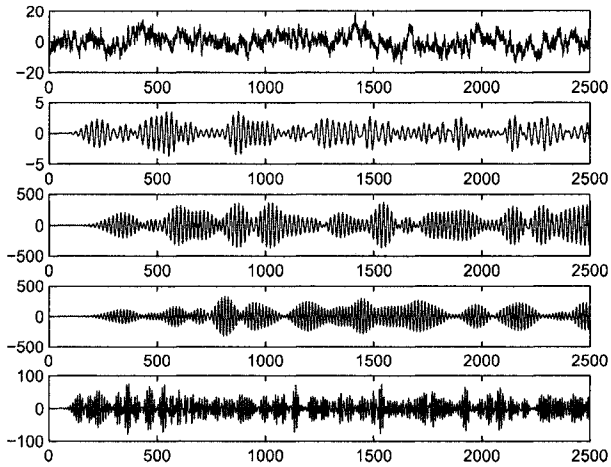


Figure 4.3: Band-pass Fourier filters applied to subject one, task two (math), trial one. From top to bottom: raw data, alpha filter, low beta filter, mid-beta filter, high-beta filter.

alpha waves, low beta waves, mid-beta waves and high-beta waves. In principle, such filtering may have potential benefits in analyzing features in the data.

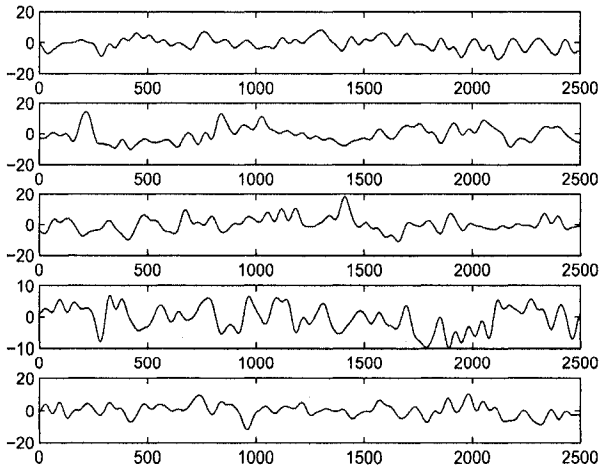
In Figure 4.4, we see the results of mapping the EEG data to a wavelet and scaling subspace (Daubechies wavelet). Here we note that a detail subspace generates a signal similar to a derivative and therefore has important ramifications for denoising. Note also that one may conceive of adapting f and g in Equation (4.5) using nonlinear function fitting approaches such as Radial Basis Functions.

Basically, the idea now is to solve the generalized singular value problem

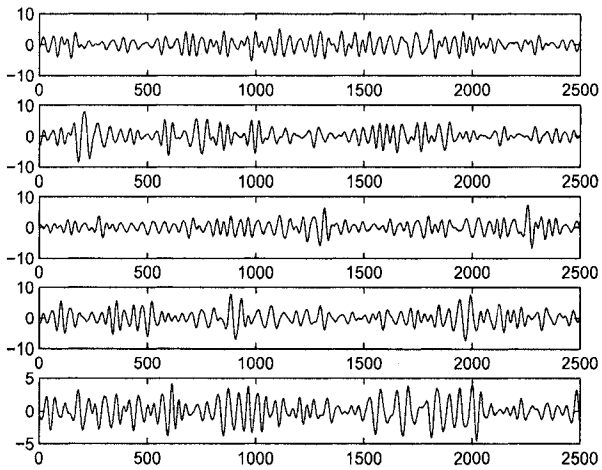
$$s_i^2 A^T A \psi_i = c_i^2 B^T B \psi_i$$

where, $A = f(X)$ and $B = g(X)$. This idea is very general and we need to carry out experiments to explore its utility. In the simplest experiment we take $A = X$, i.e., the data set and $B = W_1$, i.e., the finest wavelet subspace. We may also seek a scale for splitting that is especially useful by setting

$$B = W_i$$



(a)



(b)

Figure 4.4: Wavelet transformation of the EEG data associated with the rest task for subject one trial one. (a) Scaling subspace projections of channel one. (b) Wavelet subspace projections of channel one.

classification rate using DB10	Task 1	Task 2	Task 3	Task 4
Task 2	0.61			
Task 3	0.58	0.74		
Task 4	0.77	0.76	0.71	
Task 5	0.72	0.78	0.68	0.59

Table 4.1: SFA on classification rate for EEG signals with the Wavelet transformation DB10, KNN=1:10

classification rate using DB10	Task 1	Task 2	Task 3	Task 4
Task 2	0.83 $d=4$			
Task 3	0.77 $d=3$	0.89 $d=4$		
Task 4	0.85 $d=3$	0.92 $d=2$	0.82 $d=3$	
Task 5	0.80 $d=2$	0.97 $d=2$	0.83 $d=4$	0.68 $d=2$

Table 4.2: KSFA on classification rate for EEG signals with the Wavelet transformation DB10, KNN=1:10

with i running over the scales. We may also let

$$A = S_j, \quad B = W_j$$

Table 4.1 shows the classification rate for tasks one to five when applying SFA and K-nearest neighbor KNN=1:10. The classification rate is defined as the ratio of the number of correctly classified and the total number. Note that here the polynomial is of degree one. We also did the experiment with higher polynomial degrees ($d = 2, 3, 4$) (see Chapter 5 for details on Kernel Signal Fraction Analysis) in Table 4.2 and the result shows that mapping the data into higher dimension leads to a considerably better classification rates. It seems that this scheme is able to classify tasks two and five (task 2 is the imagined letter writing and task 5 is geometric object rotation) better than other tasks with higher classification rate and polynomial degree of two.

4.3 Maximum Noise Fraction (MNF)

In this section, we provide a self-contained treatment of MNF [92, 35]. This background discussion is based on the assumption that the signal consists of two additive

components. Let, $X \in R^{n \times p}$, $n > p$ be a multivariate data set, an addition of the desired data (signal) S, and undesired signal (noise) N. Although the columns of X were multispectral images in [92, 35], they may be signals, such as noisy time series of a process.

Therefore, the observed data is represented by

$$X = S + N \tag{4.6}$$

From Equation (4.6) we have

$$X^T X = S^T S + N^T N + S^T N + N^T S \tag{4.7}$$

We define the signal to noise ratio (SNR), the ratio of the signal variance and the noise variance. That is,

$$SNR = \frac{var(X)}{var(N)} \tag{4.8}$$

In addition, noise fraction is defined as the ratio of noise variance to the total variance. That is,

$$\mu = \frac{var(N)}{var(X)} \tag{4.9}$$

The maximum signal fraction transformation is a linear transformation defined by

$$Y = Xa = Sa + Na, \tag{4.10}$$

where a is a $p \times 1$ vector. Here, the problem is to determine a in order to maximize the signal to noise ratio for Y. The SNR for Y is:

$$SNR = \frac{var(Sa)}{var(Na)} \tag{4.11}$$

Therefore,

$$SNR = \frac{E(a^T S^T Sa) - E(a^T S^T)E(Sa)}{E(a^T N^T Na) - E(a^T N^T)E(Na)} \tag{4.12}$$

Here, we assume a zero mean data set (i.e., $E(S) = E(N) = 0$). Hence, $E(Sa) = E(S)a = 0$. Therefore, Equation (4.12) corresponds to:

$$SNR = \frac{E(a^T S^T Sa)}{E(a^T N^T Na)} \tag{4.13}$$

From Equation (4.7) we have:

$$a^T X^T Xa = a^T S^T Sa + a^T S^T Na + a^T N^T Sa + a^T N^T Na \tag{4.14}$$

Computing $a^T S^T Sa$ from Equation (4.14) and substituting it in Equation (4.13) results in

$$SNR = \frac{E(a^T X^T Xa) - E(a^T S^T Na) - E(a^T N^T Sa - E(a^T N^T Na)}{E(a^T N^T Na)} \tag{4.15}$$

Assuming signals are independent from each other and from the noise, the second term in the numerator of Equation (4.15) corresponds to:

$$E(a^T S^T Na) = a^T E(S^T N)a = a^T E(S^T)E(N)a = 0 \tag{4.16}$$

Similarly, the third term in the numerator of Equation (4.15) equals to zero, and it simplifies to

$$SNR = \frac{E(a^T X^T Xa)}{E(a^T N^T Na)} - 1 \tag{4.17}$$

Because we don't have access to the statistical information, we must use the estimation of the covariance matrices to compute the SNR. Assuming n is large, we can estimate the statistical means in Equation (4.17) via the sample mean. In this case, Equation (4.17) corresponds to

$$SNR = \frac{a^T X^T Xa}{a^T N^T Na} - 1 \tag{4.18}$$

The MNF technique is based on maximizing the signal to noise ratio powers. Therefore, to maximize the SNR we have to maximize the term

$$D(a) = \frac{a^T X^T Xa}{a^T N^T Na} \tag{4.19}$$

This optimization problem is solved by taking the derivative of $D(a)$ with respect to a and setting it to zero

$$\frac{\partial D(a)}{\partial a} = \frac{2X^TXa(a^TN^TNa) - 2N^TNa(a^TX^TXa)}{a^TN^TNa} = 0 \tag{4.20}$$

Equation (4.20) results in,

$$X^TXa(a^TN^TNa) = N^TNa(a^TX^TXa). \tag{4.21}$$

Here, we define scalars α and β as

$$\alpha = a^TX^TXa \tag{4.22}$$

$$\beta = a^TN^TNa \tag{4.23}$$

Substituting Equations (4.22) and (4.23) into Equation (4.21), we have,

$$\beta X^TXa = \alpha N^TNa \tag{4.24}$$

Now, defining

$$\lambda = \frac{\alpha}{\beta} \quad for \quad \beta \neq 0 \tag{4.25}$$

and from Equation (4.18), we conclude:

$$\lambda = SNR + 1 \tag{4.26}$$

Thus, substituting Equation (4.25) in Equation (4.24), we obtain the following generalized singular value problem

$$X^TXa = \lambda N^TNa \tag{4.27}$$

In Equation (4.27), the parameter a is the generalized singular vector and λ is the generalized singular value associated to a which maintains the connection between MNF and GSVD.

Therefore, given matrix X, we can recover S by applying MNF under the following conditions:

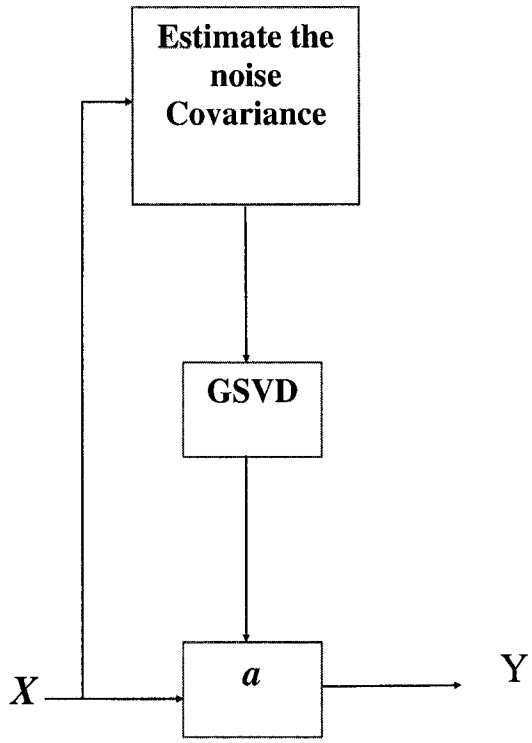


Figure 4.5: MNF Process.

1. n must be large enough (i.e.,X is a tall matrix) in order to estimate the statistical mean via the sample mean.

2. $E(S) = E(N) = E(X) = 0$ (i.e., the columns of S, N and X are assumed to have zero mean).

3. Signals are uncorrelated from the noise,(i.e., $E(S^T N) = E(N^T S) = 0$).

As a summary, the MNF process can be summarized in Figure 4.5.

The generalized singular vectors (a_i 's) are orthonormal with respect to matrix $N^T N$, and are orthogonal with respect to matrix $X^T X$, (see Appendix A and Appendix B for the proof).

4.3.1 Estimation of the Noise Covariance Matrix

In the process of MNF, the noise covariance matrix should be known; however, in many applications, it is unknown and needs to be estimated. Some of the noise covariance estimation techniques have been introduced in the literature [101, 35, 61, 90, 37]. Little work has been done to address limitations of noise covariance matrix estimation techniques.

In [35], Green has used Min/Max Autocorrelation Factors (MAF) technique for noise covariance matrix estimation of correlated data. MAF transform finds several orthogonal linear combinations of highly correlated data. He has used the covariance between neighboring differences to estimate the noise covariance. This technique is particularly useful for separating salt and pepper noise (the noise caused by errors in transmission). In communications, the noise covariance is estimated via pilot signals or when a signal is not transmitted for a period of time [101, 61, 90, 37]. This method reduces the throughput of the communication system via preventing the transmission of data or via transmission of pilot signals and can not be applied to all applications.

We have studied the method of estimating the noise covariance matrix in [41] and we will go over the estimate for the covariance of the noise via simple differencing (i.e., when the noise is estimated as the difference between the current and the neighboring data), and we derive the assumptions needed for this scheme in some details.

4.3.2 Differencing method

As we mentioned in Section 4.3 (see Equation (4.27)), to apply the MNF one requires the estimation of noise covariance matrix. Green [35] has introduced differencing method as an approximation for noise covariance matrix estimation. In this section, we briefly discuss the conditions under which the differencing method holds.

Suppose that $X \in R^{n \times p}$, $n > p$, be a multivariate data set, an addition of the desired data (signal), S, and undesired one (noise), N. Therefore, the observed data can

be represented by $X = S + N$. We denote the difference matrix as dX in which each row i (i.e., $dX(i,:)$) is the difference between the $(i+1) - th$ row of X (i.e., $X(i+1,:)$) and the $i - th$ row of X (i.e., $X(i,:)$). Let denote X and dX in their matrix form via Equations (4.28) and (4.3.2) respectively and suppose that we can estimate the noise covariance using the difference matrix as: $N^H N = dX^H dX/2$.

Therefore if

$$X = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \cdot & \cdot & \cdot & x_p(t_1) \\ x_1(t_2) & x_2(t_2) & \cdot & \cdot & \cdot & x_p(t_2) \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ x_1(t_n) & x_2(t_n) & \cdot & \cdot & \cdot & x_p(t_n) \end{pmatrix} \tag{4.28}$$

Then,

$$dX = \begin{pmatrix} x_1(t_2) - x_1(t_1) & x_2(t_2) - x_2(t_1) & \cdot & \cdot & \cdot & x_p(t_2) - x_p(t_1) \\ x_1(t_3) - x_1(t_2) & x_2(t_3) - x_2(t_2) & \cdot & \cdot & \cdot & x_p(t_3) - x_p(t_2) \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ x_1(t_n) - x_1(t_{n-1}) & x_2(t_n) - x_2(t_{n-1}) & \cdot & \cdot & \cdot & x_p(t_n) - x_p(t_{n-1}) \end{pmatrix} \tag{4.29}$$

where, $X \in \mathbf{R}^{n \times p}$, and $dX \in \mathbf{R}^{(n-1) \times p}$. Here, as we assumed, we have the additive noise signal as

$$x_m(t_i) = s_m(t_i) + n_m(t_i) \tag{4.30}$$

Therefore,

$$x_m(t_i) - x_m(t_{i-1}) = s_m(t_i) - s_m(t_{i-1}) + n_m(t_i) - n_m(t_{i-1}) \tag{4.31}$$

To simplify, we calculate matrix dX including elements of $x_m(t_i) - x_m(t_{i-1})$ and compute matrices dS and dN based on elements $s_m(t_i) - s_m(t_{i-1})$ and $n_m(t_i) - n_m(t_{i-1})$ respectively when $m = 1, \ldots, p$ and $i = 1, \ldots, n$ (see Equation (4.3.2)). Then in terms of matrix notation we use $dX = X_t - X_{t-1}$, $dS = S_t - S_{t-1}$, and $dN = N_t - N_{t-1}$ where

X_t is the same as X in Equation (4.28) and X_{t-1} is the shifted version of X_t and the same applies to S_t, S_{t-1}, N_t, and N_{t-1}. Therefore

$$dX = dS + dN \tag{4.32}$$

which leads to

$$dX^T dX = dS^T dS + dN^T dN \tag{4.33}$$

Here, using $dN = N_t - N_{t-1}$, $N_{t-1}^T N_{t-1} \cong N_t^T N_t$ and $N_{t-1}^T N_t = N_t^T N_{t-1} \cong 0$, from Equation (4.33) we get

$$dX^T dX = dS^T dS + 2N_t^T N_t \tag{4.34}$$

Assuming the sampling rate is high enough such that $S_t \simeq S_{t-1}$ concludes $dS^T dS \approx 0$ which leads to

$$dX^T dX = 2N_t^T N_t \tag{4.35}$$

This means that we could use the differencing idea for the observed data and apply it to estimate the noise covariance matrix.

4.4 SFA with Constraints

Ideally we would like to be able to satisfy the column optimization criterion

$$\lambda_c = \max_{\phi} \frac{\alpha_c(X, \phi)}{\alpha_c(Y, \phi)}$$

as well as the row optimization criterion

$$\lambda_r = \max_{\psi} \frac{\alpha_r(X, \psi)}{\alpha_r(Y, \psi)}$$

simultaneously, but unlike the SVD, there is no mapping that takes all the solutions of the row problem to be solutions to the column problem, or vice versa. One could attempt

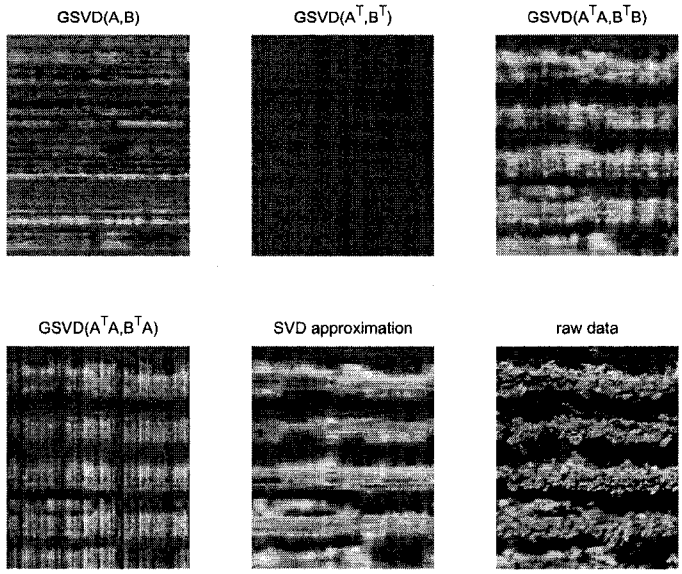


Figure 4.6: A comparison of 5 dimensional subspace representations of the 1000×500 magnified wood image.

to define a Lagrangian and balance these maximization problems, i.e., neither solution would then be optimal but would in some sense be jointly optimal. Here we consider another approach.

In general solutions to

$$\alpha X X^T \phi = \beta Y Y^T \phi$$

will not be in either of the ranges $\mathcal{R}(X)$ or $\mathcal{R}(Y)$. Thus, we may explore the possibility of solving this problem subject to the constraint, e.g., that $\phi \in \mathcal{R}(X)$. To this end, let $\phi = Xz$. Substituting this into our GSVD problem produces

$$\alpha X X^T X z = \beta Y Y^T X z$$

Applying X^T to both sides results in the column-constrained signal fraction analysis problem

$$\alpha (X^T X)^2 z = \beta X^T Y Y^T X z \tag{4.36}$$

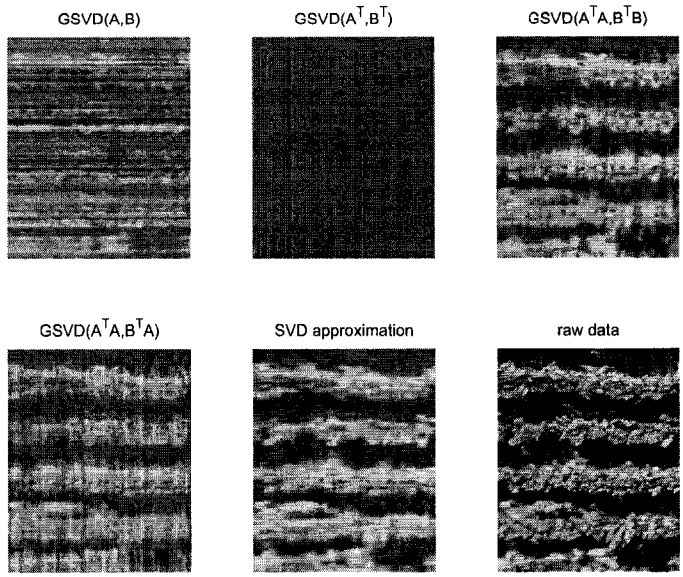


Figure 4.7: A comparison of 10 dimensional subspace representations of the 1000×500 magnified wood image.

So now, if XX^T was a large problem, then X^TX is now a small problem. Of course we could apply the same logic beginning with the row optimization problem. Also, we observe that if we let $A = X^TX$ and $B = Y^TX$ this problem is a GSVD problem of the form

$$\alpha A^T A\psi = \beta B^T B\psi.$$

We also note that if $X = Y + Q$ and $Y^TQ = 0$ then our constrained problem becomes

$$\alpha(X^TX)^2 z = \beta(Y^TY)^2 z \tag{4.37}$$

which again can be seen to be of the GSVD form with $A = X^TX$ and $B = Y^TY$. Here we adopt the shorthand notation GSVD(X^TX, Y^TY) to represent this problem statement.

To explore solutions to Equation (4.36) we consider taking the images shown in the bottom right of Figure 4.6 as X and the bottom right of Figure 4.9 as Y. Both of these

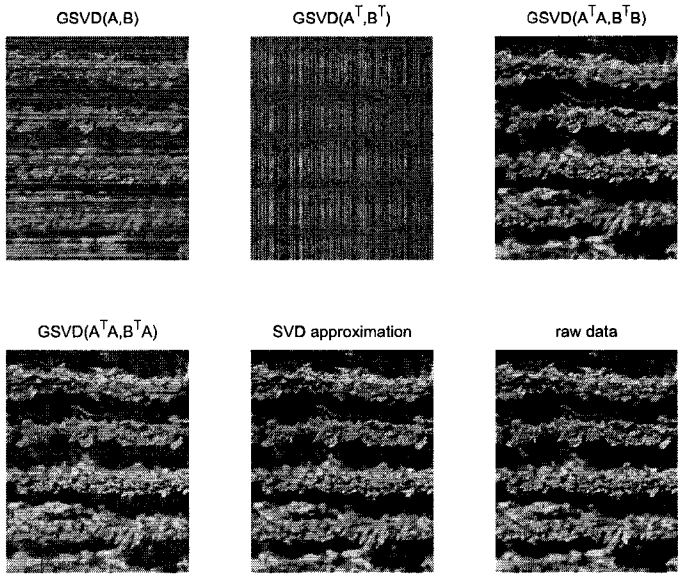


Figure 4.8: A comparison of 100 dimensional subspace representations of the 1000×500 magnified wood image.

image matrices are of size 1000×500. Using our shorthand notation above, we now propose to compare these problems.

GSVD(X, Y): Given

$$X = UCZ^T$$

and

$$Y = VSZ^T$$

we write the rank k expansions for X as

$$X_k = U_k C_k Z_k^T$$

where we sum the first k terms in the expansion. The rank k approximation for Y is similar

$$Y_k = V_k S_k Z_k^T$$

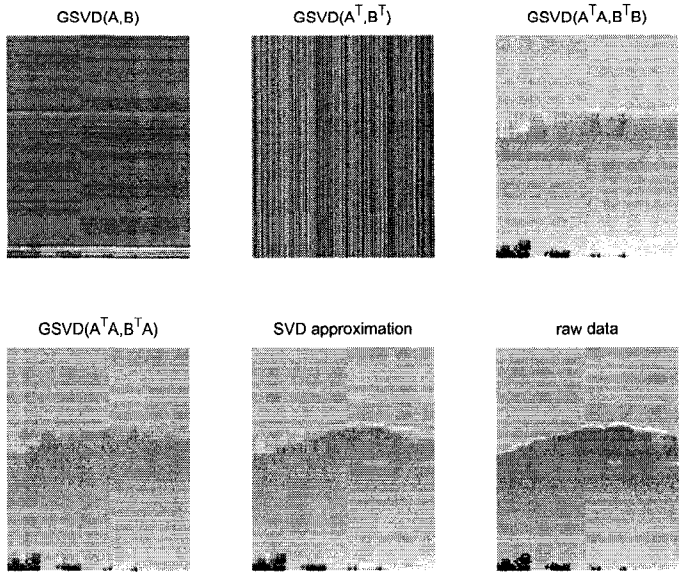


Figure 4.9: A comparison of 5 dimensional subspace representations of the 1000×500 magnified sunset image.

but now we are starting with the largest values of S and summing backwards.

GSVD(X^T, Y^T): Now we have

$$X^T = UCZ^T$$

and

$$Y^T = VSZ^T$$

where of course these are not the same U, V, C, S, Z as above. Here we write the rank k expansions for X as

$$X_k = (U_k U_k^T X^T)^T$$

where we sum the first k terms in the expansion and

$$Y_k = V_k S_k Z_k^T$$

where we are again counting from the largest s_i.

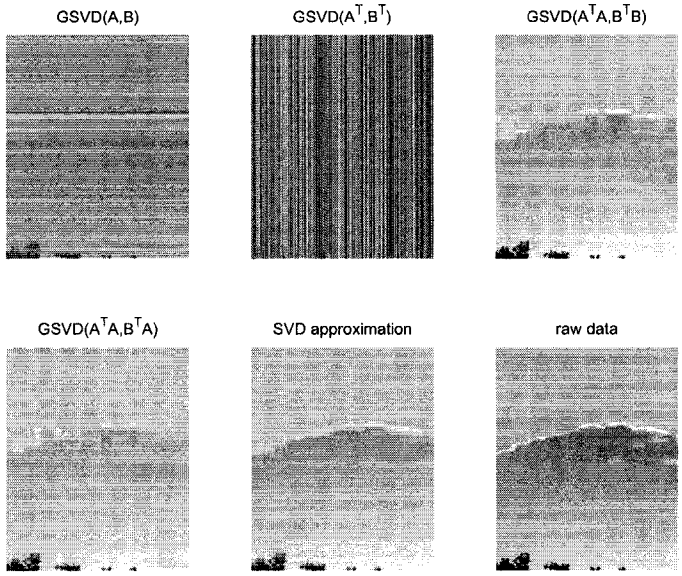


Figure 4.10: A comparison of 10 dimensional subspace representations of the 1000×500 magnified sunset image.

GSVD(X^TX, Y^TY): Now we have

$$X^TX = UCZ^T$$

and

$$Y^TY = VSZ^T$$

Our projection for X is then again

$$X_k = (U_kU_k^TX^T)^T$$

and

$$Y_k = (V_kV_k^TY^T)^T$$

GSVD(X^TX, Y^TX): This is the same as above.

In the reconstructions show in Figures 4.4-4.9 we observe several interesting features. Firstly, as one might predict, the GSVD(X, Y) compression reveals row structure and the GSVD(X^T, Y^T) thus, emphasizing the difference in these two factorizations and the associated optimization criterion. For low rank approximations we see that GSVD(X^TX, Y^TY) produces a lower error reconstruction than GSVD(X^TX, Y^TX). However, we seek the quality of GSVD(X^TX, Y^TX) surpass that of GSVD(X^TX, Y^TY) as the rank of the approximation increases .

4.5 Connection between SFA and Canonical Correlation Analysis (CCA)

Here we make a connection between Canonical Correlation Analysis (CCA) and Signal Fraction Analysis (SFA). CCA is an approach for measuring the linear relationship between two multidimensional data sets, and finds two bases, one for each data set. These bases are optimal with respect to the correlation between the two variables in the new

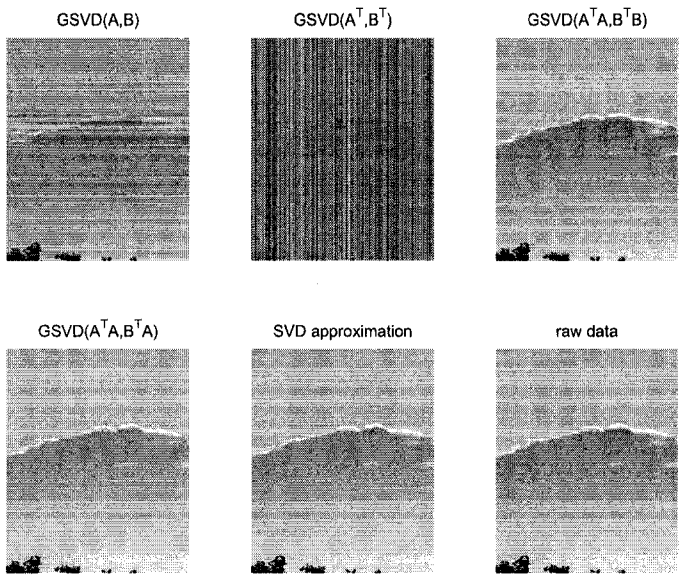


Figure 4.11: A comparison of 100 dimensional subspace representations of the 1000×500 magnified sunset image.

coordinates. Also we present an example showing that CCA can separate the mixed signals.

Suppose we have two sets of data, X and N. CCA finds two linear transformations, that maximize the correlation of X and N in the new coordinates. Let us call two sets of data $X_{n\times p}$ and $N_{n\times p}$, then consider the linear projections $y_1 = Xa$ and $y_2 = Nb$. The correlation between y_1 and y_2 is given by

$$\rho(y_1, y_2) = \frac{Ey_1y_2}{\sqrt{Ey_1{}^2Ey_2{}^2}} = \frac{a^TX^TNb}{\sqrt{a^TX^TXa}\sqrt{b^TN^TNb}}.$$

The canonical correlations can be extracted by maximizing $\rho(y_1, y_2)$

$$\max_{a,b\neq 0} \frac{a^TX^TNb}{\sqrt{a^TX^TXa}\sqrt{b^TN^TNb}}. \tag{4.38}$$

Taking the derivative of the correlation with respect to a and b and setting them to zero respectively leads to

$$N^TNb = \frac{a^TN^TNb}{a^TN^TNa}N^TNa \tag{4.39}$$

$$N^TNa = \frac{a^TN^TNb}{b^TX^TXb}X^TXb \tag{4.40}$$

If we substitute equation (4.40) in (4.39) we get

$$N^TNb = \frac{a^TN^TNb}{a^TN^TNa}\frac{a^TN^TNb}{b^TX^TXb}X^TXb \tag{4.41}$$

Now define

$$1/\lambda = \frac{a^TN^TNb}{a^TN^TNa}\frac{a^TN^TNb}{b^TX^TXb} \tag{4.42}$$

which leads to

$$X^TXb = \lambda N^TNb \tag{4.43}$$

Therefore as we see CCA and SFA are related if in CCA we suppose our two data sets are X (signal) and N (noise).

To illustrate this, we have used the four independent signals (please see Figure 4.12), we mixed them randomly (please see Figure 4.13), and we extracted the signals via CCA (Figure 4.14), and SFA methods (Figure 4.15).

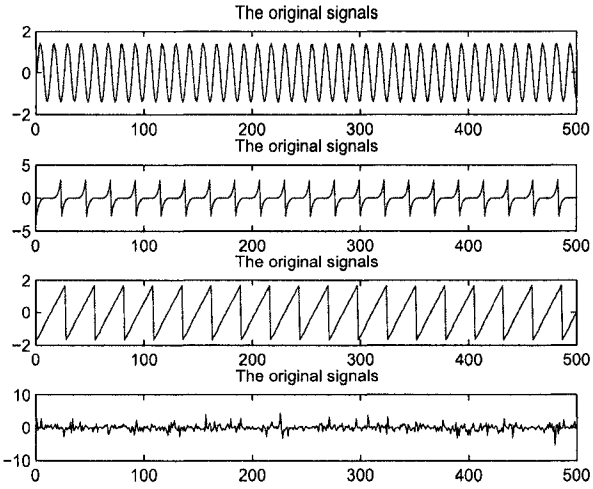


Figure 4.12: All four original signals.

We calculated the correlation between the extracted signals via applying the SFA and CCA. It appears that these estimations are almost the same (as the theory says that as well). The correlation between the original signals and the extracted signals was 0.9999 which is almost one. Note that the order of the extracted signals from CCA and SFA were the same. The spatial distributions of the extracted signals via CCA are very similar to those extracted by the SFA method.

4.6 Summary of Contributions

In this chapter we presented Signal Fraction Analysis, a general tool for computing subspaces for decomposing data into potentially useful features. We presented the row-energy and column-energy optimization problems for signal-to-signal ratios, derived the resulting generalized singular value problem and distinguish this setting from the standard SVD. We proposed that preprocessing mappings of the data be used in situations where domain specific knowledge is available as a guide. More generally, we suggest an optimization problem where these mapping functions may be adapted using a problem dependent objective function. We illustrate these ideas using Wavelet and Fourier fil-

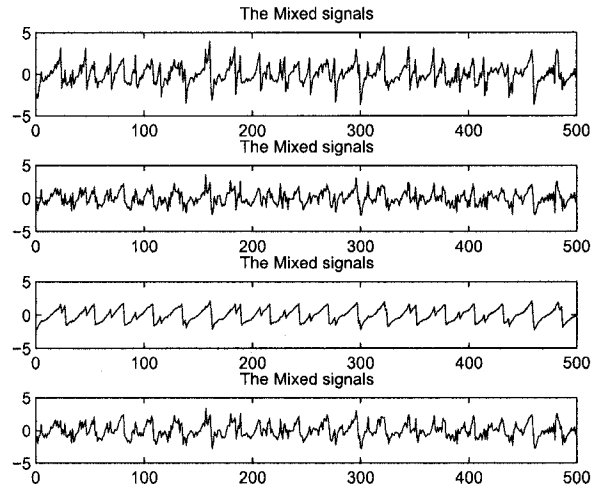


Figure 4.13: The mixed signals.

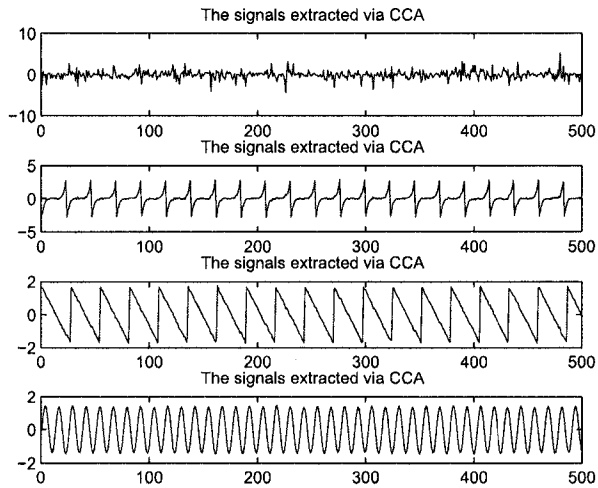


Figure 4.14: Extracted signals via CCA.

ters applied to EEG data. We present a self-contained description of the motivating maximum noise fraction method and describe a procedure for estimating the covariance matrix of the noise. We extend Signal Fraction Analysis by introducing novel constraints and solving them. We propose two new GSVD type problems for computing subspace representations. Finally, we draw a connection between Signal Fraction Analysis and Canonical Correlation Analysis.

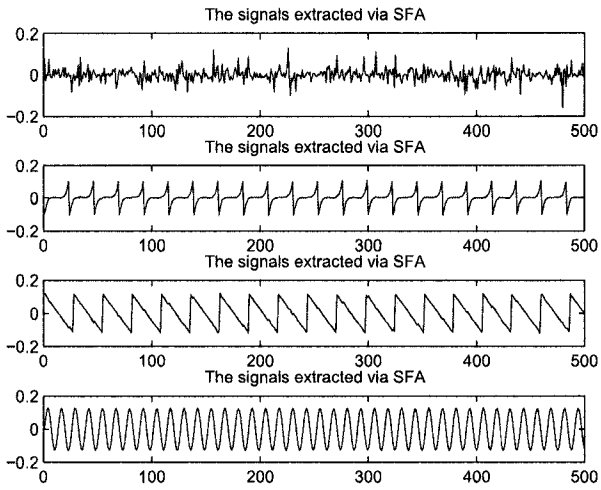


Figure 4.15: Extracted signals via SFA.

Chapter 5

KERNEL SIGNAL FRACTION ANALYSIS

An important chapter in the history of empirical data analysis began with the introduction of the idea of a support vector machine (SVM) for pattern classification [98, 99, 100]. SVMs effectively converted an important linear classification algorithm based on the computation of dot products to an effective nonlinear algorithm. Motivated by these ideas, other algorithms for empirical data analysis based on dot products have also been extended to nonlinear versions. Of particular relevance to this work is the extension of the basic principal component analysis (PCA) algorithm described in Chapter 2 to a nonlinear, or Kernel PCA (KPCA) algorithm. Since we view Signal Fraction Analysis (SFA) to be a variation of PCA focused on splitting subspaces associated with two data sets, it is natural to consider the impact of employing the SVM like non-linearization idea to SFA. We note that now a large number of effectively linear algorithms have been extended nonlinearly using the SVM suite of ideas, including Kernel Fisher Discriminant Analysis (KFDA) [70, 44], Linear Discriminant Analysis (LDA) [110], kernel Gram-Schmidt [69], kernel canonical correlation analysis, and Kernel Partial Least Square (KPLS) [10, 1, 18, 65]. All these kernel-based methods have permitted researchers to provide additional insight into problems where strictly linear transformations are inadequate. Based on these successes we are motivated to extend SFA to kernel SFA, or KSFA, in a fashion to be described below.

In this chapter, we present an overview of some of the existing and relevant kernel methods, and we propose a new method that performs nonlinear Signal Fraction Analysis

(SFA) or kernel based SFA useful for the feature extraction and data reduction in case of nonlinearly separable data sets and with the help of toy examples we compare kernel SFA (KSFA) and KPCA, and we show the advantage of applying KSFA with respect to KPCA. In Chapter 7 we see that KSFA can also outperform KPCA on the Brain Computer Interface (BCI) Problem.

The outline of this chapter is as follows: In Section 5.1 we present some of the most commonly used kernel methods; to motivate our work as well as to provide a basis for comparison, in Section 5.2 we outline how PCA generalizes to KPCA. In Section 5.3 we generalize SFA to kernel signal fraction analysis, or KSFA. We propose two methods to solve the GSVD problem in KSFA in Sections 5.3.1 and 5.3.2. In Section 5.4 we illustrate some advantages of KSFA over KPCA on a toy problem.

5.1 Overview of Kernel Methods

When the data is inherently nonlinear, e.g., when the relationship between two data sets cannot be modeled by linear mappings, it is challenging to obtain a satisfactory analysis of the data by directly applying linear schemes (here we refer to the linear methods like PCA in Section 2.2 and SFA in Section 4.3). The most obvious example is the failure of linear discriminant analysis to effectively classify data that is nonlinearly separable. In this situation a kernel based extension to such a linear method is highly appropriate. Thus we are motivated to generalize these linear methods to nonlinear versions where possible.

We illustrate the potential of kernel methods via a now well-known example. Consider the data shown in Figure 5.1. The circles label points associated with class I and the crosses label points associate with class II. No single straight line can partition the data into its two separate classes, hence it is not linearly separable, i.e., it is nonlinearly separable. However, if we apply the nonlinear map

$$(x_1, x_2) \in \mathbf{R}^2 \Longrightarrow (x_1, x_2, x_1^2 + x_2^2) \in \mathbf{R}^3 \tag{5.1}$$

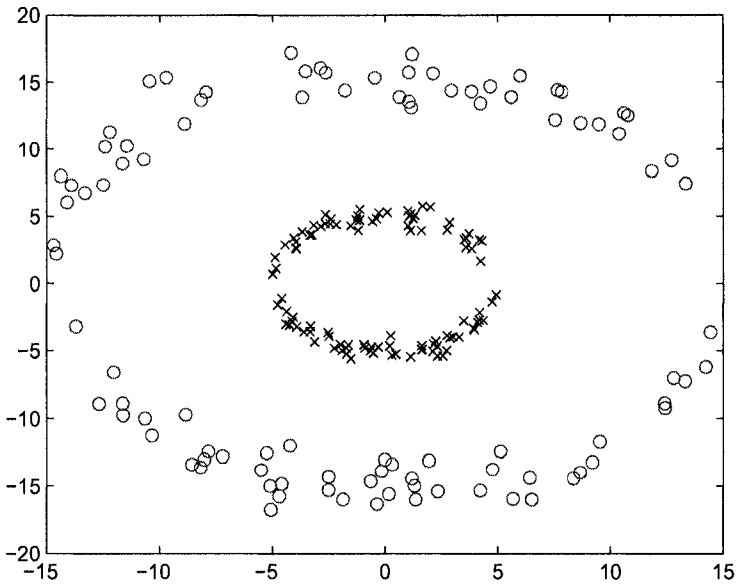


Figure 5.1: Nonlinear separable data. Class I points are labeled with circles and class II points are labeled with crosses. Note that no single line can partition the classes.

the data becomes linearly separable in $\mathbf{R}^3$ and now it can be separated via linear hyperplane; see Figure 5.2. The image space of this mapping is often referred to as a *feature space* since the components of the data now possess features (or coordinate values) that allow it to be separated. Obviously, such feature spaces are treated with a certain amount of awe.

More generally, we can propose the existence of a mapping function

$$\phi : x \in \mathbb{R}^n \rightarrow \phi(x) \in \mathbb{R}^m \tag{5.2}$$

of which Equation (5.1) is a special example. More generally, we will be interested in the Veronese mapping of degree d which takes an n-tuple $(x_1, \ldots, x_n)$ to all monomials of degree d [69]. The dimension of the feature space under that action of the Veronese mapping is

$$m = \frac{(d+n)!}{d!n!} - 1$$

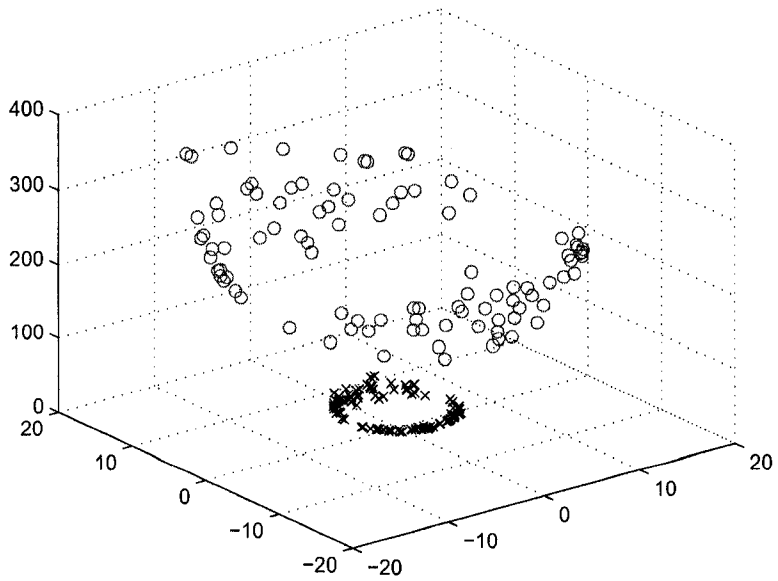


Figure 5.2: The result of mapping the data in Figure 5.1 using Equation (5.1). Notice that now the data may be separated by an appropriately placed plane.

So, for example, if we are looking at data initially in 10 dimensions, and seek a Veronese mapping of degree 5 the dimension of the feature space has 3002 dimensions. Given this explosion of dimension this method would seem to have its shortcomings. However, for the Veronese mapping (as well as other mapping functions) there is a *kernel trick* which permits the inexpensive computation of dot products in the image space of the mapping ϕ. Specifically,

$$k(x, y) = (x^T y)^d \tag{5.3}$$

For example the polynomial kernel of degree two for $x = (x_1, x_2)$ and $z = (z_1, z_2)$ can be calculated via the following:

$$\begin{aligned}(x \cdot z)^2 &= (x_1 z_1 + x_2 z_2)^2 && (5.4)\\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 x_2 z_2 && (5.5)\\ &= ((x_1^2, x_2^2, \sqrt{2x_1 x_2}), (z_1^2, z_2^2, \sqrt{2z_1 z_2})) && (5.6)\\ &= (\phi(x), \phi(z)) && (5.7)\end{aligned}$$

So this is a scaled version of the Veronese mapping.

The consequence of this observation is profound. Essentially it means that any algorithm which is based on the computation of dot products only, may be implemented in very high (even infinite) dimensions while invoking only very inexpensive computations. So while we envision the action of the general mapping function as illustrated in Figure 5.3, these points in the image space $\phi(x)$ need never actually be computed. Only the dot products are computed via the kernel function, i.e.,

$$k(\phi(x), \phi(y)) = \phi(x)^T \phi(y) \tag{5.8}$$

In pattern recognition we need to have some kind of similarity measurement to be able to study new data points and compare patterns. Dot products are known to be a good type of similarity measurements [28]. Computing the dot product lets us carry out all the geometrical constructions based on angles, lengths and distances. Not surprisingly, many interesting algorithms in pattern analysis only require the computation of dot products.

In addition to the Veronese mapping described above, other well-known and commonly used kernel functions are Gaussian radial basis functions (GRBF),

$$K(x_1, x_2) = e^{-(\|x_1 - x_2\|^2)} \tag{5.9}$$

and neural network kernels

$$K(x_1, x_2) = \tanh(a(x_1 \cdot x_2) + b), \quad a, b \geq 0 \tag{5.10}$$

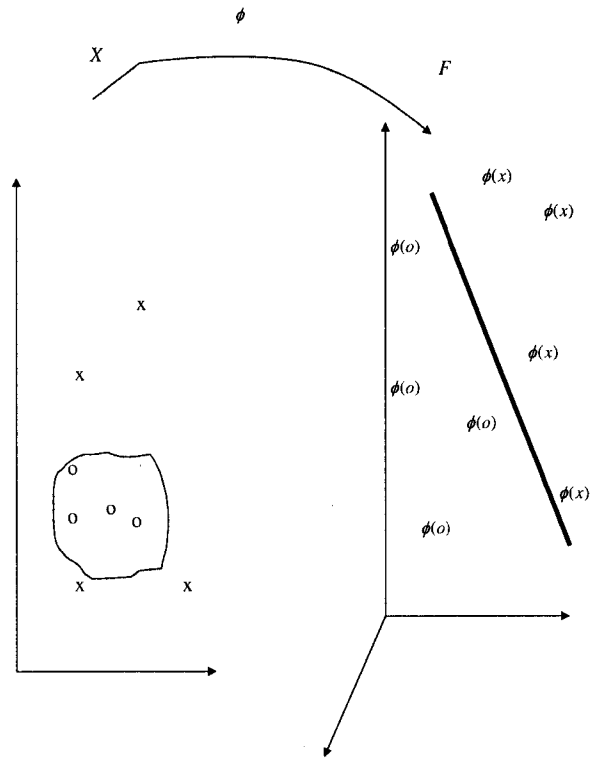


Figure 5.3: Separation of nonlinear mixed data in input space via mapping function.

For more information about the neural network kernel one might find [20] useful. Thus, in summary, a kernel function provides us with a high-dimensional similarity measurement [28].

The idea of using a kernel function has been introduced in the field of support vector machines (SVM) [99]. Support vector machines use a particular type of function induced by a kernel and they are powerful tools for data classification. Classifying data is achieved by a linear or nonlinear hyperplane in the input space [99, 23, 17, 19, 67]. Briefly speaking, SVMs are linear learning machines that produce a generalization from input space to high dimensional feature space. Indeed, we can think of SVM as a linear algorithm in high dimensional space (feature space) which does not involve any computations in that high dimension; however, all computations to compute the separating hyper plane can be performed directly in input space by the use of kernel functions without the need of knowing the nonlinear map into feature space (see Equations (5.4) - (5.7)).

Apparently, Aizerman was apparently the first to talk about kernel trick [2]. It is a widely used tool today.

5.2 Kernel Principal Component Analysis (KPCA)

Principal component analysis is based on dot product computations and it closely related to singular value decomposition (see Section 2.1). In Section 2.1 we showed the relationship between SVD and eigenvalue eigenvector problem and since in PCA we assume centered data set then the right singular vectors $v_k \in V$ are the same as the principal components of the data set x_i's and the eigenvalues of $\Sigma = X^T X$ are the same as square of the singular values.

PCA is a subspace method that can detect the linear structure of the data. However, if the data has a nonlinear structure (e.g., quadratic forms) it is well known that PCA is non-optimal. When working with the data set that has nonlinear features it has been proposed that to draw more accurate conclusions we should explore kernel PCA,

or KPCA [71]. KPCA is generated via mapping the input space (original data set) to feature space (higher dimension space than the input space) and using the kernel trick.

Let $X \in \mathbf{R}^{n\times p}$, $n > p$ be a multivariate centered data set and $X^T = [x_1, x_2, \ldots, x_n]$. PCA diagonalizes the covariance matrix of the data. If we denote the covariance of the data by $\Sigma = X^T X$, then its eigenvalue-eigenvector decomposition is

$$\Sigma v_i = \lambda_i v_i \qquad i = 1, \ldots, p \tag{5.11}$$

Where $(\lambda_i, v_i), \quad i = 1, \ldots, p$ is the i-th eigenvalue eigenvector pair for Σ . Via some manipulation we could show that the eigenvectors $v_1, v_2, \ldots, v_p$ are the linear combinations of the data $x_1, x_2, \ldots, x_n$

$$v_i = \sum_{j=1}^{n} \theta_{(i,j)} x_j \qquad i = 1, \ldots, p \quad j = 1, \ldots, n \tag{5.12}$$

Where,

$$\theta_{(i,j)} = (x_j, v_i)/n\lambda_i \tag{5.13}$$

The eigenvectors represent a set of uncorrelated variables that determines the directions of maximum variability in the data which is called principle components, and eigenvalues show the amount of information provided by the respective principal component [71]. In most of the time we work with the largest eigenvector (eigenvector that corresponds to the largest eigenvalue) to project the data.

Now we propose to illustrate the comparison between PCA and KPCA in feature space following [86].

Given mean subtracted (centered) data set $X \in \mathbf{R}^{n\times p}$, $n > p$ the covariance matrix of the data can be calculated via

$$\Sigma = 1/n \sum_{i=1}^{n} x_i x_i^T = 1/n X^T X \tag{5.14}$$

Since PCA diagonalizes the covariance matrix Σ then we have to solve the eigenvalue eigenvector equation

$$\lambda v = \Sigma v \tag{5.15}$$

In Equation (5.15) λ is the eigenvalue and $v \in \mathbf{R}^p$ is the eigenvector for the covariance matrix. If we substitute Equation (5.14) in Equation (5.15) we get

$$\lambda v = \Sigma v = 1/n \sum_{i=1}^{n} x_i x_i^T v \tag{5.16}$$

if we left multiply Equation (5.16) by x_j^T for $j = 1, \ldots, n$ we get

$$\lambda x_j^T v = 1/n \sum_{i=1}^{n} x_j^T x_i x_i^T v \tag{5.17}$$

If we use the dot product notation, (i.e., $(x.y) = x^T y$), Equation (5.17) leads to a more simplified form like

$$\lambda(x_j \cdot v) = (x_j \cdot \Sigma v) \quad for \quad j = 1, \ldots, n \tag{5.18}$$

We formulated the PCA in a way which uses dot product. Equation (5.18) is based on dot product which makes it possible to use the kernel methods for generalizing the idea (the outer product formulation of the covariance matrix does not lend itself directly to being kernelized). At this point we assume that the data has been mapped to a higher dimension that is called feature space (i.e., $\mathbf{F}$) via nonlinear mapping ϕ

$$\phi : \mathbf{R}^p \longrightarrow \mathbf{F} \tag{5.19}$$

although we emphasize that this mapping actually is never computed. We should again assume that the new mapped data set $\phi(x_i)$ is centered i.e., $\sum_{i=1}^{n} \phi(x_i) = 0$ (more on centering in high dimension space can be find in [86]) therefore the covariance matrix of the new data set equals

$$\overline{\Sigma} = 1/n \sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T \tag{5.20}$$

At this point we follow our earlier formulation of PCA and find the eigenvalue and eigenvectors for the new covariance matrix of the mapped data set

$$\lambda V = \overline{\Sigma} V \tag{5.21}$$

If we pre-multiply Equation (5.21) by $\phi(x_j)$ and do the same simplification as we did before for PCA

$$\lambda(\phi(x_j) \cdot V) = (\phi(x_j) \cdot \Sigma V) \quad for \quad j = 1, \ldots, n \tag{5.22}$$

From the fact that all eigenvectors V; correspond to nonzero eigenvalue λ; lie in the span of the data $\phi(x_i) \quad i = 1, \ldots, p$ we get [86]

$$V = \sum_{i=1}^{n} \alpha_i \phi(x_i) \quad i = 1, \ldots, n \tag{5.23}$$

If we combine Equations (5.18) and (5.23), we get

$$\lambda \sum_{i=1}^{n} \alpha_i (\phi(x_j) \cdot \phi(x_i)) = 1/n \sum_{i=1}^{n} \alpha_i (\phi(x_j) \cdot \sum_{k=1}^{n} \phi(x_k))(\phi(x_k) \cdot \phi(x_i)) \tag{5.24}$$

for all $\quad j = 1, \ldots, n.$

A kernel function $K_{i,j}$ projects x_i and x_j in input space into a higher space which we call feature space. For example if the mapping function from input space to feature space is ϕ, then

$$K_{i,j} = \phi(x_i)^T \phi(x_j) \tag{5.25}$$

Defining a matrix called kernel matrix K with the elements $K_{i,j}$ and substituting into Equation (5.24) we get

$$n\lambda K\alpha = K^2\alpha \tag{5.26}$$

Here, α is a column vector of α_i's. Equation (5.26) is the same as

$$n\lambda\alpha = K\alpha \tag{5.27}$$

In Equation (5.27), the relationship between KPCA and dot product is clear.

Throughout this work we have used polynomial kernels of degree d as defined in Equation (5.3). In the toy examples that follow we considered values of d from one to four. In the simulations we use contour lines which connect points with the same feature value. They are lines of constant projections of the test point onto the principal components. The shape of the contour lines depends on the distribution of the data and the kernel function.

5.3 Kernel Signal Fraction Analysis (KSFA)

In this section we propose to extend SFA in a manner analogous to the kernel PCA. Similarly, we refer to the new method as KSFA. In this section we propose two ways to solve the GSVD problem corresponding to KSFA. First, we propose a direct method to solve GSVD. The second method pre-processes the data to show that the KSFA problem can be solved using KPCA. Toy examples in this section suggest the promise of KSFA when compared to KPCA on a noisy data set. We present an application of KSFA to BCI in Chapter 7.

5.3.1 Direct KSFA

In this derivation we have assumed that we have both data matrices A and B in SFA estimated. Since we first propose to apply KSFA to a nonlinear function with additive noise we will assume that the matrix A consists of the data matrix X and that the matrix B is a high pass filter of the data which we denote dX. As we learned from section (4.3.2) the resulting matrix $dX^T dX$ is an estimate for the covariance matrix of the noise; see also (4.3.2). Therefore, we use Equation (5.28)

$$N^H N = dX^H dX/2 \tag{5.28}$$

as the estimate for the noise covariance matrix and our generalized singular value decompositions turns to be like Equation (5.29)

$$s_i^2 X^T X a = c_i^2 dX^T dX a \tag{5.29}$$

and in cases where s_i is not zero

$$X^T X a = (\lambda/2) dX^T dX a \tag{5.30}$$

Where,

$$c_i^2/s_i^2 = \lambda/2 \tag{5.31}$$

Solving Equation (5.30), leads to find the generalized singular vectors (a's) and generalized singular values ($\lambda/2$) and provides the estimation of the desired signal (or we could say it separates signal from noise). Thus, if

$$X = [x_1, x_2, \ldots, x_p] \tag{5.32}$$

and

$$dX = [dx_1, dx_2, \ldots, dx_p] \tag{5.33}$$

we map Equations (5.32) and (5.33) to feature space by

$$\phi(X) = [\phi(x_1), \phi(x_2), \ldots, \phi(x_F)] \tag{5.34}$$

and

$$d\phi(X) = [d\phi(x_1), d\phi(x_2), \ldots, d\phi(x_F)] \tag{5.35}$$

Therefore, Equation (5.30) becomes

$$\phi(X)^T \phi(X) b = (\tau/2) d\phi(X)^T d\phi(X) b \tag{5.36}$$

Now, defining

$$\begin{cases} K = \phi(X)^T \phi(X) \\ dK = d\phi(X)^T d\phi(X) \\ \eta = \tau/2 \\ b = \sum \beta_i \phi(x_i) \end{cases} \tag{5.37}$$

the $(i,j) - th$ component of matrix K and dK can be written as

$$k_{ij} = (\phi(x_i) \cdot \phi(x_j)) \tag{5.38}$$

and

$$dk_{ij} = (d\phi(x_i) \cdot d\phi(x_j)) \tag{5.39}$$

Now substituting Equations (5.37), (5.38), (5.39) in (5.36) and pre-multiply both sides by $\phi(x_k)$, we get

$$\begin{aligned} &(\phi(x_k) \cdot \sum \phi(x_i)\phi(x_j) \sum \beta_l \phi(x_l)) = \\ &\eta(\phi(x_k) \cdot \sum d\phi(x_i)d\phi(x_j) \sum \beta_l \phi(x_l)) \end{aligned} \tag{5.40}$$

Equation (5.40) in matrix form is

$$K^2\beta = \eta K dK \beta \tag{5.41}$$

Equation (5.41) is the same as

$$K\beta = \eta dK \beta \tag{5.42}$$

The above process explains our proposed kernel based SFA (KSFA).

5.3.2 KSFA via Noise Adjustment for SFA

In this section, we first adjust or remove the noise before applying the kernel function to SFA method [74, 58, 68].

From Section 4.3 we know to apply SFA transformation one should solve the GSVD problem that is $X^T X a = \lambda N^T N a$. Lets denote $\Sigma = X^T X$, and $\Sigma_N = N^T N$; therefore, the GSVD problem simplifies to

$$\Sigma a = \lambda \Sigma_N a \tag{5.43}$$

We first find the eigenvalue matrix (B), and eigenvector matrix (A) for the noise covariance matrix $\Sigma_N = N^T N$. Thus, we have

$$A^T \Sigma_N A = B, \quad A^T A = I \tag{5.44}$$

Then we need to transform our data to a new set of data with the identity matrix as its noise covariance matrix. For this we calculate

$$\tilde{A} = AB^{-1/2} \tag{5.45}$$

Thus,

$$\tilde{A}^T \Sigma_N \tilde{A} = I, \quad \tilde{A}^T \tilde{A} = B^{-1} \tag{5.46}$$

Now, if we pre-multiply Equation (5.43) by $\tilde{A}^T$ and do the change of bases for a via $a = \tilde{A}\tilde{a}$, we get

$$\tilde{A}^T \Sigma \tilde{A} \tilde{a} = \lambda (\tilde{A}^T \Sigma_N \tilde{A}) \tilde{a} \tag{5.47}$$

We can simplify the right hand side of Equation (5.46) using Equation (5.47). Hence,

$$\tilde{A}^T \Sigma \tilde{A} \tilde{a} = \lambda \tilde{a} \tag{5.48}$$

Now considering the data transformation as

$$\tilde{X} = \tilde{A}^T X \tag{5.49}$$

and denoting

$$\tilde{\Sigma} = \tilde{X}^T \tilde{X} \tag{5.50}$$

we have

$$\tilde{\Sigma} \tilde{a} = \lambda \tilde{a} \tag{5.51}$$

Therefore, to apply KSFA first we could adjust the noise and then apply the KPCA on the transformed data set and find the eigenvalue, eigenvector for the covariance matrix of the transformed data set and substitute it in $a = \tilde{A}\tilde{a}$ to find the weights.

Now we formulate and summarize KSFA method based on adjusting the noise. First, we estimate the covariance of the noise if it is unknown. Then we find its eigenvalue matrix B, and eigenvector matrix A to calculate the transformation matrix $\tilde{A}$

$$\tilde{A} = AB^{-1/2} \tag{5.52}$$

We transform the original data X to a new data set $\tilde{A}$ as

$$\tilde{X} = \tilde{A}^T X \tag{5.53}$$

Finally, we apply KPCA to this transformed data. This process provides us a KSFA via adjusting noise.

5.4 Toy Examples

In this section we use toy examples and apply KPCA and KSFA and we compare these two techniques and we show the advantage of applying KSFA. Our first technique is KSFA via applying noise adjustment (i.e., whitening the covariance of the noise as pre-processing) that we call it KSFA via noise adjustment. To compare the performance of KPCA and KSFA we provide a toy example and use polynomial kernels from degree one to degree four for both cases KPCA and KSFA in Section 5.4.1. We generated the contour lines of the constant component values which connect points with the same feature. Then in Section 5.4.2 we have provided another toy example with sinusoidal property.

5.4.1 Quadratic Toy Example

In [86], the authors have shown that for the toy example

$$y = x^2 + N,$$

KPCA is capable of extracting interesting features from the data. Here we show that for the same toy example KSFA extracts different data features and produces a lower reconstruction error than KPCA. We select the x values to have a uniform distribution on the interval $[-1, 1]$ and the y values to be the squared power of x plus a centered normal noise with standard deviation equal to 0.2. Therefore, we have

$$y = x^2 + \varepsilon \tag{5.54}$$

where

$$\varepsilon \sim N(0, 0.2^2) \tag{5.55}$$

We consider 50 data points. In our work, we ran set of experiments using polynomial kernels of degree one to four. The kernel function used in our simulation is

$$K(x, y) = (x \cdot y)^d \quad for \quad d = 1, 2, 3, 4. \tag{5.56}$$

Figure 5.4 shows the KPCA performance with kernels of degree one (first column in the left) to degree four (last column in the right). It is clear that the first column on the left is like performing a PCA in input space and shows two eigenvalues. Since kernelizing is mapping nonlinearly the input space to feature space, the contour lines of constant projections on the principal components are nonlinear for kernels of degree two and more. The contour lines show the structure of the data. The last row with low eigenvalues are related to the noise subspace and shows the amount of information that can be used considering that principal component and since the related eigenvalue is small we conclude that there is not much information here.

We have used the same toy example to compare KPCA and KSFA via noise adjustment performance. One important characteristic of KPCA, like PCA, is data reduction. Figures 5.4 and 5.5 show that we even could do further dimension reduction via applying KSFA via noise adjustment over applying KPCA. It is because in Figure 5.5 the

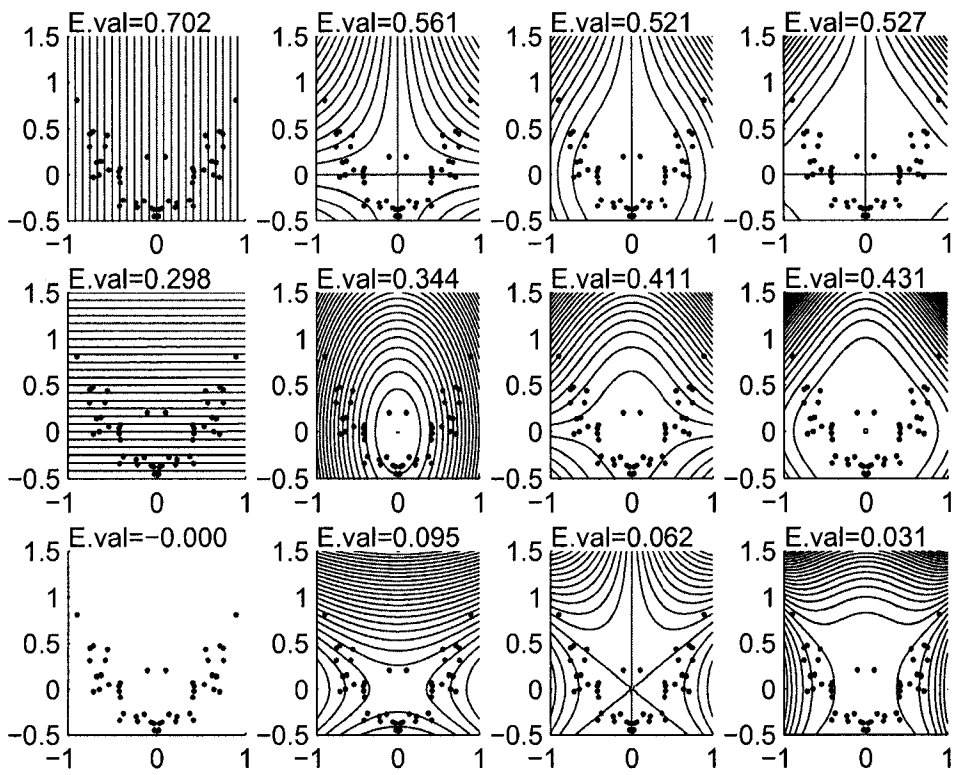


Figure 5.4: Toy example for performing KPCA. From left to right the degree of the polynomial kernel increases from one to four. From top to bottom the contour lines of constant principal components associated to decreasing order of eigenvalues are shown.

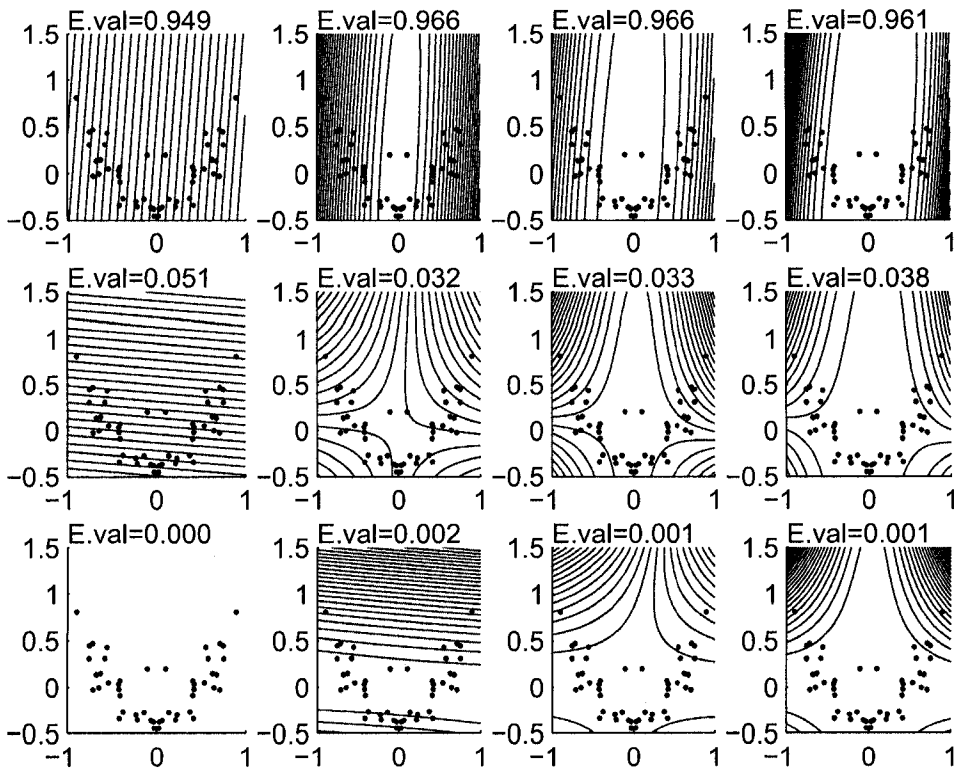


Figure 5.5: Toy example for performing KSFA. From left to right the degree of the polynomial kernel increases from one to four. From top to bottom the contour lines of constant maximum signal fraction associated to decreasing order of eigenvalues are shown.

eigenvalues in the second and third row are very small showing that these components of maximum signal fraction consist of more noise than signal (if we call the data as our signal). Hence, ignoring the second and third component of maximum signal fraction does not cause missing much of information. Thus it suggests that just the first maximum signal fraction component would be enough to account for most of the information live in the original data set. However, in KPCA case we need to use the first two components to be able to use as much information as possible. These points result in further reduction via applying KSFA. Therefore, the simulation shows that applying the KSFA has potential advantages for separating signals (or signal and noise) over SFA. This result very interesting, but possibly not too surprising given SFA's performance compared to PCA's performance on the signal separation problem.

We also kernelized the SFA method directly without preprocessing the data and we did the simulations on toy examples to compare the performance of SFA and KSFA. In this case, we use the same toy example introduced in Section 5.4 , and we apply it to KPCA and KSFA and we compare these two techniques for polynomial kernel functions of degrees one to four. In our toy example, standard PCA and standard SFA lead to two nonzero eigenvalues and generalized eigenvalues, respectively (first left column in Figures 5.6 and 5.7. However, nonlinear PCA and SFA allow more features (PCA components and SFA components) to be extracted.

In Figure 5.6, nonlinear PCA contains contour lines of constant feature values showing the structure hidden in the data better than in linear PCA case. We see that the last components show almost the same behavior for different polynomial degrees. It seems that the last principal component associated to the smallest eigenvalue picks up the variance caused by the noise (this is more obvious in the case of polynomial of degree two; see the second column in Figure 5.6). This means, by ignoring the last component we can reduce the noise and do the noise reduction.

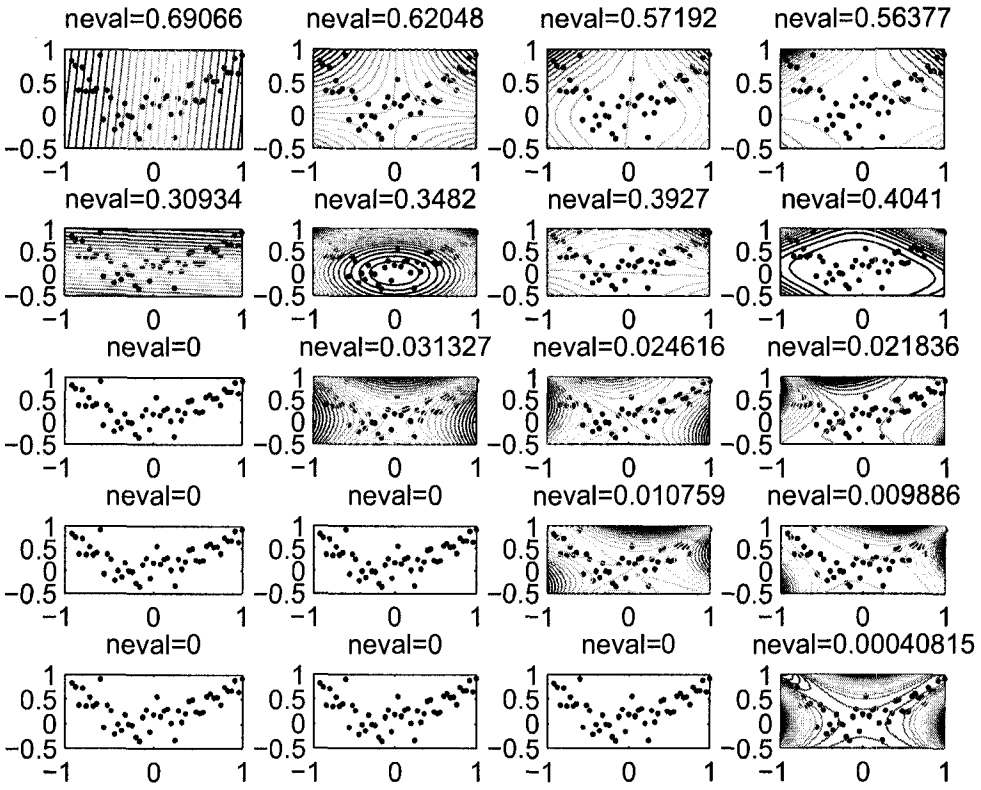


Figure 5.6: Toy example for performing KPCA. From left to right the degree of the polynomial kernel increases from one to four. From top to bottom the contour lines of constant principal components associated to decreasing order of eigenvalues are shown.

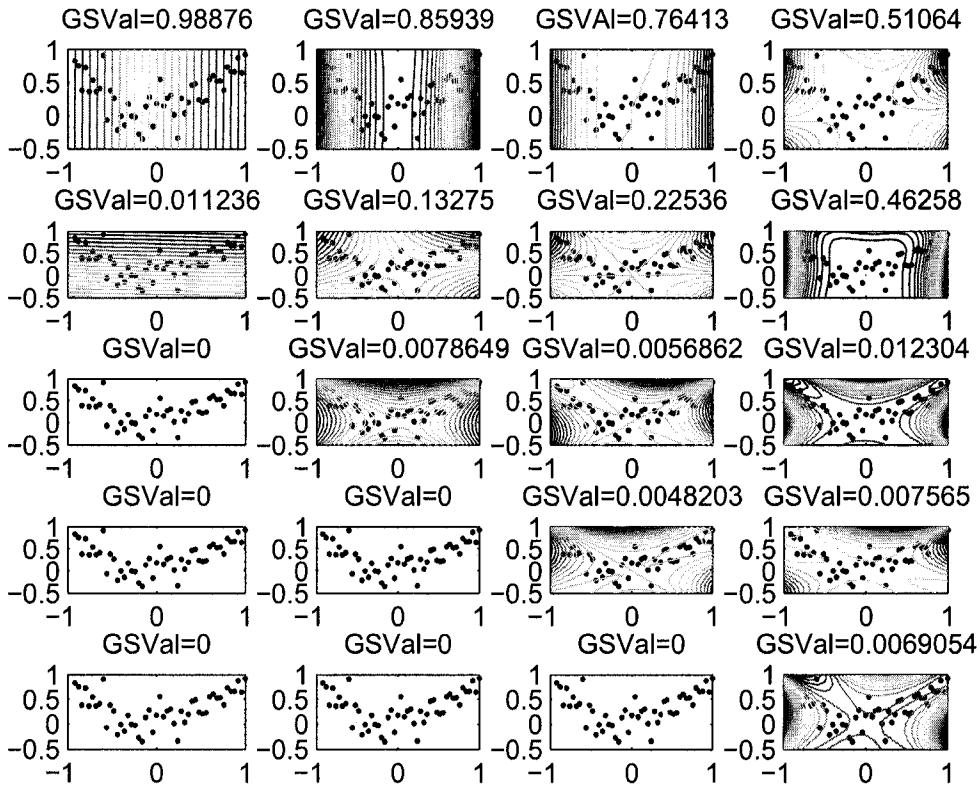


Figure 5.7: Toy example for performing KSFA. From left to right the degree of the polynomial kernel increases from one to four. From top to bottom the contour lines of constant SFA components associated to decreasing order of generalized eigenvalues are shown.

In Figure 5.7, KSFA contains contour lines of constant feature values showing the structure hidden in the data better than in linear SFA case. Here again we see that the last SFA components have the same behavior for different polynomial degrees, even their behavior is similar to the last components in PCA case. These results, on this example, suggest the potential for noise reduction in both SFA and PCA and we can obviously look at the last components in both Figures 5.6 and 5.7 which are associated with small eigenvalues and generalized eigenvalues as components that can be ignored to do the noise reduction (or we can connect them to noise subspace extraction).

If we compare the results in Figures 5.6 and 5.7 we see that applying KSFA could reduce the noise better using fewer components. It is because first SFA components in each polynomial degree contain more signal than noise which is not the case in KPCA. Note that, PCA finds the direction of maximum variance in the data which is mixed with noise; however, SFA finds the direction of maximum variance in signal and minimum variance in noise meaning that SFA picks directions of mostly just signals.

The results show that both KPCA and KSFA are subspace methods that apparently can separate the signal subspace and noise subspace, and both methods extract the structure of the data. However, comparing Figures 5.6 and 5.7 and looking at the eigenvalues and generalized eigenvalues for both methods suggests that KSFA we can do a better job in feature extraction.

Looking at the second column in Figures 5.6 and 5.7 shows that two of the contours are essentially the same while in KPCA case we see a set of concentric circles for basis vector two and in contrast KSFA case shows almost straight lines. These become clear in Figures 5.8 and 5.9 (the difference is that one is a parabola and one is the bowl). To see the results in three dimensions we also generated the figures for a better way of visualizing the problem (Figures 5.8 and 5.9). We viewed the results for another run with the three dimensional figures to be able to further interpret the behavior of KPCA and KSFA.

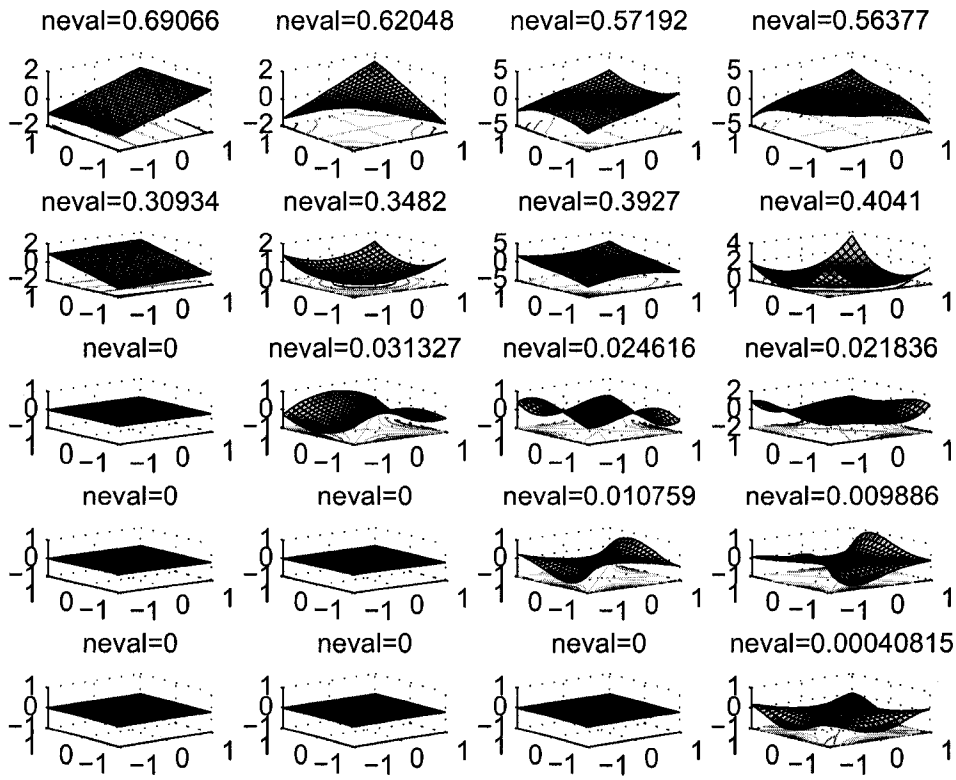


Figure 5.8: Toy example for performing KPCA in 3-D. From left to right the degree of the polynomial kernel increases from one to four. From top to bottom the contour lines of constant principal components associated to decreasing order of eigenvalues and the hyper-plane that captures the structure of the data are shown.

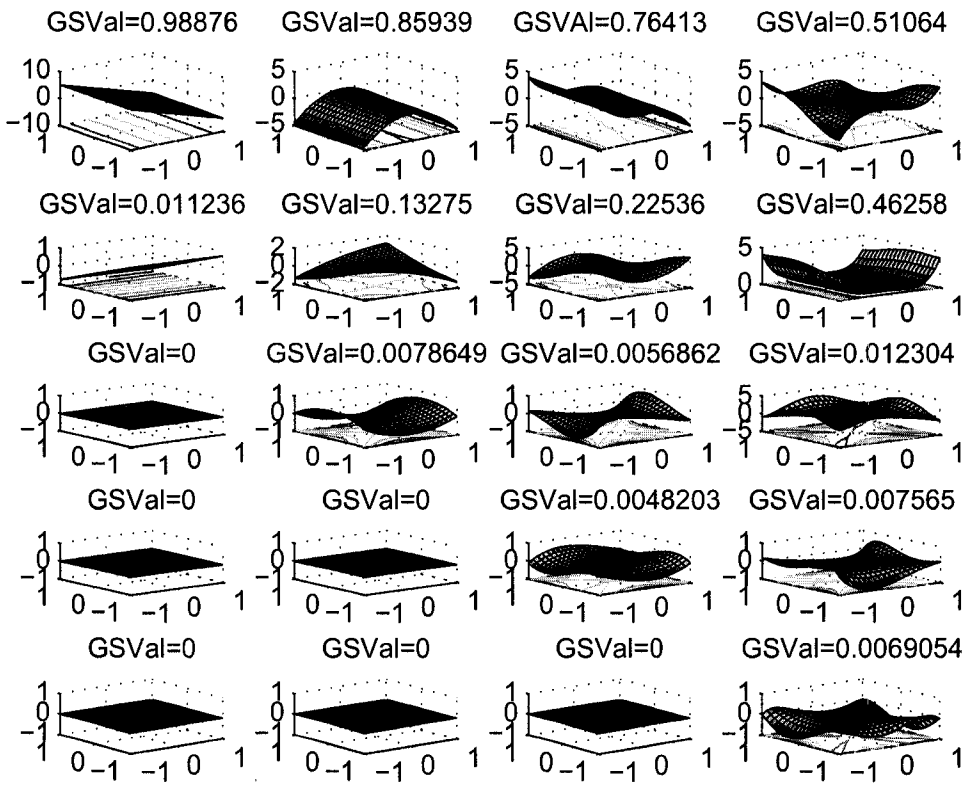


Figure 5.9: Toy example for performing KSFA in 3-D. From left to right the degree of the polynomial kernel increases from one to four. From top to bottom the contour lines of constant SFA components associated to decreasing order of generalized eigenvalues and the hyper-plane that captures the structure of the data are shown.

Figures 5.8 and 5.9, show the results for performing KPCA and KSFA respectively on the same toy example that we used previously. In this section for each graph we have the contour lines as well as the hyper-planes; the surface of the projections of test points in feature space onto the eigenvectors and generalized eigenvectors for KPCA and KSFA cases respectively that capture the structure of the data.

From the results in Figures 5.8 and 5.9, we observe the same feature (feature for generated data which is a quadratic feature) is captured in KSFA case with polynomial degree of two (see the first graph in the second column in Figure 5.9).

The hyper-planes capture the structure of the data in both Figures 5.8 and 5.9 but it appears that via KSFA these hyper-planes find the structure of the data in less steps (note that the structure of the data was quadratic and the first figure in the second column from left in Figure 5.9 shows it all, however it is not that clear in KPCA case) when they also capture more signals than noise at the same time, and this is not the case in KPCA. Here again we see that the last SFA hyper-planes have almost the same shape for different polynomial degrees, even their shapes are similar to the last hyper-planes in PCA case.

Even these simplified examples are non-trivial to interpret. In Chapter 7 this comparison is facilitated by the existence of a problem dependent objective function. The next toy example helps us visualize the differences between KPCA and KFSA.

5.4.2 Sinusoidal Toy Example

As described above, it is nontrivial to make an objective interpretation of what is happening when one kernelizes SFA and PCA. In part this is due to the fact that the data is never actually computed in the feature space, and we are left to infer what is happening without actually seeing it. However, for low-dimensional examples, (here we select the domain to have dimension 2 and the degree of the polynomial to be one to four) we can actually compute explicitly what is happening in the range, of feature space.

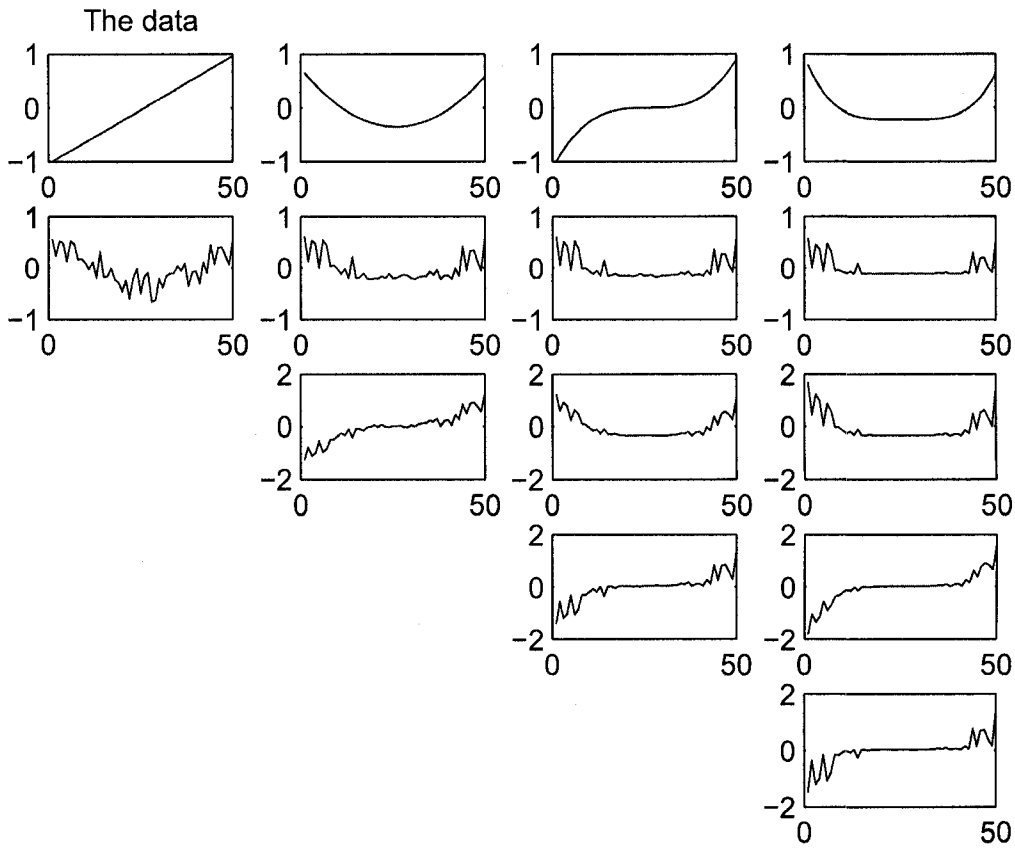


Figure 5.10: The data $(x, \sin \pi x)$ mapped using a Veronese mapping of degree the same as the column number.

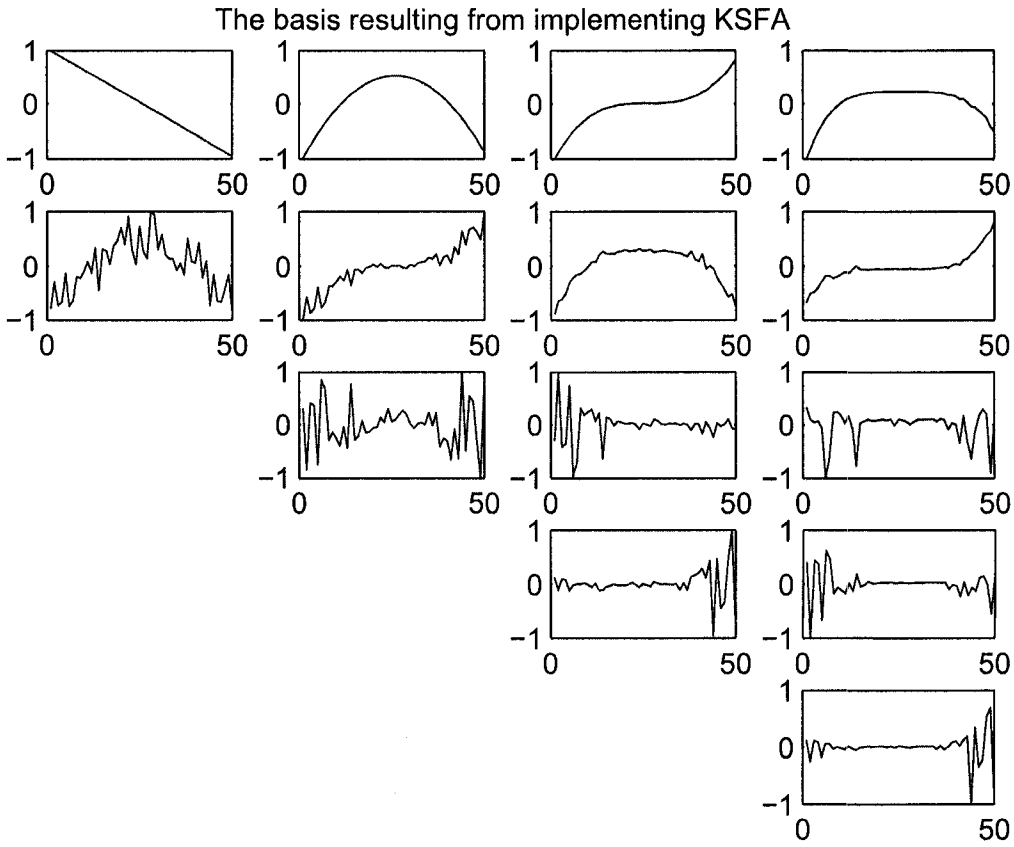


Figure 5.11: The basis resulting from implementing KSFA. The basis vectors correspond to the data in Figure 5.10.

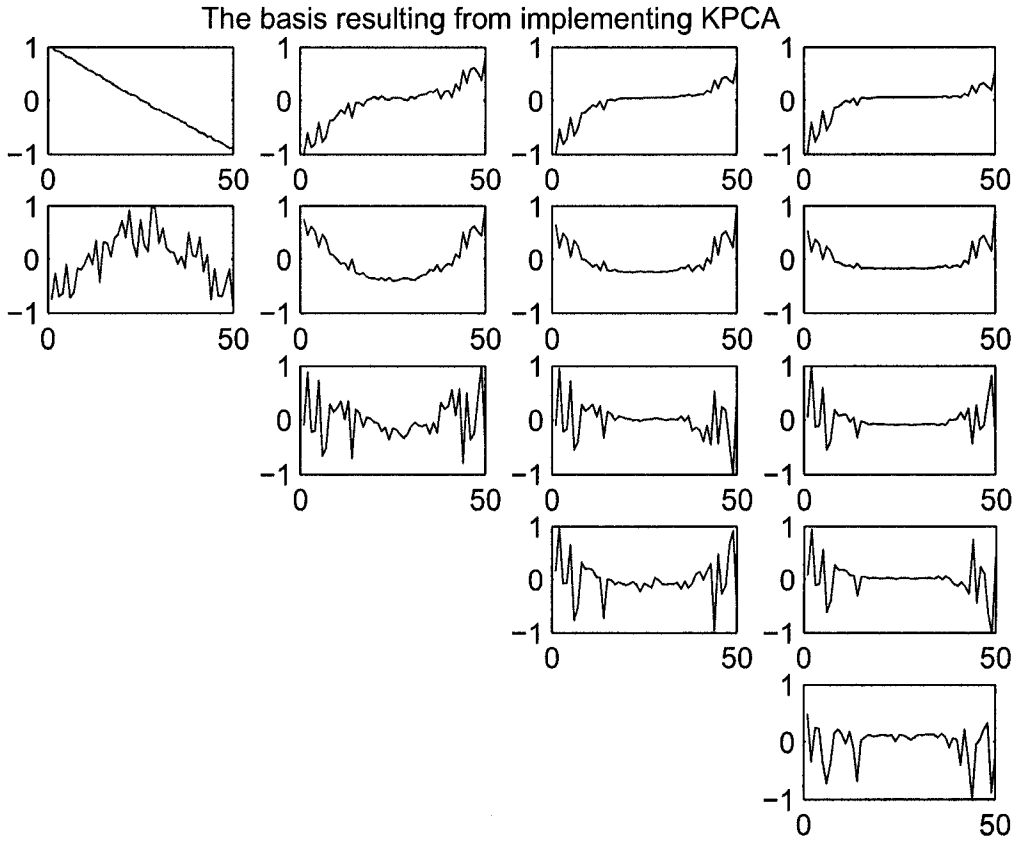


Figure 5.12: The basis resulting from implementing KPCA. The basis vectors correspond to the data in Figure 5.10.

Now we use the data set generate from 50 points sampled evenly in x between (-1,1) as

$$y = \sin(\pi x) + N$$

Our first data matrix x_1 consists of 50 rows of these values of (x, y). For a Veronese mapping of degree two each sample is now $(x^2, y^2, \sqrt{2}xy)$, hence x_2 is a 50×3 matrix. For the Veronese mapping of degree 3 we have each point mapped to $(x^3, y^3, \sqrt{3}x^2y, \sqrt{3}xy^2)$ and for degree 4 $(x^4, y^4, \sqrt{6}x^2y^2, 2x^3y, 2xy^3)$. The resulting data is shown in Figure 5.10. The associated KSFA and KPCA bases are shown in Figures 5.11 and 5.12, respectively. In these figures it is striking how the KSFA mode with the highest signal to noise ratio picks up the clean data x raised to the appropriate power. This example illustrates clear differences between the behavior of KSFA and KPCA.

5.5 Summary of Contributions

We have proposed a nonlinear extension to signal fraction analysis based on a kernel method, i.e., kernel signal fraction analysis. In addition, we have proposed a second algorithm that uses noise adjustment in the data domain before kernelization. We present a detailed derivation of the methodology using kernel principal component analysis as a prototype. We compare these methods using two toy examples and illustrate the benefits of kernel signal fraction analysis.

For example, in [34] Gordon has used an extension on MNF via applying nonlinear form of it to time-dependent airborne electromagnetic (AEM) data. He generated new bands by raising the original bands to the powers of $q = 1, \ldots, 6$, effectively converting the linear MNF transformation to a polynomial filter. In his work, he has used the inverse matrix for the signal covariance and applied the eigenvalue decomposition to find the linear transform coefficients. He showed that this generalization leads to improved performance over the basic linear MNF; however, in general this method can be characterized as a narrow extension of MNF while the work proposed here contains this approach as a special case.

Chapter 6

APPLICATION IN SIGNAL SEPARATION AND COMMUNICATION

In this chapter we investigate some applications of SFA in adaptive beam forming for wireless communications. In a wireless communication system the base station communicates with a number of mobiles simultaneously. Hence, in the course of communication with a particular mobile the base station not only receives signals from that mobile but also from other mobiles. This enhances multi-user-interference (MUI) and degrades the performance of wireless systems. Multiple access (MA) schemes have been developed to reduce MUI effects by generating a form of orthogonality between signals transmitted to (or received from) mobiles. Space division multiple access (SDMA) and code division multiple access (CDMA) are two popular multiple access schemes [63].

SDMA is achievable via adaptive smart antennas. It reduces interference effects and enhance wireless network capacity by directional-beam antennas [63]. The ultimate goal of adaptive antennas is to create an antenna pattern with its main lobe directed toward the desired user and its nulls directed toward the interfering users. This is possible via beam forming techniques. In a time varying environment, this process is accomplished using adaptive beam formers.

Adaptive beam formers are essentially time varying optimal filters that make the optimum estimate of the transmitted signal via minimizing interference effects. Adaptive beam formers observe the received signal (which is a merger of the desired signal, interfering signals and noise) and generate the finest estimate of the desired signal by applying

weight matrix to the observed data. This weight matrix is computed via optimizing a cost function, e.g., signal to noise ratio (in SFA).

In CDMA schemes the orthogonality is generated by assigning orthogonal codes (in time or frequency domain) to the users [63]. If the codes are applied in the frequency domain, it is called multi-carrier CDMA (MC-CDMA). In our work, we merge multi-carrier code division multiple access (MC-CDMA) systems and adaptive antenna arrays and we apply SFA technique as a beam forming technique to minimize the interference and noise effects. Via the merger of SFA and MC-CDMA systems we achieve: 1) A very high probability-of-error performance via reducing the interference effects using both beam forming scheme and multiple access technique, reducing fading effects using frequency diversity inherent in MC-CDMA, and reducing noise effects with SFA technique, and 2) A very high capacity via directionality created by antenna arrays and MC-CDMA.

Traditionally, the key to meeting network capacity demands in a wireless environment has been spatial reuse: Split the wireless communication coverage area (e.g., a city) into smaller areas called "cells" [78, 84, 85, 88]. The base station (BS) located at the center of cell transmits the signals to all users in its cell and receives the signals from all these users. Neighboring cells avoid interference with one another via either frequency reuse [77, 15, 16, 111] or code reuse [57, 72] scheme (or possibly a combination of both) (see figure 6.3). This cellular concept increases power efficiency by reducing the amount of power that must be transmitted from BS to users and vise versa. This reduces the cost of transmitters (power amplifiers) at both the BS and mobile user.

Adaptive antenna array forms main lobes in the direction of the intended users and simultaneously produces nulls at the positions of other users. In this way, users' signals are effectively separated from one another based on each users' geographic location, and this allows for significant gains in network capacity. As evidenced from this discussion, adaptive antenna arrays are a technological innovation with a very promising future in the world of wireless telecommunications. Adding to the value of the antenna array is

the fact that it not only provides significant increases in network capacity (measured by numbers of users per cell), it can alternatively improve signal quality (measured by probability-of-bit-error) via various transmit diversity techniques [4, 56, 76, 38, 108]. The main goal of adaptive antennas, i.e., steering the antenna main lobe toward the desired user and its nulls toward the interfering users is attainable by employing adaptive BF techniques. Adaptive beam formers are essentially time varying optimal filters make the best estimate of the transmitted signal by minimizing noise and interference effects in the observed signal. In general, the observed signal can be considered as the observed data, and adaptive beam formers can be viewed as data analysis techniques. In a wireless communication system, adaptive beam formers observe the received signal which is a merger of the desired signal, interfering signals and noise and generate the best estimate of the desired signal by applying a weight matrix to the observed data. This weight matrix is computed by optimizing a cost function.

In this dissertation, we have conducted a study to investigate the capabilities of SFA merger with MC-CDMA [multi carrier code division multiple access (MA) scheme] [3, 37]. MC-CDMA is emerging as a powerful multiple access protocol [87, 47]. In MC-CDMA, each user's information symbol is transmitted over N carriers (frequencies) simultaneously. To ensure orthogonality of all users' data streams at the receiver, each user assigns a unique spreading sequence to the N carriers.

In order to apply SFA beam forming technique to detect the signals transmitted from the desired user, base station (BS) incorporates an M-element antenna array whose elements are connected to a bank of MC-CDMA receivers, specifically designed for that user (user j). The SFA beam former is applied to the output of receivers (here, MC-CDMA receivers). The output of SFA beam former would be an estimation of the transmitted signal by the desired user (see Figure 6.1. In Figure 6.2 the structure of the SFA beam former is shown.

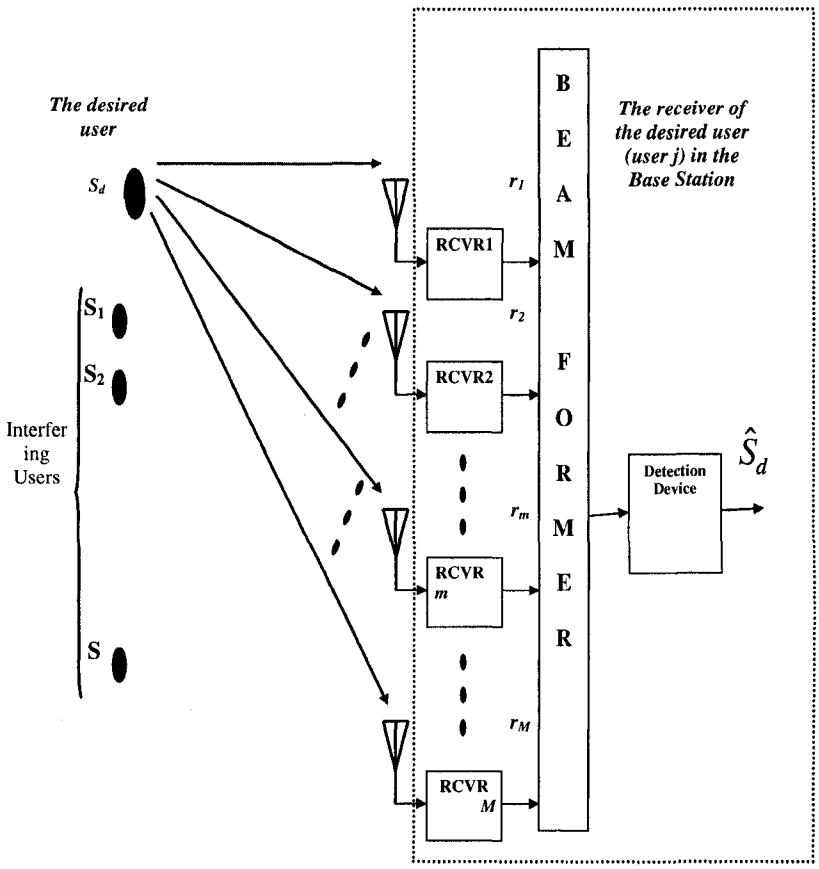


Figure 6.1: System structure.

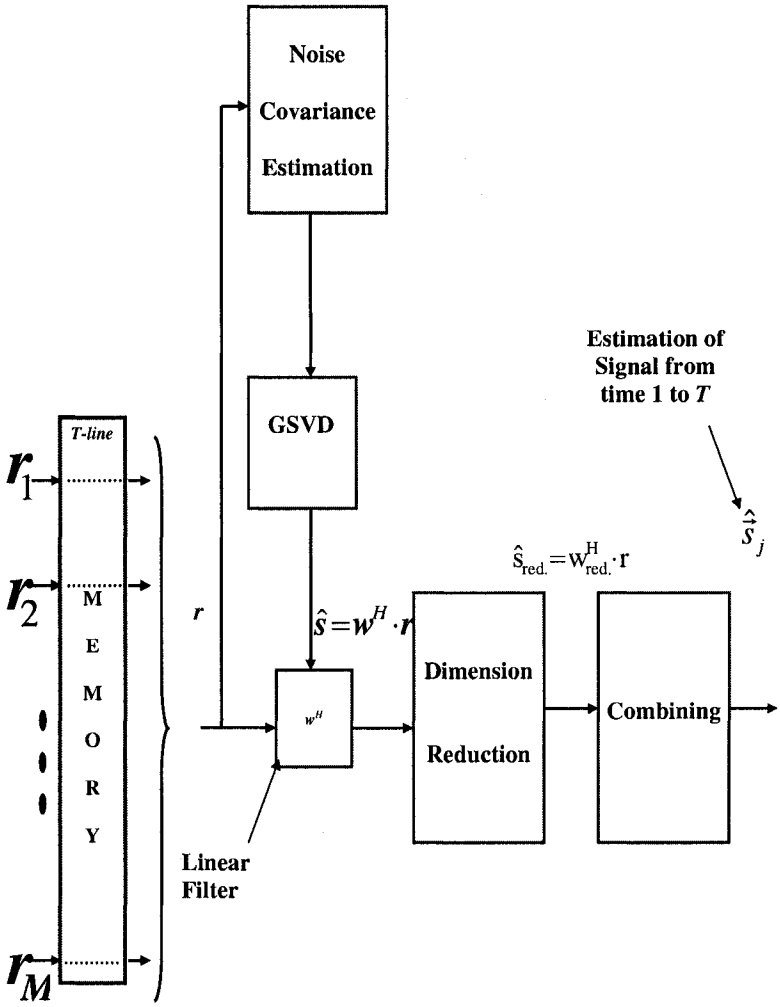


Figure 6.2: The structure of SFA beam former.

6.1 SFA and MC-CDMA merger

In this section, we represent the performance of SFA via a merger with MC-WCDMA systems. Here, we consider an uplink scenario shown in Figure 6.3. In this figure, the BS receiver structure has shown for a specific user that is called the desired user (here, user j). Hence, the receiver of this user considers the signals from other users as interfering signals. The signals transmitted from users are assumed independent. In order to apply SFA BF technique to detect the signals transmitted from the desired user (user j), BS incorporates an M-element antenna array whose elements are connected to a bank of MC-CDMA receivers, specifically designed for user j (see Figure 6.1). The SFA BF is applied to the output of receivers. The output of SFA BF is an estimation of the transmitted signal by the desired user.

In MC-CDMA, each user's bit is transmitted simultaneously over N narrow band sub-carriers (frequencies) [102, 3, 37]. Sub-carriers are equally spaced in frequency by Δf. To ensure that the users are separated at the receiver side, each user applies a unique spreading code to the carriers. Assuming BPSK modulation and a multi-path channel, the received signal vector $\vec{r}(t)$ at an M - element array corresponds to:

$$\vec{r}(t) = \sum_{k=1}^{K} \sum_{v_k=1}^{v_k} \sum_{n=0}^{N-1} \alpha_k^{n,v_k} \beta_k^n b_k[i] \vec{\Phi}(\phi_k^{v_k}) \cos(2\pi(f_0 + n\Delta f)t + \phi_k^n) + \vec{n}(t) \qquad (6.1)$$

Where $b_k[i] \in -1, 1$ is the transmitted bit, f_0 is the carrier frequency, $\Delta f = 1/T_s$ (T_s: Symbol time duration) is employed to maintain carrier orthogonality and $\beta_k^n \in -1, 1$ refers to the nth element of the spreading sequence of user k, $k \in [1, K]$, where $K = 1+K_I$ is the number of users and K_I refers to the number of interfering users (see Figure 6.1) [29]. is the fade amplitude which is a random variable considered equivalent over all antenna elements (assuming the elements are located close enough, e.g., the distance between these elements is $d = \lambda/2$ ($\lambda = C/f_0$ is the wave length and C is the speed of

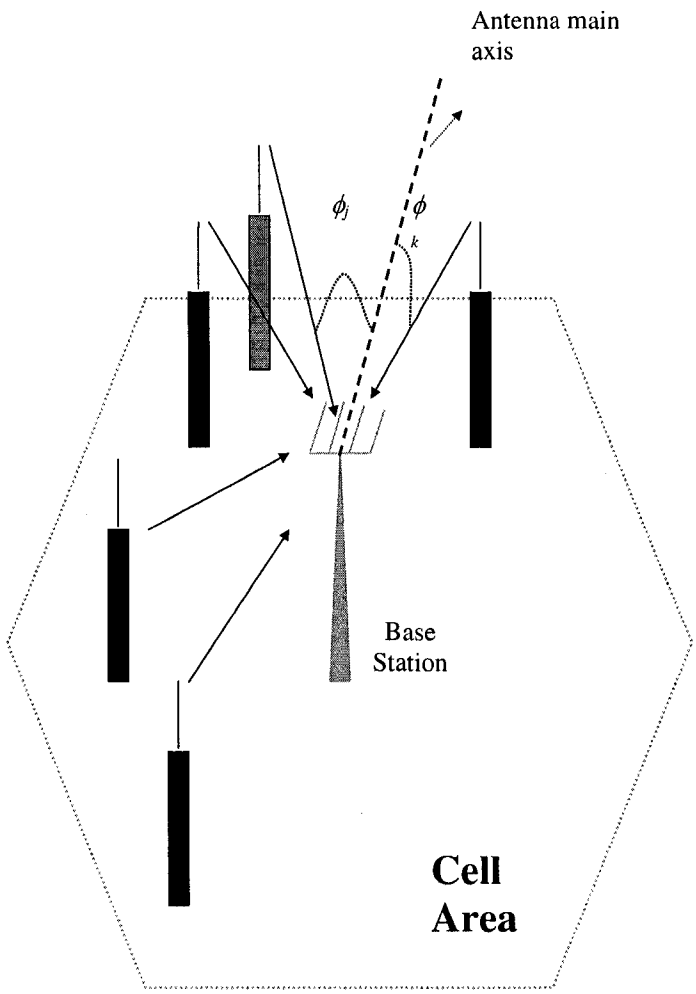


Figure 6.3: Uplink condition. Users transmit to the base station.

light), independent for different users, for each symbol durations T_s and each path v_k , and correlated over frequency components. The correlation coefficient between the pth subcarrier fade and the qth subcarrier fade is characterized by [46]:

$$\rho_{p,q} = \frac{1}{1 + ((p-q) \cdot (\Delta f/(\Delta f)_c)))^2} \tag{6.2}$$

In Equation (6.2) $(\Delta f)_c$ is the coherence bandwidth of the channel.

We assume ϕ_k^n is the fade phase of the frequency component n of user k received from direction v_k, a uniform i.i.d. random variable between 0 and 2π. Here after, to verify the capabilities of SFA, we assume $\phi_k^{v_k}$, i.e., the fade phase of the desired user, is tracked and removed. The vector $\vec{\Phi}(\phi_k^{v_k}) \in \mathbf{R}^M$ is the array response vector of the kth user's v_k th direction (path) corresponds to:

$$\vec{\Phi}(\phi_k^{v_k}) = [1, e^{j\frac{2\pi l}{\lambda}\cos(\phi_k^{v_k})}, \ldots, e^{j(M-1)\frac{2\pi l}{\lambda}\cos(\phi_k^{v_k})}]^T \tag{6.3}$$

Here, $\phi_k^{v_k}$ is the angle of arrival (AOA) which is the angle between line of arriving signal from path v_k of user k and the main axis of the antenna array and assumed to be estimated, and is the Gaussian noise vector. At the MC-CDMA receiver of the jth user, a bank of optimum filters permits the signal corresponding to one carrier, n orthogonal codes β_j^n of the jth user to reduce inter-user interference. Hence the signals at each carrier frequency n corresponds to

$$\vec{r}^n[i] = \sum_{v_j=1}^{V_j} \alpha_j^{n,v_j}[i] b_j[i] \vec{\Phi}(\phi_j^{v_j}) + \sum_{k=1,k\neq j}^{K} \sum_{v_k=1}^{V_k} (\alpha_k^{n,v_k} \vec{\Phi}(\phi_k^{v_k}) e^{j(\phi_{k,v_k}^n - \phi_{j,v_j}^n)} \beta_k^n \beta_j^n) b_k[i] + \vec{n}^n \tag{6.4}$$

Signals are combined over frequency components to enhance the performance of the carrier. Considering EGC over the frequency components, the received signal array corresponds to

$$\vec{r}[i] = \sum_{n=1}^{N} \vec{r}^n[i] \tag{6.5}$$

which is,

$$\vec{r}[i] = \sum_{n=1}^{N} \sum_{v_j=1}^{V_j} \alpha_j^{n,v_j}[i] b_j[i] \vec{\Phi}(\phi_j^{v_j}) + \sum_{n=1}^{N} \sum_{k=1,k\neq j}^{K} \sum_{v_k=1}^{V_k} (\alpha_k^{n,v_k} \vec{\Phi}(\phi_k^{v_k}) e^{j(\phi_{k,v_k}^n - \phi_{j,v_j}^n)} \beta_k^n \beta_j^n) b_k[i] + \vec{n} \tag{6.6}$$

The first term in the right hand side of Equation (6.6) is the desired signal, the second term is the interfering signal, and the last term is the noise. In general, the signal at the input of SFA beam former is characterized by

$$\vec{r}[i] = \vec{s}[i] + \vec{\eta}[i] \tag{6.7}$$

where

$$\vec{s}[i] = \sum_{n=1}^{N} \sum_{v_j=1}^{V_j} \alpha_j^{n,v_j} \cdot \vec{\Phi}(\phi_j^{v_j}) \cdot b_j[i] \tag{6.8}$$

is the desired signal and

$$\vec{\eta}[i] = \sum_{n=1}^{N} \sum_{k=1,k\neq j}^{K} \sum_{v_k=1}^{V_k} (\alpha_k^{n,v_k} \vec{\Phi}(\phi_k^{v_k}) e^{j(\phi_{k,v_k}^n - \phi_{j,v_j}^n)} \beta_k^n \beta_j^n) b_k[i] + \vec{n} \tag{6.9}$$

represents the total interference and noise.

6.2 Structure of non-overlapping window adaptive algorithm

Linear SFA beam former is a linear filter that generates an estimate of the transmitted desired signal. The linear filter (i.e., matrix w in Figure 6.2) is generated by solving an optimization problem. Here, we consider a non-overlapping window (NLW) adaptive realization of SFA (NLW/SFA). NLW/SFA BF incorporates a memory to save a block of data in the sliding non-overlapping windows of length T (see Figure 6.2). The block of observed signals from an M-element antenna array ($\vec{r}[i] = [r_1[i], r_2[i], \ldots, r_M[i]]^H$, $i \in 1, 2, \ldots, T$) within the time duration T is used to generate and update the linear filter components (i.e., matrix w components), and estimate the transmitted signals within the same portion of time.

As described earlier, the linear SFA optimization problem maximizes the total signal-to-signal ratio, and leads to a GSVD problem [31, 91, 80, 64, 26, 62]. The GSVD problem computes generalized eigenvectors (EVECs) (w_m, $m \in 1, 2, \ldots, M$) and their corresponding generalized eigenvalues (EVALs). Each EVAL is proportional to a signal-to-noise ratio (SNR). Hence, the computation complexity can be reduced by removing small EVALs (related to small SNR) and their corresponding eigenvectors ("Dimension Reduction" box in Figure 6.2). Finally, the estimated desired signals corresponding to each eigenvector is combined to generate the final estimation of the desired signal.

As a result, at the output of the memory of Figure 6.2 we have a matrix of data at times 1 to T, and over all antenna elements $m \in 1, 2, \ldots, M$ that can be represented by:

$$r = s + \eta \tag{6.10}$$

where $r \in R^{M \times T}$, and $T \gg M$

$$r = [\vec{r}[1], \vec{r}[2], \ldots, \vec{r}[T]] \tag{6.11}$$

Here $\vec{r}[i]$ represents the received signal from elements 1 to M at time i,$i \in 1, 2, \ldots, T$. The SFA transformation is a linear transformation of the received signal which estimates the desired signal via maximizing the SNR. The estimation of the desired signal is defined by

$$\hat{s} = w^T \cdot r \tag{6.12}$$

In Equation (6.12) $w^T = [\vec{w}_1, \vec{w} - 2, \ldots, \vec{w}_M]^T \in R^{M \times M}$ where $\vec{w}_m \in R^M$,$m \in 1, 2, \ldots, M$ are the weight vectors. These weight vectors are computed by maximizing the signal-to-noise ratio. SNR is defined as a ratio of SFA beam former output signal power (variance) and SFA beam former output noise power (variance). Assuming $\Gamma_{obs} = r \cdot r^T$ and $\Gamma_n = \eta \cdot \eta^T$ the GSVAL problem for all values of m corresponds to

$$\Gamma_{obs} \vec{w}_m = \lambda_m \Gamma_n w_m \tag{6.13}$$

In Equation (6.13) λ_m is the generalized singular value (GSVAL) and w_m is the corresponding generalized singular vector (GSVEC). In addition, $(1/T)\Gamma_o bs \in \mathbf{R}^{M\times M}$ is the observed signal sample covariance matrix over antenna elements, and it is known. $(1/T)\Gamma_n \in \mathbf{R}^{M\times M}$ is the noise sample covariance matrix, and it is unknown and should be estimated. Equation (6.13) is solved for all (λ_m, w_m) , $m \in 1,\ldots,M$. Hence, the GSVAL problem for all values of m corresponds to

$$\Gamma_{obs} w = \vec{\lambda}\Gamma_n w \tag{6.14}$$

where w is introduced in Equation (6.12) and $\vec{\lambda} = [\lambda - 1,\ldots,\lambda_M]$. Organizing the GSVALs in a decreasing order, the GSVALs and their associated GSVECs are $\lambda_1 \geq \lambda_2,\ldots,\geq \lambda_M$ and $w = [\vec{w}_1,\ldots,\vec{w}_M]$, respectively. Hence, as shown in Figure 6.2, to generate the SFA weight vectors, we should estimate the noise covariance matrix and solve the GSVD problem in Equation (6.14). Now, considering $\lambda_m = SNR_m + 1$ generalized vectors associated to the small generalized singular values correspond to small SNRs. Hence, in order to reduce the computational complexity, we can reduce the dimension of w and change it to $w_red = [\vec{w}_1,\ldots,\vec{w}_Q]$, $Q < M$. Here, for example, we disregard singular vectors corresponding to singular values that are much smaller than the largest singular value, i.e., λ_1 . This process is called the dimension reduction (Figure 6.2). Now, we have Q estimations of the desired signal from time 1 to time T corresponds to

$$\hat{s}_{red} = [\hat{\vec{s}}[1],\ldots,\hat{\vec{s}}[T]]_{Q\times T} \tag{6.15}$$

where $\hat{\vec{s}}[i] = [\hat{s_1}[i],\ldots,\hat{s_Q}[i]]^T$. These replicas of the desired signal estimations are associated with high signal to noise ratios and combined to better estimate the desired signal. The combiner output corresponds to

$$\hat{\vec{s_d}} = [\hat{s_d}[1],\ldots,\hat{s_d}[T]] \tag{6.16}$$

where $\hat{s_d}[i] = \sum_{q=1}^{Q} z_q s - q[i]$ when $i = 1,\ldots,T$ and $\vec{z} = [z_1,\ldots,z_Q]$ is the weight vector corresponding to the combiner.

6.3 Assumptions in the Simulation

To investigate potentials of SFA beam former we assumed:

1. 7-element antenna array (M = 7)
2. 32-Carrier MC-CDMA ($N = 32$)
3. The desired user, and six interfering users are at 0, +3, -4, +5, +50, -40, and +80 degrees, respectively;
4. Perfect power control for all users and directions, this leads to the same average power for the signals received from different directions and different users;
5. T =100-line memory, each sample corresponds to one transmitted bit (see Figure 4.5);
6. Ideal estimation of noise covariance matrix is assumed;
7. Equal gain combining technique in SFA Beam forming is used;
8. Dispersion angles considered Gaussian with the means in assumption 3, and standard deviations of 0, 2, 10 and 20 degrees. Hence, with the angular positions represented in assumption 3, angular standard deviations of 0 and 2 leads to a low dispersive channel while angular standard deviations of 10 and 20 leads to a high dispersive channel.

6.4 Simulation results

We apply SFA to adaptive smart antennas in wireless communication and beam forming to extract the desired signal from the received signal in different environments. We have considered two environments: a) low dispersive environment (see Figure 6.4), where the signals sent by users and scattered by the objects in the environment are not

overlapping and b) high dispersive environment (see Figure 6.5) where the signals from different users are overlapping. We have generated the Probability-of-error performance curves in both low and high dispersive environments and the effect of length of data and perturbation in noise covariance estimation on the performance of wireless system is studied in this section.

6.5 Study of SFA on High and Low Dispersive Environment

Assuming no interfering user the simulation results are shown for an additive white Gaussian noise (AWGN) channel (see the curves in the bottom in both Figures 6.7, and 6.6 as well as a fading channel (the curve at the top in Figure 6.7, also see Figure 6.6). Since there is no interfering user in these two cases, we call them ideal AWGN and ideal fading channels, respectively. In our work, the SFA beam former performance curves are compared with these two ideal performance curves.

The other two curves in Figure 6.7 represent SFA simulation results for a low dispersive environment. It is seen that applying SFA makes the probability-of-error performance even slightly better than the performance in an interference free environment (the ideal performance curves).

This outcome can be explained as follows. The ideal fading channel performance curve shows the results for an interference free fading environment, while the noise effect still exists. However, when we apply SFA beam former for a low dispersive channel, it highly reduces both the interference and noise. These simulations represent the capability of linear SFA beam former to separate the desired user from the interfering users and extract its signals perfectly in low dispersive environments.

Figure 6.6 represents the degradation in performance in high dispersive environments. Specifically, we see a considerable reduction in performance as the dispersion angle increases. This shows the linear SFA is not capable of separating the desired user signal from the interfering users' signals in these environments.

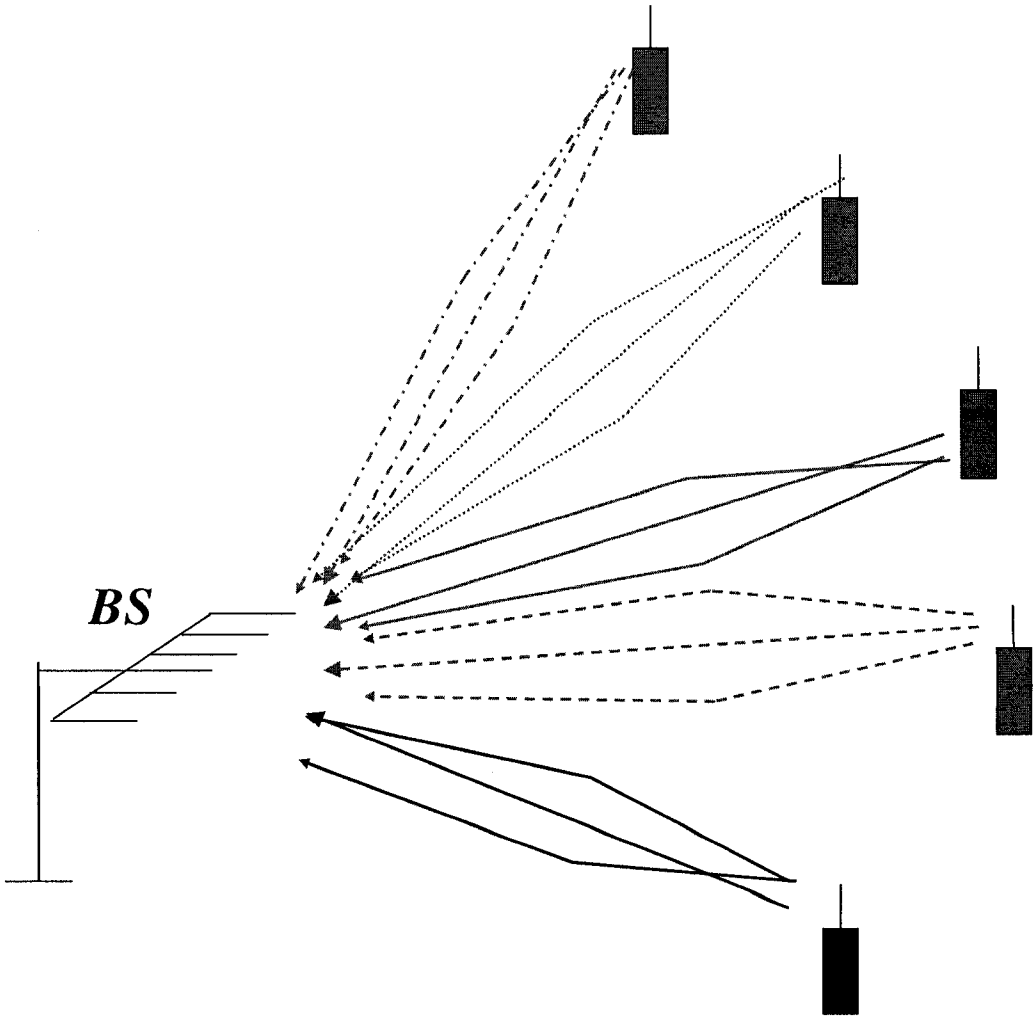


Figure 6.4: Low dispersive environment.

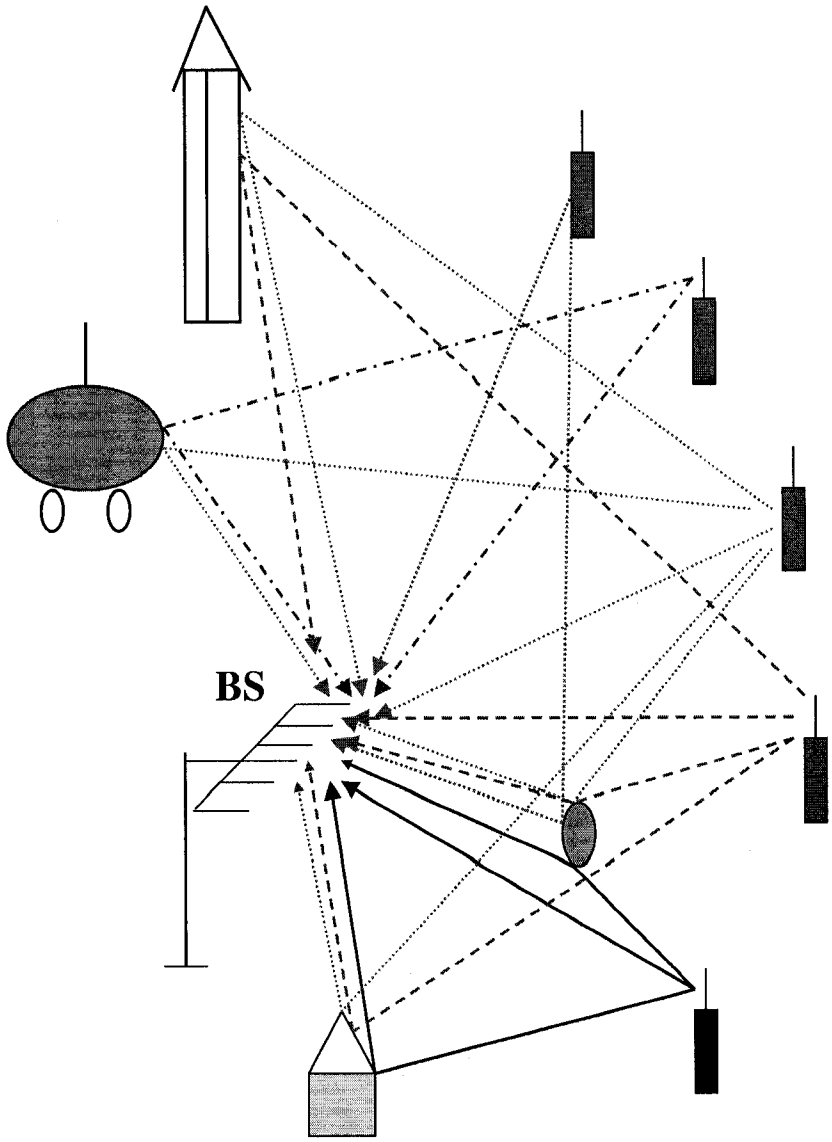


Figure 6.5: High dispersive environment.

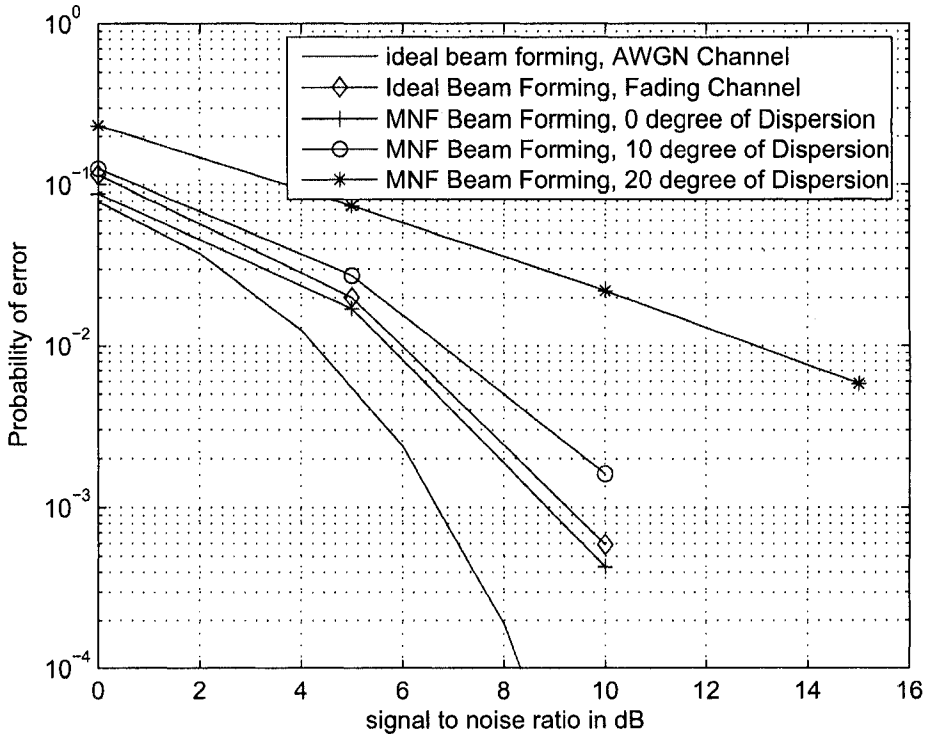


Figure 6.6: probability-of-error performance simulation results.

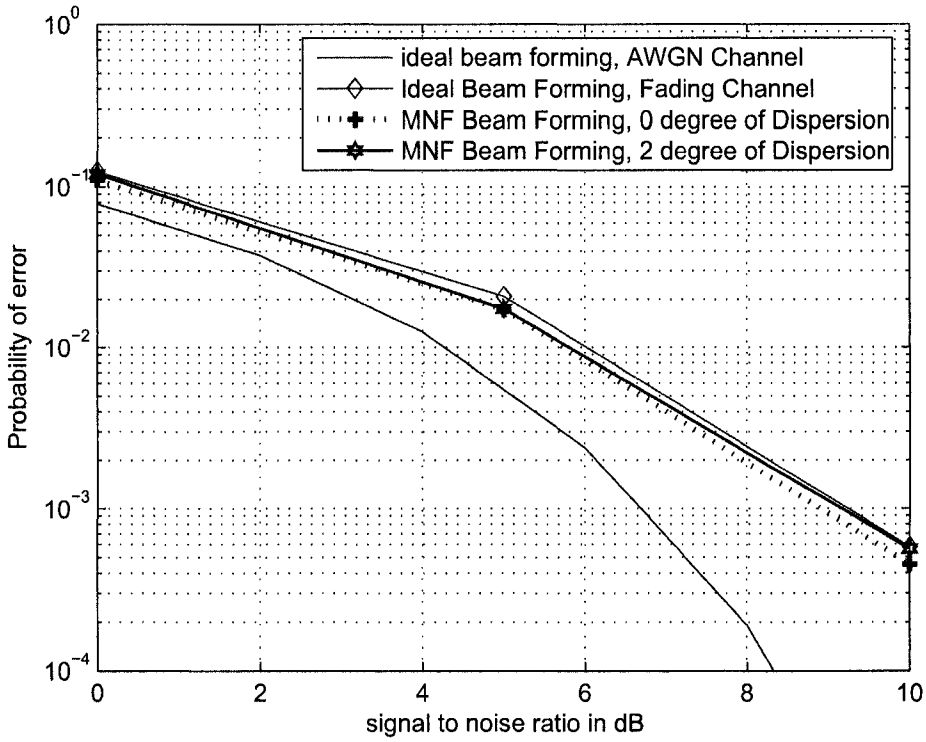


Figure 6.7: probability-of-error performance simulation results.

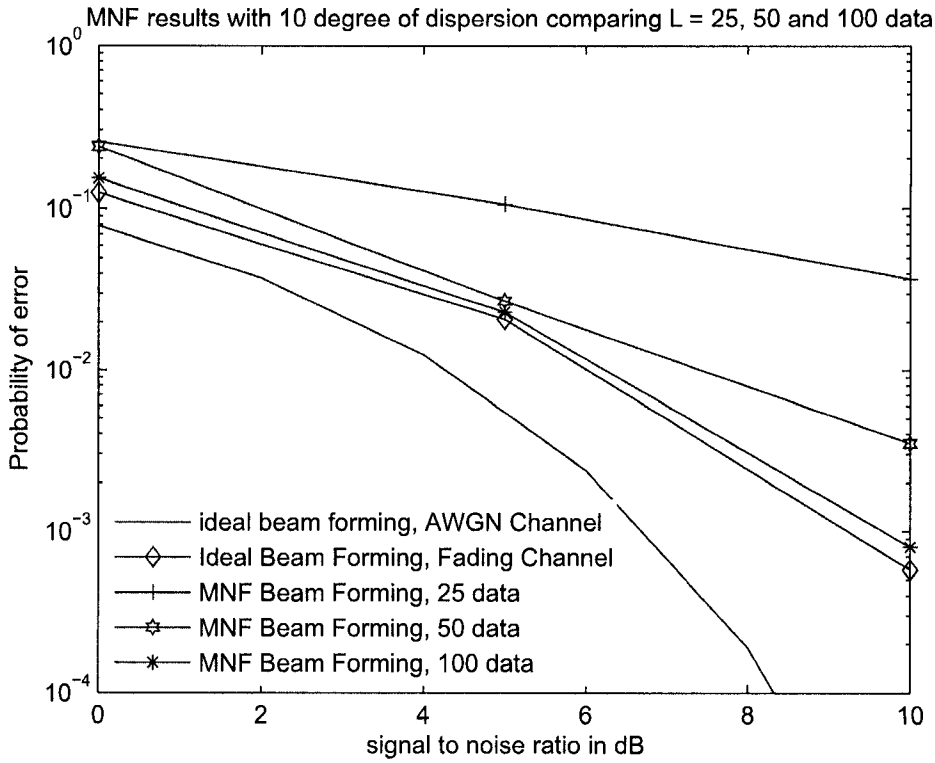


Figure 6.8: Comparisons between different lengths of data.

6.6 Study of the Data Length

We have also studied the effect of the selection of the number of time components on the performance of the receiver. In Figure 6.8 we have generated the performance simulation results allowing different widths (L) for the blocks of data. As we mentioned previously to convert the statistical means in Equation (4.17) to sample means in Equation (4.18), a large number of samples should be available. The simulations results in Figure 6.8 represent a dramatic improvement in the performance as the number of time samples increases from 25 to 100. Increasing the number of samples beyond 100 does not change the performance considerably. We conclude that the length of 100 is a good selection.

6.7 Study of the Perturbation Effects

We have also studied the perturbation effects in the noise and the observed covariance matrices of an SFA beam former merged with an MC-CDMA system in low and high dispersive environments.

To study perturbation effects, we define the percentage of perturbation as the ratio of the difference between the standard deviation of the true and the perturbed estimation of the elements of (noise or observed) covariance matrix, and the standard deviation of the true estimation of the elements of noise covariance matrix. Simulations show that a perturbation in the order of 10% degrades the performance of SFA/MC-CDMA, while the system tolerates minor perturbations in the order of 3%. This characterizes SFA beam forming a promising technique implemented in adaptive smart antenna for future generations of wireless communications.

We observe that this merger leads to a high probability-of-error performance compared to an ideal beam former in low dispersive environments, since SFA beam former reduces noise and interference effects at the same time while ideal beam former reduces just the interference effect. However, in high dispersive environments a lower probability-of-error performance is achievable.

SFA solves a GSVD problem, includes the noise and the observed covariance matrices; hence, perturbations in these matrices reduces the performance of this beam former. Large percentage of perturbations, e.g., in the order of 10 percent, highly degrades the performance of SFA systems.

We simulated the probability-of-error performance curves assuming 0% (ideal estimation of the noise or observed covariance matrix), 3% and 10% of perturbation generated via Gaussian noise with variances of 0.001 and 0.01 respectively.

Figures 6.9 and 6.10 also show 3% of perturbation lowers the performance at the probability-of-error of 10^{-3} for 0 and 15 degrees of dispersion, respectively, which is

relatively small. However, increasing perturbation to 10% and or considering a hybrid perturbation, the performance degrades by a large factor. Moreover, Figure 6.9 and 6.10 represents the simulation results considering a hybrid perturbation of 3% and 10% in the observed and the noise signal covariance matrices, respectively. These results show SFA performance is highly sensitive to the accuracy of the noise covariance matrix estimation.

When we apply SFA beam former, it highly reduces both the interference and the noise effects. Hence, specifically, at low dispersion areas with an ideal estimation of the covariance matrices, the performance of SFA beam former is even better than ideal beam forming (See Figure 6.9).

This result represents the capability of linear SFA beam former to separate the desired user from the interfering users and extracts its signals in relatively low dispersive environments when the noise covariance matrix is ideally estimated and the received signal covariance matrix is computed with zero perturbation.

6.8 Beam Pattern for Some Degrees of Dispersion

The probability-of-error performance simulation results generated for 0, 2 and 10 degrees of dispersion are shown in Figures 6.11, 6.13, and 6.15, respectively. Assuming interfering users are ideally nulled, i.e., we completely remove their interfering effects, the simulation results are generated for an additive white Gaussian noise (AWGN) channel (see the curves in the bottom of figures 6.11, 6.13, and 6.15) as well as a fading channel (the next curve at the top the AWGN channel curve). Since there is no interfering user in these two cases, we call them ideal AWGN and ideal fading channels, respectively.

The other three curves from top to the bottom in Figures 6.11, 6.13, and 6.15 represent BER simulation results for 10%, 3% and 0% of error in the noise covariance matrix estimation, respectively. Their associated beam pattern averaged over 25 samples is sketched in Figures 6.12, 6.14, and 6.16. The beam pattern nulls of Figures 6.12, 6.14,

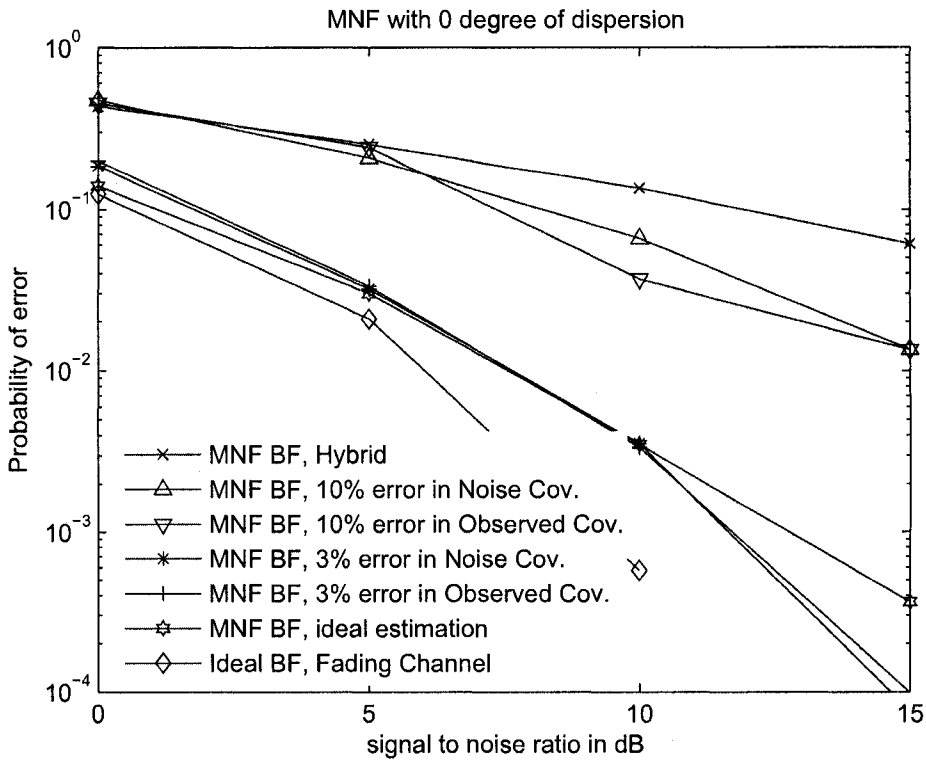


Figure 6.9: Performance of SFA with perturbations in noise and observed covariance matrices for channels with 0 degree of dispersion.

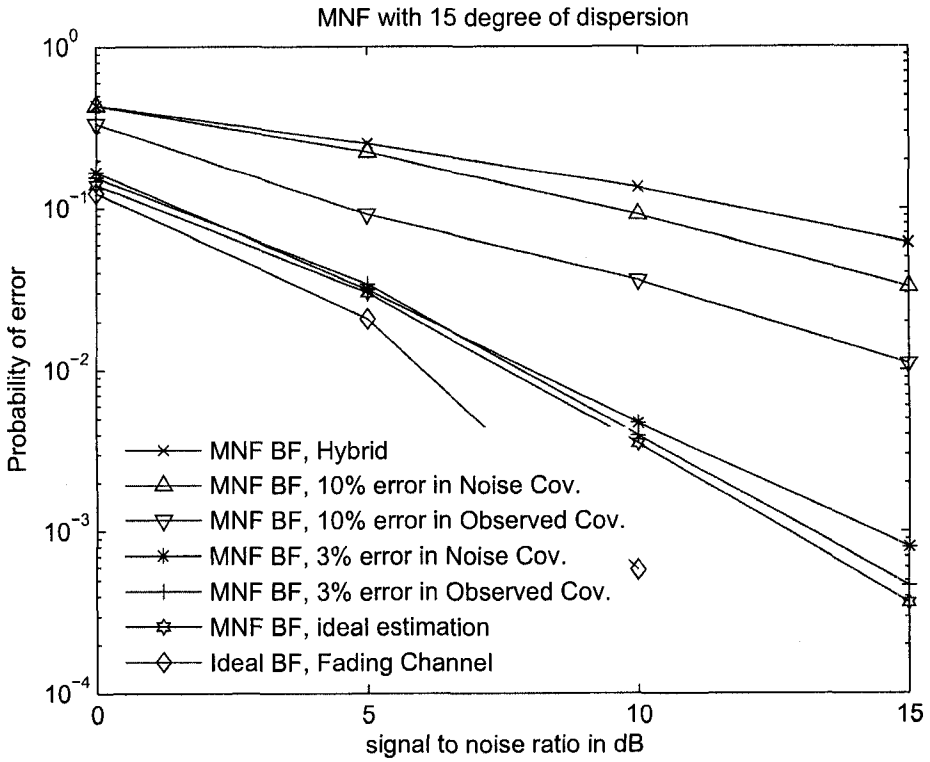


Figure 6.10: Performance of SFA with perturbations in noise and observed covariance matrices for channels with 15 degres of perturbation.

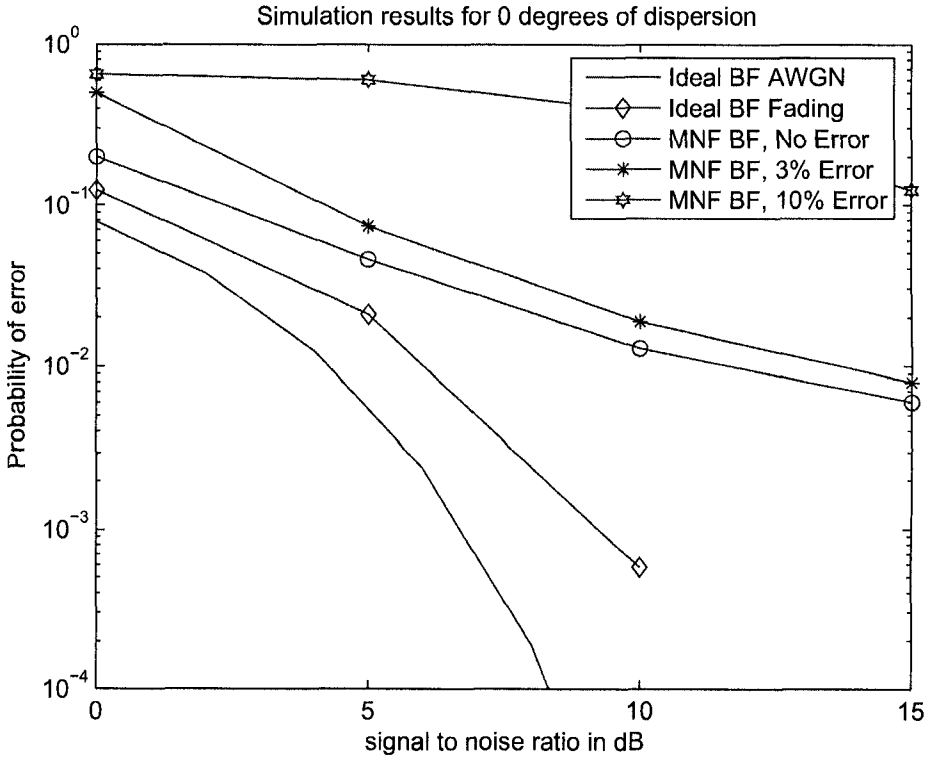


Figure 6.11: Performance simulation results for 0 degree of dispersion.

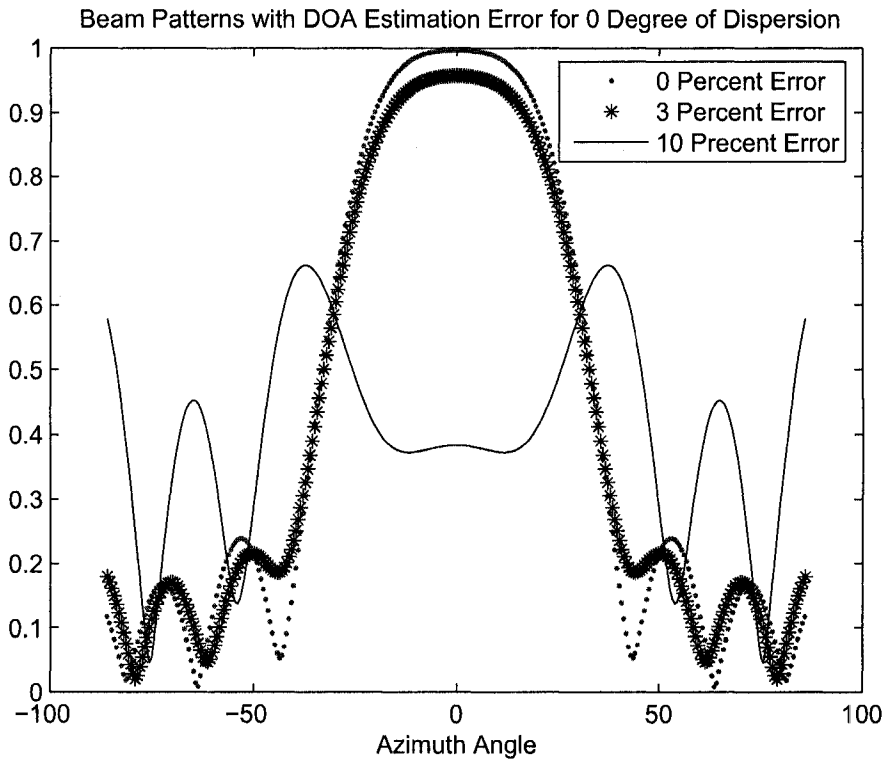


Figure 6.12: Beam patterns for 0 degree of dispersion.

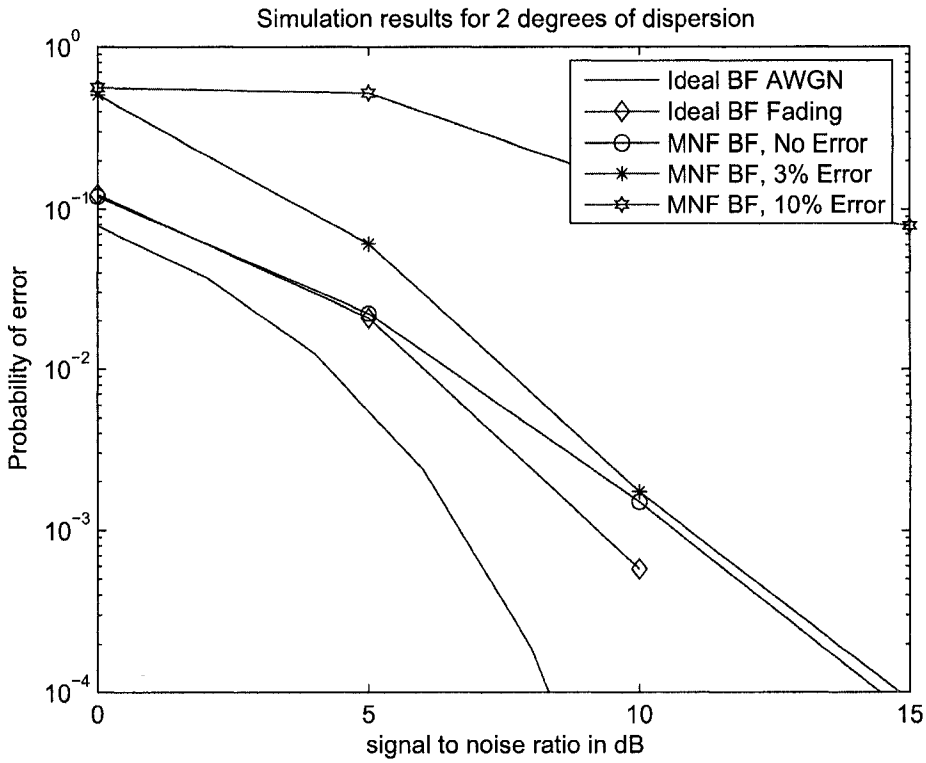


Figure 6.13: Performance simulation results for 2 degrees of dispersion.

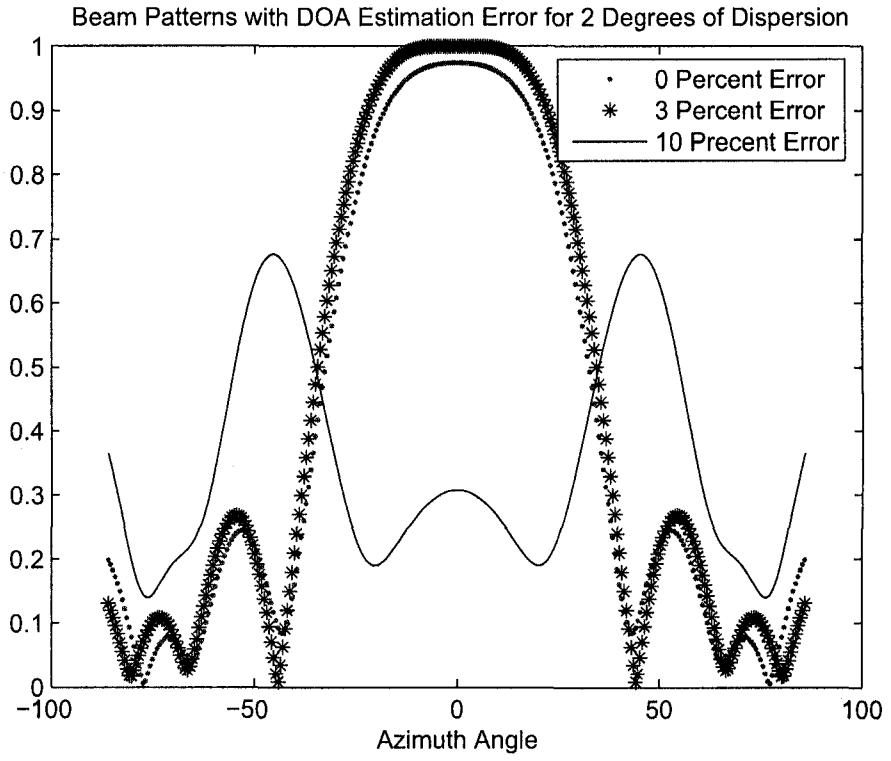


Figure 6.14: Beam patterns for 2 degrees of dispersion.

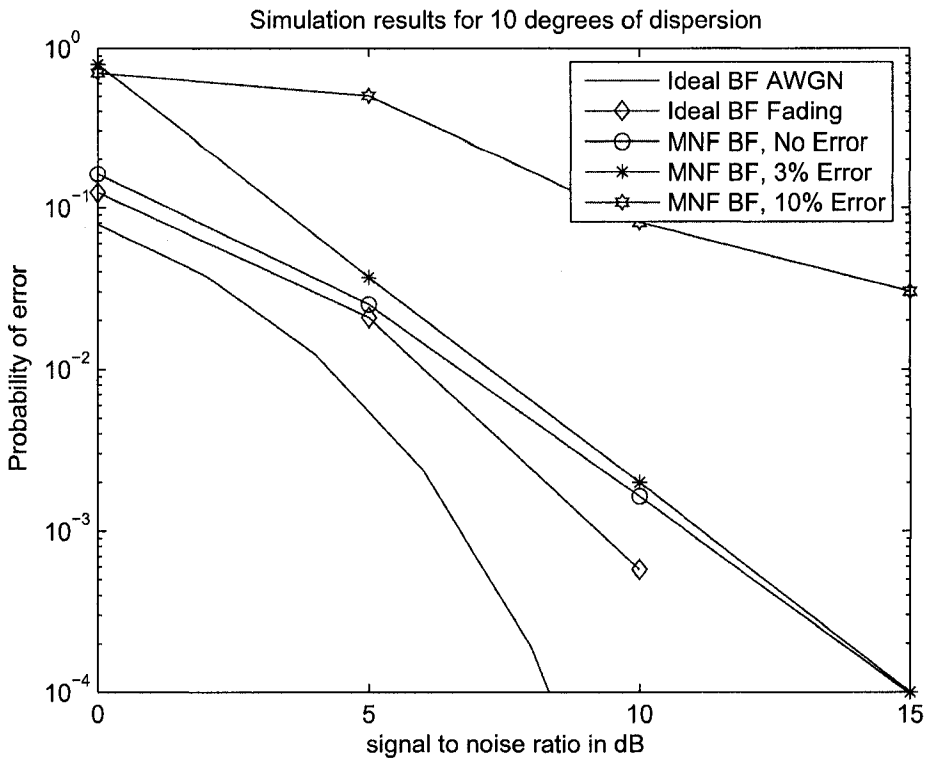


Figure 6.15: Performance simulation results for 10 degrees of dispersion.

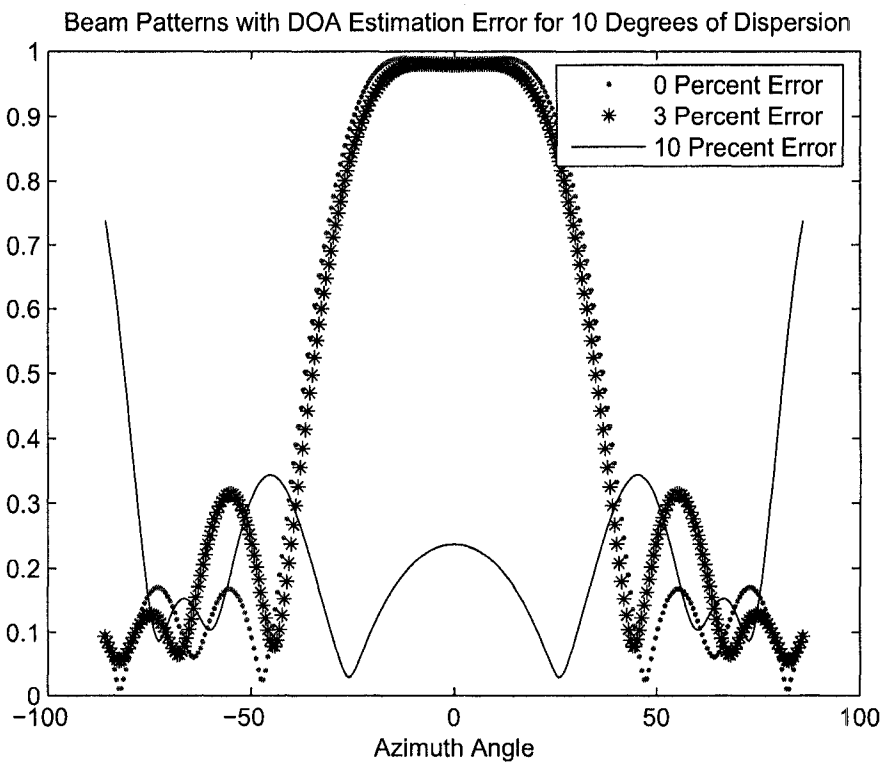


Figure 6.16: Beam patterns for 10 degrees of dispersion.

and 6.16 for estimation errors corresponding to 0% and 3% of perturbations are approximately at -40, +50 and +80 degrees, which is the actual direction of the interfering users. The interference effect of users located at +3, -4, and +5 degrees is mainly reduced via MC-CDMA scheme. It is seen that the noise covariance matrix estimation perturbations corresponding to 10% highly degrades the beam pattern and performance. Hence, noise covariance estimation errors up to 3% do not have a major effect on the performance, which represents that SFA BF technique is relatively robust with respect to this type of error. These simulations represent that SFA promises performance and separability of interfering users which leads to higher capacity of wireless future generations and key applications of wireless systems such as positioning applications.

6.9 Summary of Contributions

This work introduces the potential of SFA beamforming technique via its merger with a wide band MC-CDMA system. We introduced the details of non-overlapping window adaptive realization of SFA. We also discussed the relationship between the SFA and DOA estimation via MUSIC. SFA and DOA estimation schemes are similar as both need the estimation of the noise covariance matrix. We proposed a novel structure for wide band MC-CDMA systems that in fact utilizes the benefits of both path diversity (inherent in direct sequence CDMA) and frequency diversity (inherent in MC-CDMA systems). Simulations were performed to study the impact of perturbations on the performance of SFA. Simulations confirm that SFA promises performance and separability of interfering users which leads to higher capacity of wireless future generations.

Chapter 7

KSFA AND THE BRAIN COMPUTER INTERFACE PROBLEM

Given the promising results in Chapter 5 of KSFA on toy problems involving inherent nonlinearities, we turn our attention to the Brain Computer Interface (BCI) problem to assess the extension of SFA to KSFA. The BCI problem has the advantage that it blends a real world problem with an objective measure of performance, i.e., the classification rate, that will help us assess KSFA. For the purposes of a complete evaluation we also include results for PCA and KPCA on the BCI problem.

7.1 The BCI Problem

The Brain Computer Interface problem involves developing approaches that will allow a person to control a computer via brain waves in a non-invasive fashion [7]. Currently approaches for the BCI problem involve training subjects using bio-neuro-feedback to control their alpha and beta waves which can be readily detected by monitoring electroencephalographic (EEG) data.

We are interested in using pattern recognition approaches to classify EEG signals and to exploit this ability to assist a human to control a computer. Here we work with the EEG data set consisting of five tasks

- Resting task
- Imagined letter writing

- Mental multiplication
- Visualized counting
- Geometric object rotation

Our data sets are collected as matrices X of size 2500×7. Thus, X_{ij} corresponds to sample i at electrode j. To identify the task we will use the notation X^{μ} where $\mu = 1, \ldots, 5$. Each task has a set of ten trials, i.e., ten different instances where the subject performed the task. To indicate the trial k for task μ we use the notation X^{μ_k}. In real time each task had a duration of 10 seconds with a sampling rate of 250Hz. For more information see [50].

7.2 Signal Fraction Mapping

7.2.1 Wavelet, Fourier transformation and KSFA

Generally, most of the signals are functions of time, i.e., time-domain signals, and it is necessary to extract features to aid in its analysis. Mathematical transformations would change the original data set into a new data set to reveal information hidden in the raw data set and make the interpretation and decision making easier. There are different transformations that can be applied on the original data set. Commonly used transforms include the Fourier transform and wavelet transform [9]. Plotting the time-domain signals we obtain the time-amplitude representation which is not always the best representation of the signal. In many signals most of the information is hidden in the signal frequency content. Frequency is a measure of signal rate change. It means when the signal is changing rapidly it has a high frequency, if the changing rate is low it has a low frequency and if the signal does not change at all it has a zero frequency. The frequency can be found by taking the Fourier transformation of the signal and is measured in cycles/second, or in "Hertz". Hence, the Fourier transform gives the frequency components in the signal. The wavelet transform is a powerful tool to transform time-domain signals to a time

scale domain. In this work we merge SFA and wavelet. We first transform the data via Harr wavelet transformation and then we apply the SFA technique and we call it wavelet based SFA (WBSFA).

The notion of *signal fraction analysis* is a natural extension to the maximum noise (or signal) fraction (SFA) that promises to provide a flexible new means for extracting information out of data. As has been shown, signal fraction analysis can separate linearly mixed signals. In this setting, the estimation of the covariance matrix of the noise may be interpreted as the computation of the covariance matrix of the projection of the data onto the finest Haar wavelet subspace. As such, the signal separation may be viewed as scale based, i.e., the signals are being split into subspaces of increasing scale. So patterns with differing scales may be pulled apart.

Of course the Haar wavelet is only an example of the way data can be projected onto subspaces of differential scales. Other wavelets, such as Daubechies wavelets, have better approximation properties [66].

It has long been hypothesized that cognitive tasks effect the frequency spectrum of EEG signals. With this in mind we create new data matrices F_j that filter the EEG data based on frequencies of interest. Transformations that correspond to band pass filters of alpha waves, low beta waves, mid beta waves and high beta waves are natural candidates for this study.

7.2.2 Multi-resolution Signal Filtering

Given the above discussion, we may construct new ways to split a data set S by going beyond the calculation of the covariance matrix of the noise as

$$N^T N \approx \frac{1}{2} dX^T dX \tag{7.1}$$

We may view dX more generally as a mapping of the original data X into a subspace of interest. For example, the Daubechies wavelet D_{10} can be used via a multi-resolution analysis to produce new views of a data set X at differing scales. We may write

$$X = S_1 + W_1 \tag{7.2}$$

where the matrix S_1 is a view of X with reduced resolution in time (the transform is performed on the columns) and W_1 is the detail missing from S_1 needed to recover X. Continuing,

$$S_j = S_{j+1} + W_{j+1} \tag{7.3}$$

Thus the data matrix X may be decomposed into a sequence of matrices that encode the information at reduced scales and details.

7.2.3 Description of the Algorithm

Basically the idea now is to solve the generalized singular value problem

$$s_i^2 A^T A \psi_i = c_i^2 B^T B \psi_i \tag{7.4}$$

where now we construct the column basis via

$$\Phi = A\psi \tag{7.5}$$

In the simplest experiment we take $A = X$, i.e., the data set and $B = W_1$. Here W_1 comes from Equation (7.2).

How the resulting solutions of the GSVD problem should be used for the classification problem? Well, it appears there are many attractive options. Simplest is generalizing the procedure that we are currently using. We compute the GSVD problem on sub-windows of the data for each task. We apply the same procedure to the sub-windows of the test data to be classified. Here we applied KSFA and KPCA and K nearest neighbor (KNN) to classify the tasks.

In our programming our focus is on distinguishing between two different tasks each time that we run the program. We consider using five trials of each task, and we use the data from the first person. (Note: the original big data set includes the data for 5 different people; we just use the data from the first person). Therefore we have 5 data sets each of size 7×2500 from task i and 5 data sets of the same size from task j associated to the first person's data (here, $i = 2, 3, 4, 5$, $j = 3, 4, 5$ makes 6 combination of different task pairs, we ignored comparing the first task with others for now).

If we want to use the wavelet data to estimate the covariance of the noise we also choose the wavelet data sets with 5 trials from each of the tasks for the first person.

We calculated the mean for each defined data set, which means 10 data sets; 5 for the task i and 5 for task j. Then we took the mean subtracted data sets and used them for our entire program.

We defined data matrix containing all the 5 trials from task i, and we also defined data set containing all the 5 trials of task j. Therefore they have size 7×12500 since $5 \times 2500 = 12500$.

We also defined a data matrix containing all 5 trials of the weights from the wavelet data sets of task i and defined a data matrix containing all 5 trials of the weights from the wavelet data sets of task j both with size 7×12500. We made sliding windows of length 64. Therefore we got 195 windows for each data set we made. For each window we estimated the covariance of the noise via differencing method when we wanted to use the differencing method; and when we wanted to use the wavelet or the Fourier transformed data we just partitioned those data sets exactly as we partitioned our data sets into windows of 64 length and used them as the estimate of the noise covariance.

We called another program that used these windows as its input to find the kernel function (polynomial kernel) of size $d = 1, \ldots, 20$ where d is the degree of the polynomial kernel.

We computed the parameters for the output of the above program, i.e., each polynomial of degree d window. In fact we calculated the eigenvalue, eigenvectors for each polynomial of degree d that we computed in the previous step for KPCA case. We have used the right singular vectors as candidates for classification of the data for KPCA case, and used the generalized eigenvectors as candidates for classification of the data for KSFA case.

We calculated the generalized eigenvalue, generalized eigenvectors for each polynomial of degree d. We stored, in order, all the eigenvectors of all windows into a big matrix of size 7×1365 for KPCA and KSFA case. ($1365 = 7 \times 195$ note: number of windows are 195). Then, we defined the training set and test set as follows:

For all windows, we stored a training set of all generalized eigenvectors. The size of this matrix is 7×195 since we had 195 windows and we just collect the last mode of each window. We repeat all the steps above for all the modes and task j and we call it test sets.

Therefore we have 7 training sets for task i and 7 test sets for task j including mode$= 1, \ldots, 7$ each time for each window.

We do the same as two steps above for KPCA case using eigenvectors. We collected the first half of the above training set and the first half of the test sets into a matrix and call it as our final training set. We collected the second half of the above training set and the second half of the test sets into a matrix and call it as our final test set. (We do this for KPCA and KSFA case separately).

Hence we have made one training set and one test set for KPCA case and another training set and test set for KSFA case. For each of these sets we keep the size as 7×194.

We classified our data using k-nearest neighbors when $k = 1, \ldots, 10$ as follows.

We calculated the Euclidean distance between the training and test sets. Then we sorted the distance matrix (with size 194×194) in an increasing order and found the index for the sorted distance. Then the first row gives us the first nearest neighbor the

first and second row show the 1st and the second nearest neighbor and so on. We defined the label vector as a vector of having zeroes for the first half and having ones for the second half. The size of the label vector is 1×194.

Then we looked column-wise at the index and we labeled them as follows: if the index was less than 194/2 we labeled it 0 and otherwise we labeled it 1. After that we checked each column that we labeled with the label vector we defined previously (first half zero and next half ones).

Then we calculated the probability of error as well by finding the ratio of the number of misclassified windows and the total number. Then we counted the 0 labels we assigned for each column and the 1 labels we assigned for each column. If we had assigned more zeros than ones for this column we choose the final labeling for this column to be zero and if the number of assigned 1 labels were more than zeros we assigned 1 for this column. Therefore we are performing a data reduction from a matrix of 194×194 to a matrix of size 1×194. This was the way we made our decision about labeling each window based on the KNN number.

Then we compared this new labeled vector for the windows and the original labeling which had the first half just zeros and the second half just ones. Note that the classification is correct if the labeling for the windows matches the original labeling. Then we counted the number of windows correctly classified and we divided that by the total number to get the classification rate.

We determined our classification rates based on the ratio of the numbers of windows correctly classified and the total number. We did all these steps for both KPCA and KSFA case and we stored the results.

Therefore we made a matrix of 7×10 to store the classification rates. Note that 7 here relates to the 7 modes and 10 relates to the $k = 1, \ldots, 10$ KNN (we considered 1st nearest neighbor and second nearest neighbor and third,..., and tenth nearest neighbor).

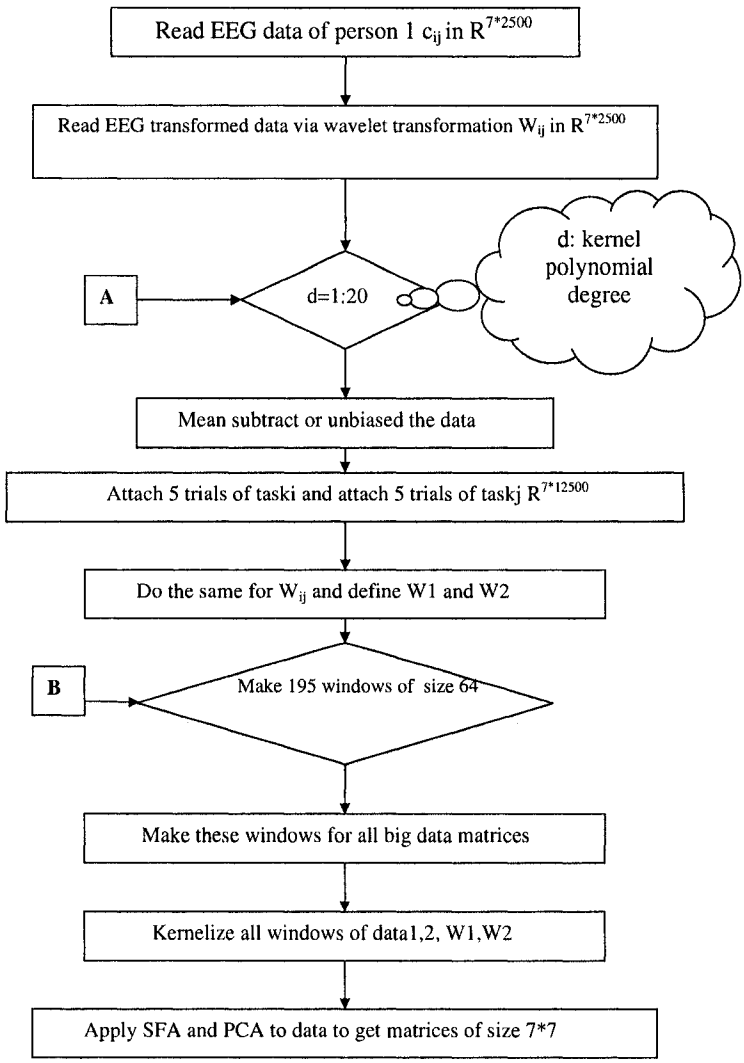
After evaluating this 7×10 matrix which includes the classification rates for all modes 1 to 7 and all knn=1:10, we calculated the maximum of each column and stored it in a vector of size 1×10, also we stored all the index for that to know for example this maximum classification rate is associated to which mode. Then we calculated the maximum of these rates $(max(max(classificationrates)))$ to find out among all these 7 modes and all these KNN nearest neighbors what the best of all classification rate is and to which mode it is associated. Then we got the best mode after finding the best classification rate.

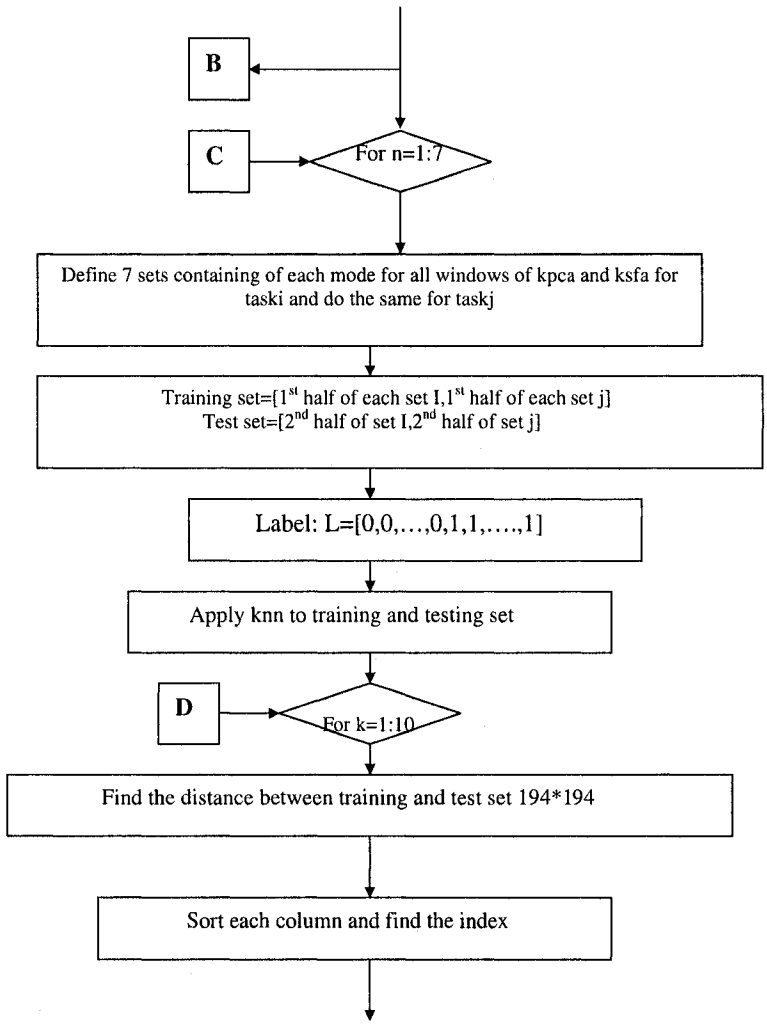
We also calculated the relative performance for both methods and we compared them in Figures 7.10, 7.11, and 7.12. To find the relative performance we found the mean of the ratio of each element in the matrix of percent correct classified in the first and second set, and we have used the right singular vectors as candidates for classification of the data for KPCA case, and the generalized eigenvectors as candidates for classification of the data for KSFA case. The flowchart of the algorithm of the work is as in Figure 7.1.

7.2.4 Results for EEG data set

We compared KSFA and KPCA for when we used different polynomial degrees $d = 1, 2, 3, 4$ to see the percent of test samples correctly classified via applying KNN classifier. Figures 7.2 - 7.9 are the results of applying KNN classifier (k=1,...,10) to the data sets task1 (resting task) and task2 (imagined letter writing) (using two trials for each task)applying KSFA for polynomial degrees $d = 1, \ldots, 4$. We could say from these figures that in the KPCA case for d=2, 3, 4 the first mode is the mode which does the best discrimination between the tasks. However in the KSFA case the best mode is the 6th mode that best classifies the tasks.

Here we have used the generalized eigenvectors as candidates for classification of the data for KSFA case. The classification rates are determined by the ratio of the number of windows correctly classified and the total number. Comparing Figures 7.2, 7.3, 7.4,





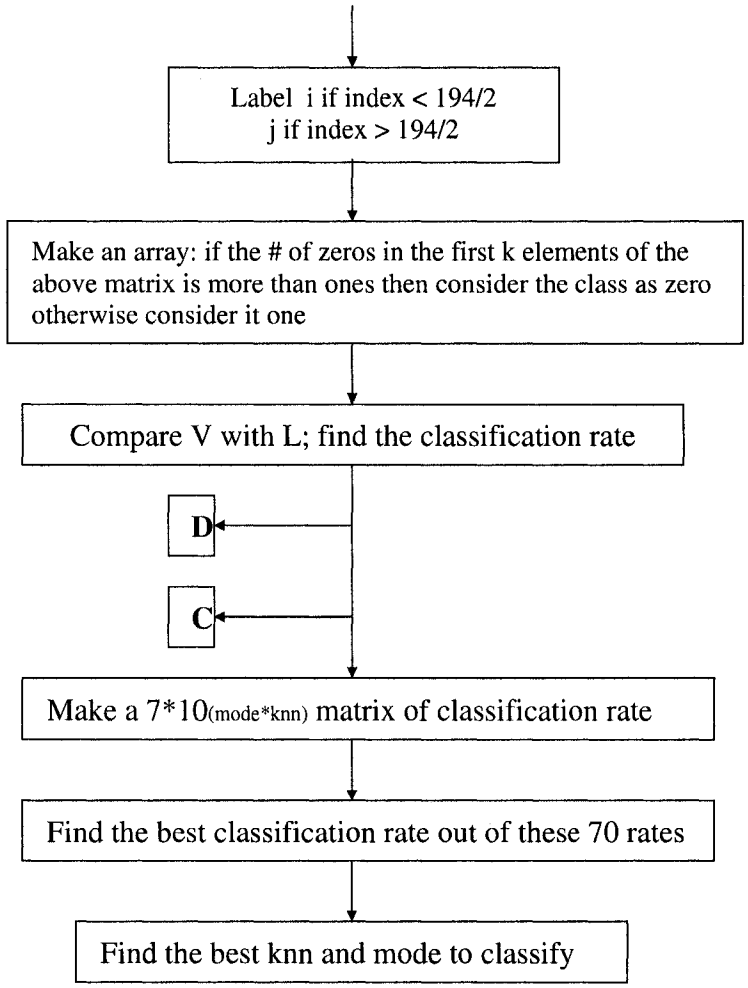


Figure 7.1: The flow chart of the algorithm.

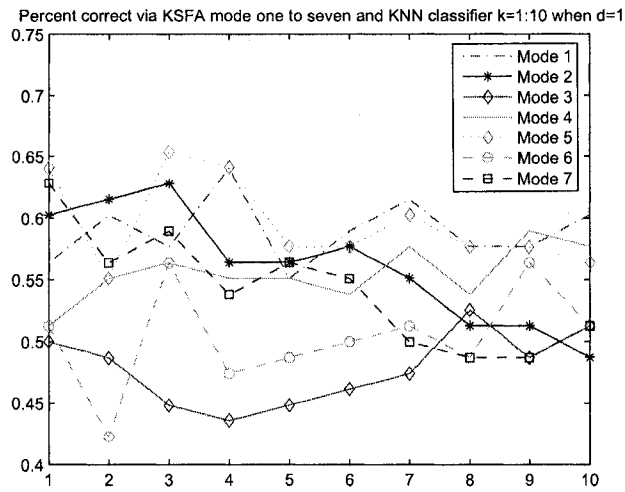


Figure 7.2: The percent classified correctly via KSFA all seven modes for when K=1:10 in KNN and when d=1.

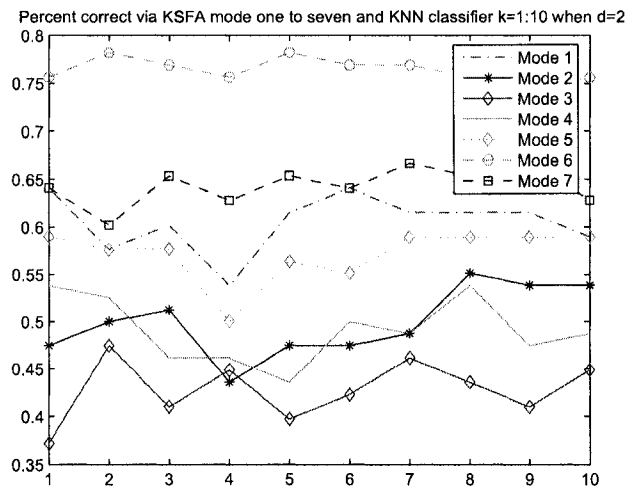


Figure 7.3: The percent classified correctly via KSFA all seven modes for when K=1:10 in KNN and when d=2.

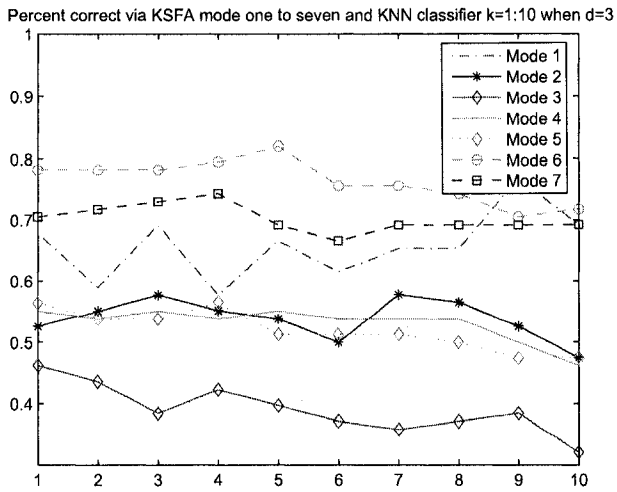


Figure 7.4: The percent classified correctly via KSFA all seven modes for when K=1:10 in KNN and when d=3.

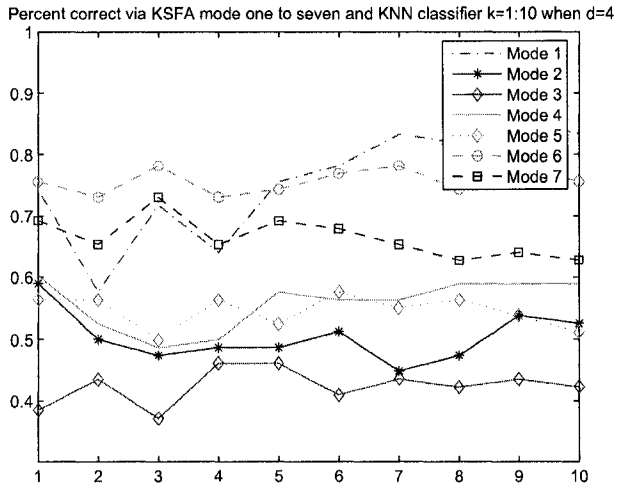


Figure 7.5: The percent classified correctly via KSFA all seven modes for when K=1:10 in KNN and when d=4.

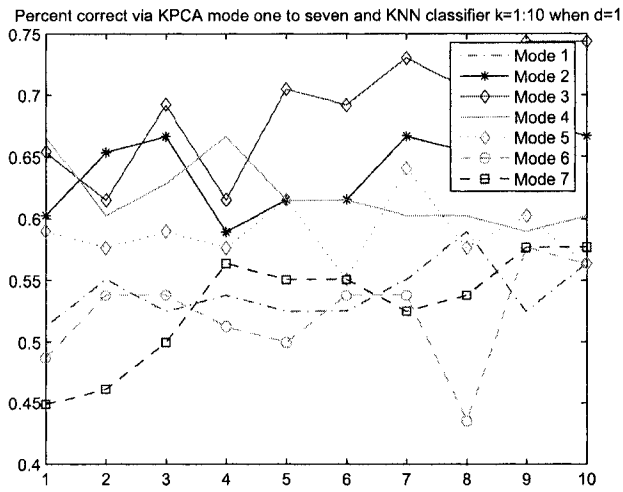


Figure 7.6: The percent classified correctly via KPCA all seven modes for when K=1:10 in KNN and when d=1.

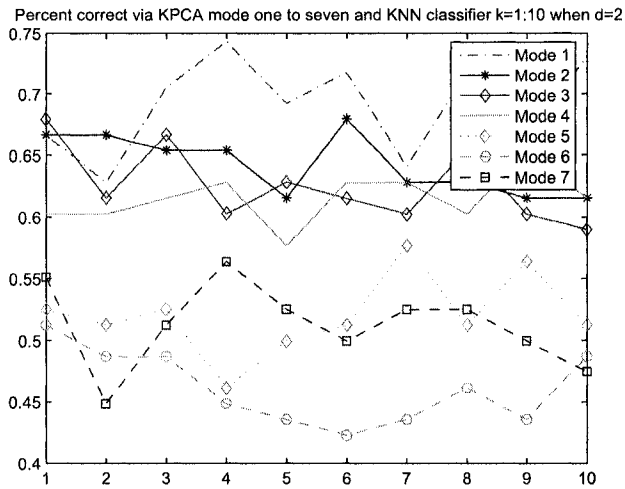


Figure 7.7: The percent classified correctly via KPCA all seven modes for when K=1:10 in KNN and when d=2.

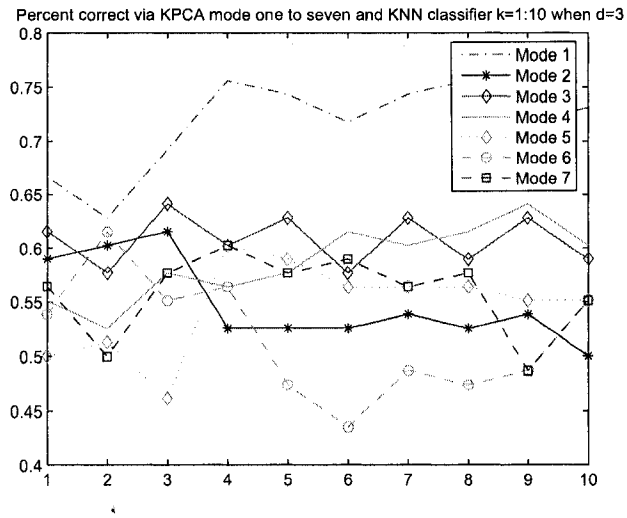


Figure 7.8: The percent classified correctly via KPCA all seven modes for when K=1:10 in KNN and when d=3.

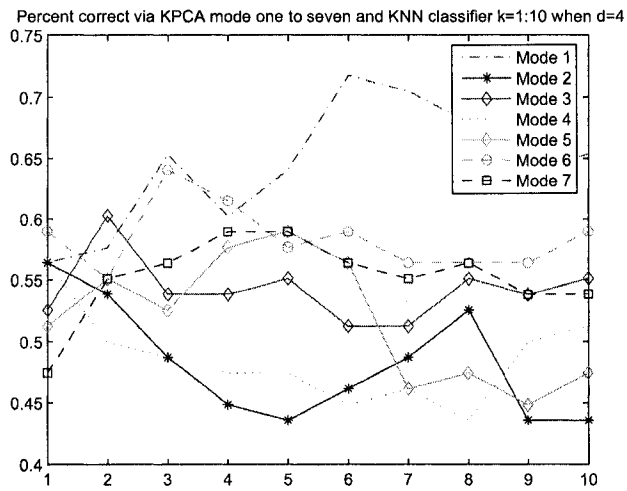


Figure 7.9: The percent classified correctly via KPCA all seven modes for when K=1:10 in KNN and when d=4.

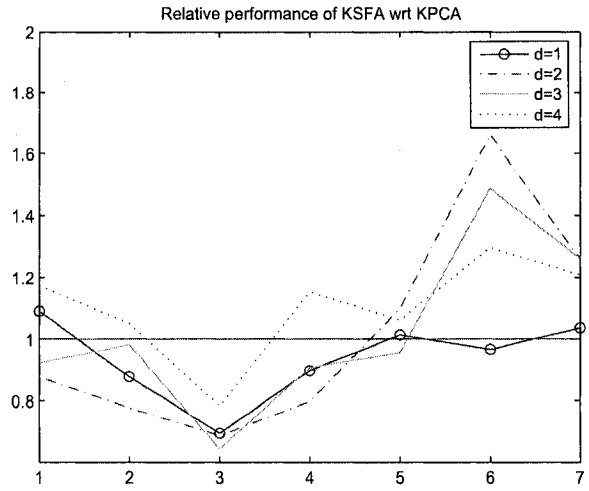


Figure 7.10: The relative performance of KSFA with respect to KPCA.

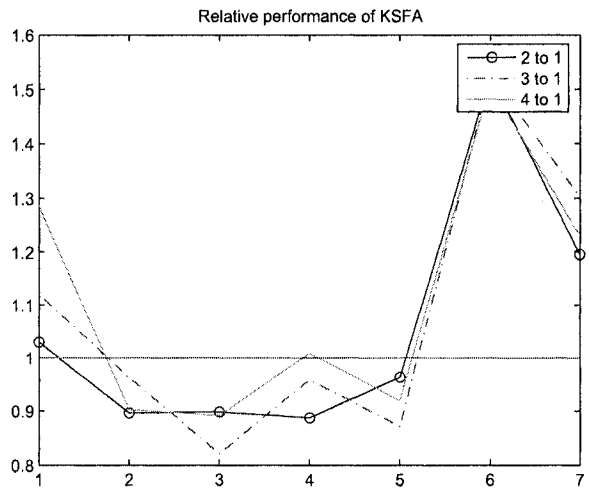


Figure 7.11: The relative performance of KSFA.

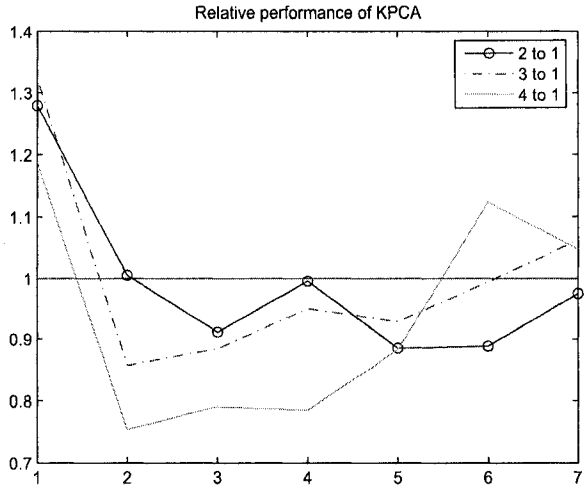


Figure 7.12: The relative performance of KPCA.

7.5 with Figures 7.6, 7.7, 7.8, 7.9 shows that KSFA could do a better job in classification than KPCA.

To further investigate the performance we calculated the relative performance for all the cases for polynomial degrees $d = 1, \ldots, 4$ and it shows in Figure 7.10 that the KSFA has a better performance than KPCA using the 6th mode. We also calculated the relative performance for KSFA for cases $d = 2$ with respect to $d = 1$ and $d = 3$ with respect to $d = 1$ and $d = 4$ with respect to $d = 1$ and we evaluated these performances for KSFA in Figure 7.11 and for KPCA in Figure 7.12.

Figure 7.13 shows the result of applying KSFA and KPCA and KNN for classifying task1 and task2 for 50 trials. Figure 7.13 shows the results for when we used db10 data set as our estimate for the noise covariance. Figure 7.14 represents the results when we used db4 as the estimation of the noise covariance, Figure 7.15 shows the result for db1, Figure 7.16 shows the result for sym10, and Figure 7.17 shows the result for when we used the Fourier transform data set as the candidate for the noise covariance estimation. It seems that Figure 7.13 which is for db10 and w1 case has the best result among other

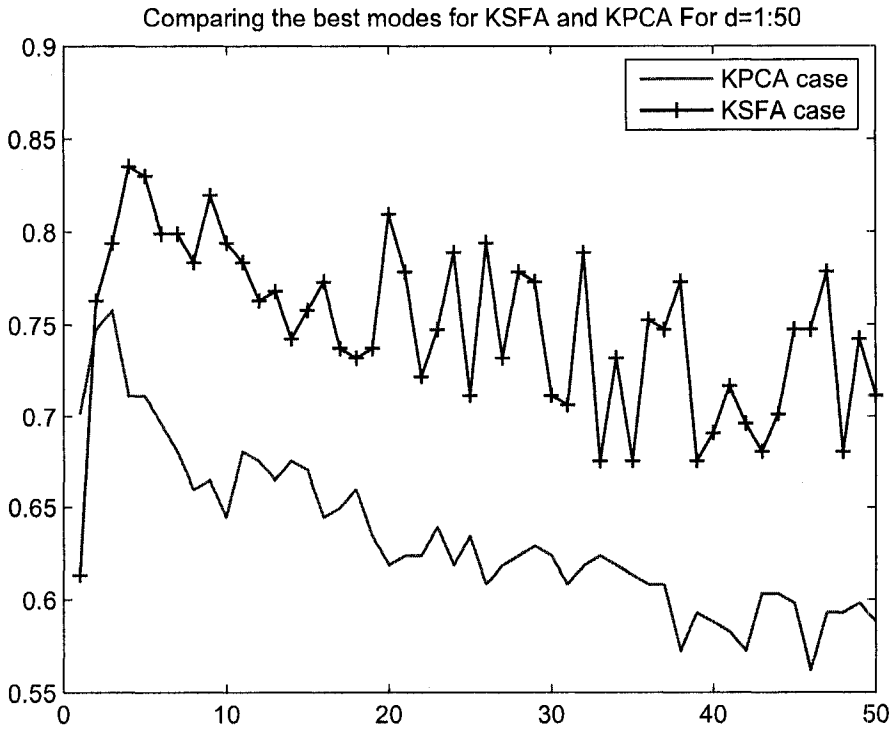


Figure 7.13: Comparing the best modes for KSFA and KPCA for d=1,...,50 applying KNN to classify tasks one and two via $db10$.

cases confirming that using d10 data set is a better choice to get a better performance in classifying the data.

Based on the Figures 7.18, and 7.19; that compares the wavelet transform with the differencing method, we could say that applying Wavelet makes the classification rate higher than applying differencing method. These figures show the comparison between the two methods differencing and wavelet to find out which rate was the maximum rate of classification. Here the first mode was the most discriminating mode.

We also did run the program for Fourier transform and all the five task pairs to find out the best mode and the best classification rate. The results were very interesting which follow:

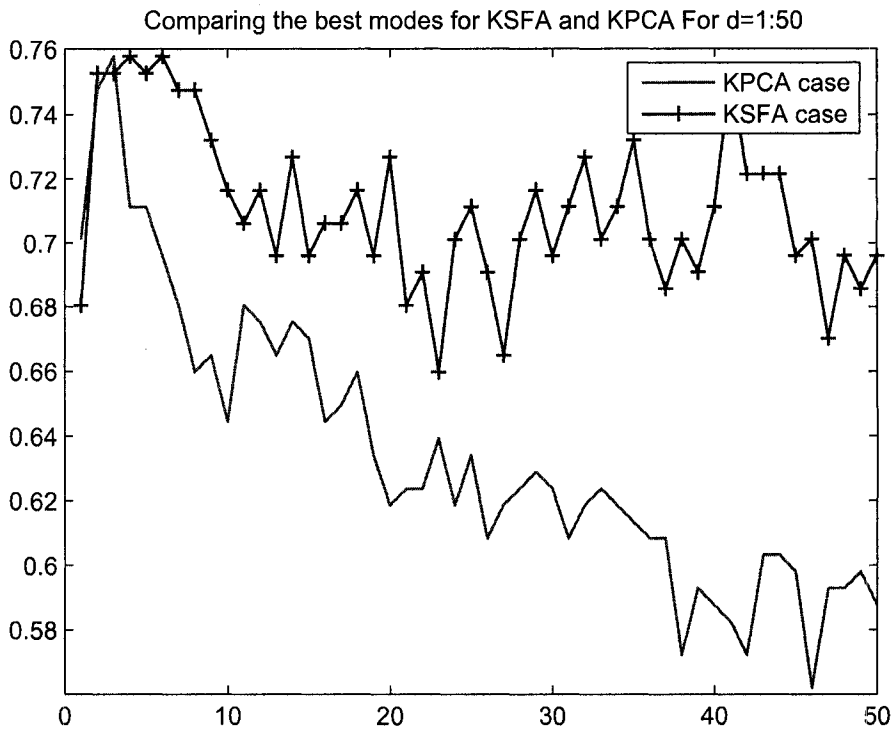


Figure 7.14: Comparing the best modes for KSFA and KPCA for d=1,...,50 applying KNN to classify tasks one and two via *db*4.

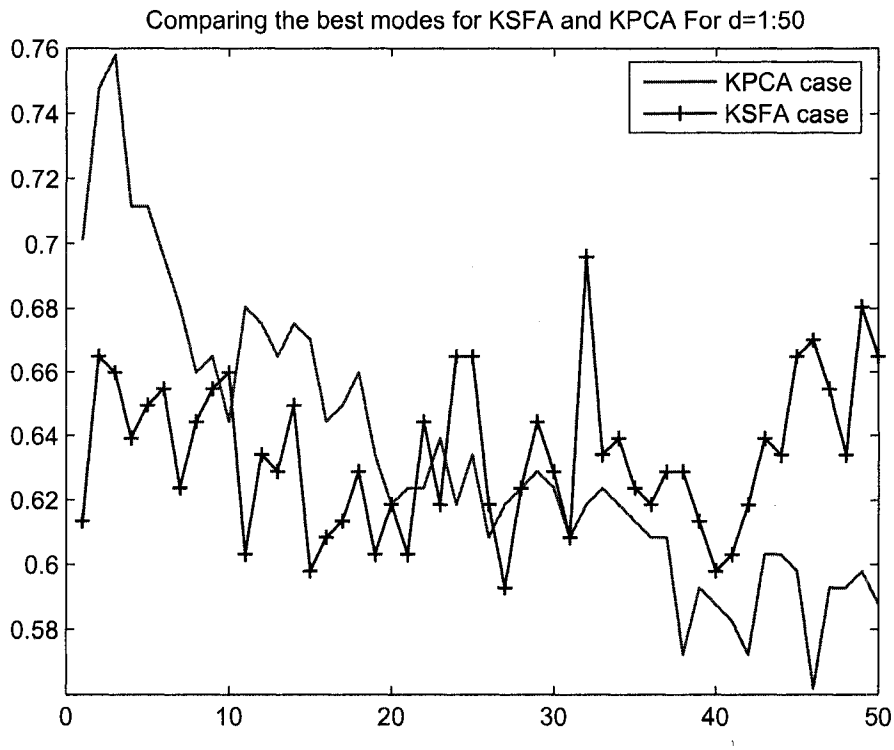


Figure 7.15: Comparing the best modes for KSFA and KPCA for d=1,...,50 applying KNN to classify tasks one and two via $db1$.

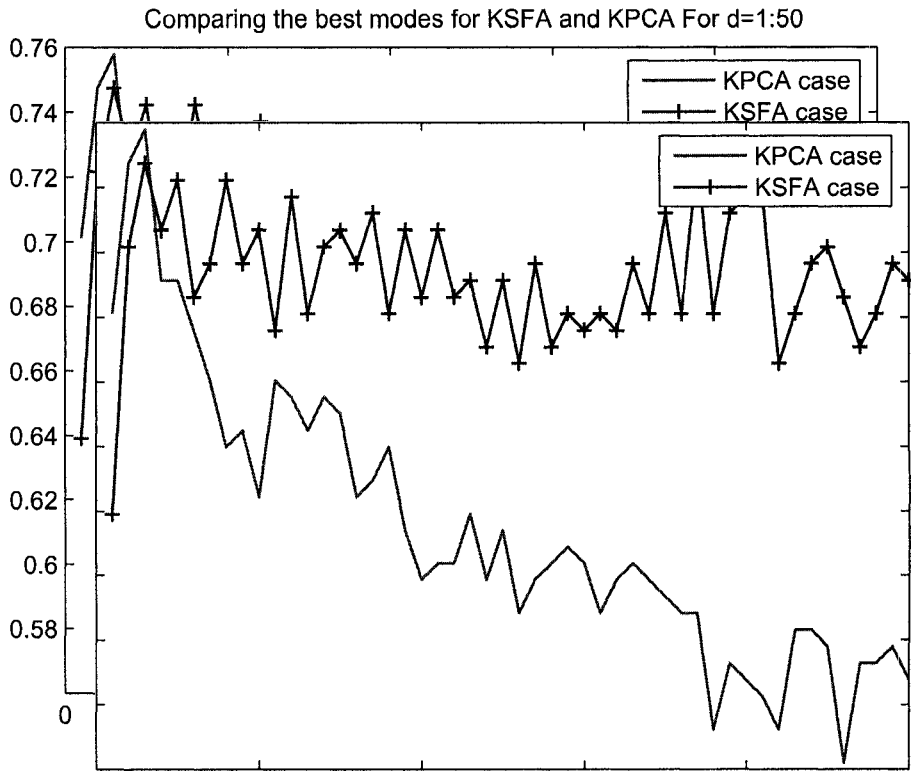


Figure 7.16: Comparing the best modes for KSFA and KPCA for d=1,...,50 applying KNN to classify tasks one and two via $sym10$.

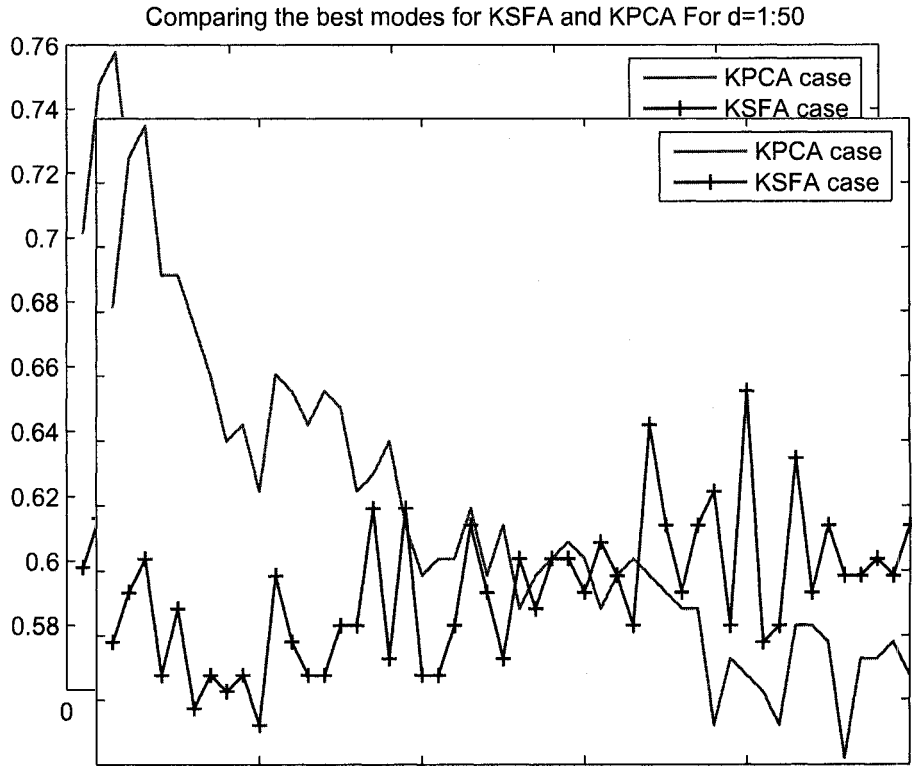


Figure 7.17: Comparing the best modes for KSFA and KPCA for d=1,...,50 applying KNN to classify tasks one and two via Fourier transform.

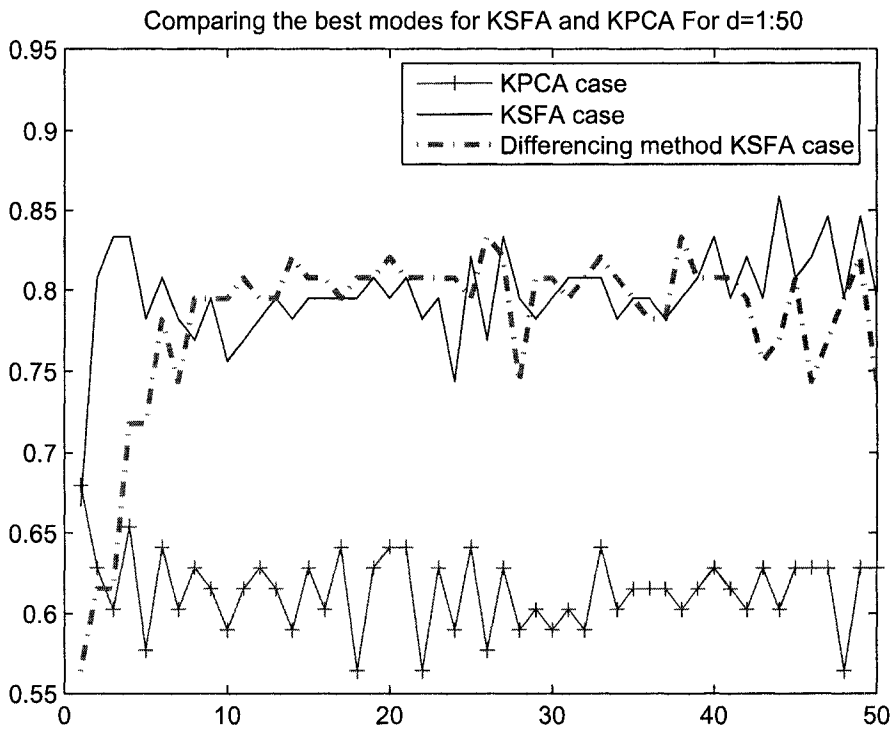


Figure 7.18: Comparing the best classification rate and the best modes for KSFA and KPCA via applying differencing and wavelet method S1 and W1.

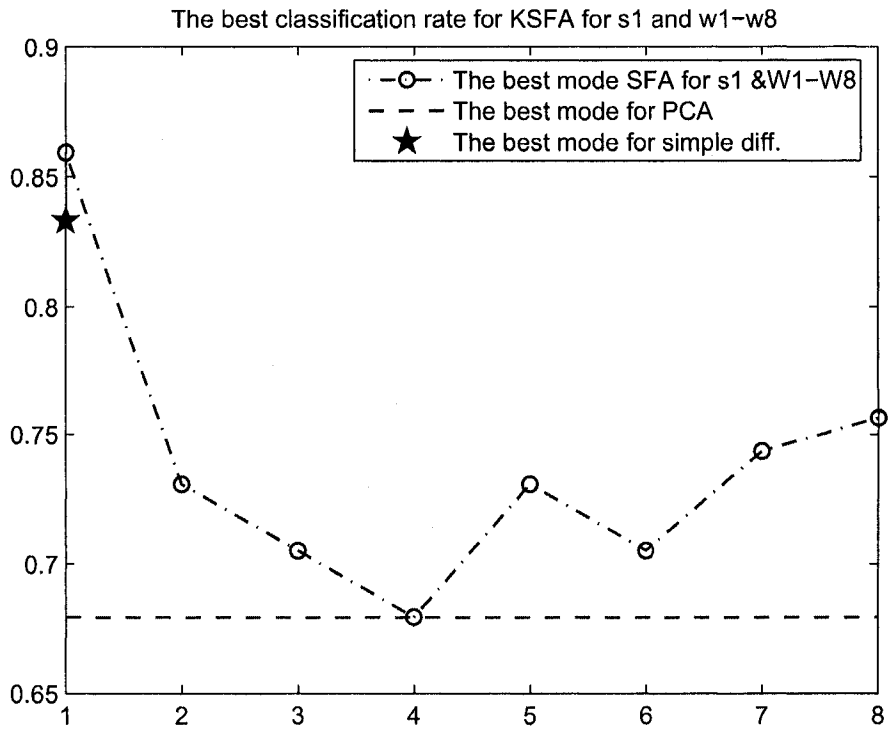


Figure 7.19: Comparing the best classification rate for KSFA and KPCA via applying differencing and wavelet method ($db10$) $S1$ and $W1$.

For all pairs of tasks (2,3), (2,4), (2,5), (3,4), (3,5), (4,5) for all alpha, beta, mid-beta, high-beta, and low-beta data sets, the best results came when we wanted to classify task 3 (mental multiplication) and task 5 (geometric object rotation). The classification rate for all the data sets for these two tasks (task 3 and 5) were about .9278 via applying PCA when $d = 1$ (d is the degree of polynomial), the best PCA mode is 3 and the KNN number equals 10.

7.3 Conclusions and Relationship to Other Work

In this work, we worked on the EEG data sets and applied KSFA and KPCA and used KNN to classify different tasks. We used different estimation of the noise covariance, such as the differencing method, and wavelet and Fourier transformations to see how they perform and we compared their relative performance. We discovered that applying wavelet makes the classification rate higher than differencing and Fourier transformation.

Appendix A

In this appendix we prove that if

$$X^T X a_i = \lambda_i N^T N a_i \tag{A-1}$$

and $N^T N$ is a positive definite matrix, then the generalized singular vectors are orthonormal with respect to $N^T N$, i.e.,

$$a_i^T N^T N a_j = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{A-2}$$

Moreover, the generalized singular vectors are orthogonal with respect to $X^T X$ i.e.,

$$a_i^T X^T X a_j = \begin{cases} \lambda_j & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{A-3}$$

Proof

For simplicity we define

$$A = X^T X \tag{A-4}$$

and

$$B = N^T N \tag{A-5}$$

It is clear that $A^T = A$ and $B^t = B$ which means A and B are Hermitian. Therefore, equation A-1 simplifies to

$$A a_i = \lambda_i B a_i \tag{A-6}$$

Thus we need to prove

$$a_i^T B a_j = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{A-7}$$

and

$$a_i^T A a_j = \begin{cases} \lambda_j & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{A-8}$$

Suppose for $i \neq j$ that $\lambda_i \neq \lambda_j$ be two distinct generalized singular values with corresponding generalized singular vectors a_i and a_j respectively. Therefore, for two cases i and j when $i \neq j$ we use A-6 and A-9 respectively

$$A a_j = \lambda_j B a_j \tag{A-9}$$

If we pre-multiply equation A-6 by a_j^T and equation A-9 by a_i^T we respectively obtain

$$a_j^T A a_i = \lambda_i a_j^T B a_i \tag{A-10}$$

$$a_i^T A a_j = \lambda_j a_i^T B a_j \tag{A-11}$$

From equation A-10 we get

$$(a_j^T A a_i)^T = (\lambda_i a_j^T B a_i)^T \tag{A-12}$$

and since A and B are Hermitian A-12 results

$$a_i^T A a_j = \lambda_i a_i^T B a_j \tag{A-13}$$

Now subtracting equation A-11 from equation A-13 leads to

$$(\lambda_i - \lambda_j) a_i^T B a_j = 0 \tag{A-14}$$

Since the generalized singular values are distinct (i.e., $\lambda_i \neq \lambda_j$) we conclude that

$$a_i^T B a_j = 0 \quad for \quad i \neq j \tag{A-15}$$

If we substitute A-15 into A-10 we obtain

$$a_i^T A a_j = 0 \quad for \quad i \neq j \tag{A-16}$$

Since A and B are Hermitian, and B is positive definite, Equation (A-6) has p linearly independent generalized singular vectors (see the proof in Appendix B). Here, we indicate the generalized singular vectors by $a_1, a_2, \ldots, a_p$. Suppose C is an arbitrary vector as a linear combination of these generalized singular vectors

$$C = c_1 a_1 + c_2 a_2 + \ldots + c_i a_i + \ldots + c_p a_p \tag{A-17}$$

If we pre-multiply Equation (A-17) by $a_i^T B$ we get

$$a_i^T BC = c_1 a_i^T B a_1 + c_2 a_i^T B a_2 + \ldots + c_i a_i^T B a_i + \ldots + c_p a_i^T B a_p \tag{A-18}$$

From Equation (A-15) which is the orthogonal property of the generalized singular vectors with respect to B, it follows that except one term $c_i a_i^T B a_i$ in the right hand side of A-18, all other terms are zero. Therefore:

$$a_i^T BC = c_i a_i^T B a_i \tag{A-19}$$

Moreover, since B is positive definite then $a_i^T B a_i \neq 0$. If we solve A-19 for $a_i^T B a_i$ we obtain

$$a_i^T B a_i = \frac{a_i^T BC}{c_i} \tag{A-20}$$

since C was arbitrary, assuming $c_i = a_i^T BC$ results in

$$a_i^T B a_i = 1 \tag{A-21}$$

In Equation (A-10) if $i = j$ we get

$$a_i^T A a_i = \lambda_i a_i^T B a_i \tag{A-22}$$

If we substitute A-21 in A-22 we obtain

$$a_i^T A a_i = \lambda_i \tag{A-23}$$

which results in Equations (A-15), (A-16), (A-21), and (A-23) complete the proof.

Appendix B

If A and B are Hermitian matrices, B is positive definite, and if $a_1, a_2, \ldots, a_p$ and $\lambda_1, \lambda_2, \ldots, \lambda_p$ are corresponding generalized singular vectors and distinct generalized singular values respectively for $Aa_i = \lambda_i Ba_i$, then $a_1, a_2, \ldots, a_p$ are linearly independent.

Proof

Suppose that $a_1, a_2, \ldots, a_p$ are linearly dependent, then in the following linear combination of these generalized singular vectors

$$c_1a_1 + c_2a_2 + \ldots + c_ia_i + \ldots + c_pa_p = 0 \qquad \text{(B -1)}$$

at least one of the c_i's must be some non-zero. If we suppose that $c_i \neq 0$ is the one and we pre-multiply equation B -1 by $a_i^T B$ we get

$$c_1a_i^TBa_1 + c_2a_i^TBa_2 + \ldots + c_ia_i^TBa_i + \ldots + c_pa_i^TBa_p = 0 \qquad \text{(B -2)}$$

From Equation (A-15) we know that all the terms in (B -2) is zero except the term $c_ia_i^TBa_i$. Therefore Equations (B -2) and (A-15) leads to

$$c_ia_i^TBa_i = 0 \qquad \text{(B -3)}$$

Which means $c_i = 0$ that shows the contrary with the assumption that we had for linear dependency (i.e., $c_i \neq 0$). From this contradiction we conclude that the generalized singular vectors $a_1, a_2, \ldots, a_p$ are linearly independent.

Bibliography

[1] A. Adams and J. F. Fournier. *Sobolev Spaces*, volume 140 of *Pure and Applied Mathematics*. Academic Press, Boston, 2003.

[2] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[3] M. Akhtar and J. Asenstorfer. Iterative detection for MC-CDMA system with base station antenna array for fading channels. *proceedings of IEEE Globecom'98*, pages 241–246, 1998.

[4] S. M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE Journal on Selected areas in Communications*, 16(8):1451–1458, 1998.

[5] M. Anderle and M. Kirby. The application of the maximum noise fraction method to filtering noisy time-series. In J.G. McWhirter and I.K. Proudler, editors, *Mathematics in Signal Processing V*, The Institute of Mathematics and Its Applications Conference Series, pages 223–237. Oxford University Press, 2002.

[6] M. G. Anderle. *Modeling Geometric Structure in Noisy Data.* PhD dissertation, Colorado State University, 2001.

[7] C.W. Anderson and M.J. Kirby. EEG subspace representations and feature selection for brain-computer interfaces. In *Proceedings of the 1st IEEE Workshop on Computer Vision and Pattern Recognition for Human Computer Interaction (CVPRHCI)*, Madison, Wisconsin, June 2003.

[8] B. Ans, J Herault, and C. Jutten. Adaptive neural architectures: detection of primitives. *In Proc. Of COGNITIVA*, pages 593–597, 1985.

[9] R. Azencott, J.P. Wang, and L. Younes. Texture classification using windowed fourier filters. *PAMI*, 19(2):148–153, February 1997.

[10] A. K. Aziz, Babŭska, and M. Ivo. Part I , survey lectures on the mathematical foundations of the finite element method. In A.K. Aziz, editor, *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, pages 1–362. Academic Press, New York, 1972.

[11] Z. Bai. The CSD, GSVD, their applications and computations. Technical Report IMA Preprint Series 958, Minneapolis, MN, 1992.

[12] De Moor Bart, Jan Staar, and Joos Vandewalle. Oriented energy and oriented signal-to-signal ration concepts in the analysis of vector sequences and time series. In *SVD and Signal Processing, Algorithms, Applications and Architectures*, pages 209–231, 1988.

[13] A. Belouchrani, K. A. Meraim, J. F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEETRSP*, 45(2):434–44, 1997.

[14] M. Berman. The statistical properties of three noise removal procedures for multichannel remotely sensed data. Technical Report 8531, CSIRO division of mathematics and statistics consulting rep, 1985.

[15] R.C. Bernhardt. Time-slot management in frequency reuse digital portable radio systems. *proceedings IEEE Vehicular Technology Conference, VTC'90*, pages 282–286, 1990.

[16] R.C. Bernhardt. Call performance in a frequency reuse digital portable radio system. *IEEE Transactions on Vehicular Technology*, 40(4):777–785, Nov. 1991.

[17] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13(1):1–10, 2000.

[18] James H. Bramble and Xeujun Zhang. The analysis of multigrid methods. In *Techniques of Scientific Computing (Part 3)*, volume VII of *Handbook of Numerical Analysis*, pages 173–415. North-Holland, Amsterdam, The Netherlands, 2000.

[19] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[20] C. J. C. Burges. Geometry and invariance in kernel based methods. *Advances in kernel methods- support vector learning*, pages 89–116, 1999.

[21] J. F. Cardoso. Blind identification of independent signals. *In Proc. Workshop on Higher Order Spectral Analysis, Vail, Colorado*, 1989.

[22] C. Chang and Q. Du. A canonical correlation approach to exploratory data analysis in fMRI. *IEEE Transactions on Geoscience and Remote sensin*, 37, No. 5, September 1999.

[23] V. Cherkassky and F. Mulier. *Learning from data concepts, theory and methods.* John Wiley and Sons, New York, 1998.

[24] M.T. Chu, R.E. Funderlic, and G.H. Golub. On a variational formulation of the generalized singular value decomposition. *SIAM J. Matrix Anal. and App.*, 18(4):1082–1092, 1997.

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Willey and Sons, New York, N. Y., 1991.

[26] James Demmel and Krešimir Veselić. Jacobi's method is more accurate than QR. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1204–1245, 1992.

[27] G. Dornhege, B. Blankertz, and G. Curio. Speeding up classification of multi-channel brain-computer interfaces: Common spatial patterns for slow cortical potentials. In *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, pages 591–594, 2003.

[28] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA PNAS*, 95:14863–14868, Dec. 1998.

[29] F. Emdad, S. A. Zekavat, and M. Kirby. Adaptive antenna beam forming via maximum noise fraction for multi carrier CDMA systems. *ICWN*, pages 431–437, 2003.

[30] Evans, James R., and Andrew Abarbanel. *Introduction to Quantitative EEG and Neurofeedback.* Academic Press, New York, N.Y, 1999.

[31] C. f. Van Loan. Generalizing the singular value decomposition. *SIAM J. Numer. Anal.*, 13:76–83, 1976.

[32] K. Fukunuaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, 1972.

[33] G.H. Golub and C.F. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1996.

[34] C. Gordon. A generalization of the maximum noise fraction transformation. *IEEE transactions on geoscience and remote sensing*, 38(1), January 2000.

[35] A.A. Green, M. Berman, P. Switzer, and M.D. Craig. A transformation for odering multispectral data in terms of image quality with implications for noise removal. *IEEE Transactions on Geoscience and Remote Sensing*, 26(1):65–74, January 1988.

[36] C. Guger, H. Ramoser, and G. Pfurtscheller. Real-time eeg analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Trans Rehabil Eng.*, 8(4):447–56, Dec. 2000.

[37] S. Hara and R. Prasad. Overview of multi-carrier CDMA. *IEEE Communications Magazine*, 35(12):126–133, Dec. 1997.

[38] W. R. Heath and A. Paulraj. Transmit diversity using decision-directed antenna hopping. *IEEE Communications Mini-Conference*, pages 141–145, June 1999.

[39] J. Herault and B. Ans. Circuits neuronaux a synapses modifiables: decodage de messages composites par apprentissage non supervise. *C.-R. de l'Academiedes Sciences*, 229(III-13):525–528, 1984.

[40] J. Herault, C. Jutten, and B. Ans. Detection de grandeurs primitives dens un message composite par une architecture de calcul neuromimetique en apprentissage non supervise. *In Actes du Xeme colloque GRETSI*, pages 1017–1022, 1985.

[41] D. Hundley, M. Kirby, and Markus Anderle. A solution procedure for blind signal separation using the maximum noise fraction approach: Algorithms and examples. *Proceedings of the Conference on Independent Component Analysis*, pages 337–342, December 2001.

[42] D. R. Hundley, Michael J. Kirby, and Markus Anderle. Blind source separation using the maximum signal fraction approach. *Signal Processing*, 82(10):1505–1508, October 2002.

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

[44] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Sons, New York, 2001.

[45] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[46] J.W. C. Jakes. *Microwave Mobile Communications.* New York, Wiley, 1974.

[47] Li Jian, P. Stoica, and Wang Zhisong. On robust capon beamforming and diagonal loading. *IEEE Transactions on Signal Processing*, 51(7):1702–1715, July 2003.

[48] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, Englewood Cliffs N.J., 1982.

[49] I.T. Jolliffe. *Principal Component Analysis.* Springer, New York, 1986.

[50] Z.A. Keirn and J.I. Aunon. A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering*, 37(12):1209–1214, December 1990.

[51] M. Kirby. *Geometric Data Analysis: An empirical approach to dimensionality reduction and the study of patterns.* John Wiley and Sons, New York, N.Y, 2001.

[52] Michael Kirby and Chuck Anderson. Geometric analysis for the characterization of nonstationary time-series. In Ehud Kaplan, Jerry Marsden, and Katepalli R. Sreenivasan, editors, *Perspectives and Problems in Nonlinear Science: A Celebratory Volume in Honor of Larry Sirovich*, Springer Applied Mathematical Sciences Series. Springer-Verlag, 2003.

[53] J. Knight. Signal fraction analysis and artifact removal in eeg. Master's thesis, Colorado State University, 2003.

[54] Z. J. Koles. The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroenceph. Clin. Neurophysiol*, 79(6):440–447, Dec. 1991.

[55] Z. J. Koles and A. C. K. Soong. Eeg soure localization: implementing the spatio-temporal decomposition approach. *Electroencephalogr. Clin. Neu-rophysiol*, 107(5):343–352, 1998.

[56] K. Kuchi, R. Pirhonen, and R. Srinivasan. Hopped delay diversity for EDGE. *proceedings IEEE Vehicular Technology Conference, VTC'01*, pages 152–156, May 2001.

[57] M. Kuwahara and N. Doi. Code reassignment scheme on CDMA smart antenna system. *proceedings IEEE Vehicular Technology Conference, VTC'2000*, pages 2137–2141, 2000.

[58] J.B. Lee, A.S. Woodyatt, and M. Berman. Enhancement of high spectral resolution remote sensing data by a noise-adjusted principal component transform. *IEEE Transactions on Geoscience and Remote Sensing*, 28(3):295–304, May 1990.

[59] T. Lee. Nonlinear approaches to independent component analysis. *Processings of the American institute of physics*, 1999.

[60] T. Lee, B. Koehler, and R. Orglmeister. Blind source separation in nonlinear mixing models. *in neural networks for signal processing*, 1997.

[61] K. K. Leung, J. H. Wnters, L. J. Cimini, and Jr. Interference estimation in presence of noise for broadband wireless packet networks. *IEEE VTC*, 2001.

[62] R. Li. Bounds on perturbations of generalized singular values and associated subspaces. *SIAM J. Mat. Anal. Apl.*, 14(1):195–234, 1993.

[63] J.C. Liberti, Jr., and T.S. Rappaport. *Smart Antennas for Wireless Communications: IS-95 and third generation CDMA applications.* Prentice Hall, Upper Saddle River, NJ, 1999.

[64] C. Van Loan. Computing the CS and the generalized singular value decomposition. *Numer. Math.*, 46:479–491, 1985.

[65] Olson Luke. *Multilevel Least-Squares Finite Element Methods for Hyperbolic Partial Differential Equations.* Ph.D. dissertation, University of Colorado, Boulder, 2003.

[66] S. Z. Mahmoodabadi, A. Ahmadian, and M. D. Abolhasani. Ecg feature extraction using daubechies wavelets. *Proceeding of the fifth lasted international conference, visualization, imaging, and image processing*, pages 343–348, September 7-9 2005.

[67] O. L. Mangasarian. *Generalized Support Vector Machines*, chapter Advances in large margin classifiers, pages 135–146. Cambridge, MA, 2000.

[68] H. Martens, M. Hoy, B. M. Wise, R. Bro, and P. B. Brockhoff. Pre-whitening of data by covariance-weighted pre-processing. *Journal of Chemometrics*, 17:153–165, 2003.

[69] S. Martin. *Techniques in Support Vector Classification.* dissertation, Colorado State University, Colorado State University, Fort Collins, CO, Spring 2001.

[70] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. *IEEE Neural Networks for Signal Processing Workshop*, pages 41–48, 1999.

[71] S. Mika, B. Scholkopf, A. Smola, K. Muller, M. scolz, and G. Ratsh. Kernel pca and de-noising in feature spaces. *Neural computation NIPS*, 1998.

[72] T. Minn and Kai-Yeung Siu. Dynamic assignment of orthogonal variable-spreading-factor codes in W-CDMA. *IEEE Journal on Selected Areas in Communications*, 18(8):1429–1440, August 2000.

[73] B. L.R. De Moor and G. H. Golub. Generalized singular value decompositions: A proposal for a standardized nomenclature. Technical report, Numerical analysis ESAT-SISTA Rep. 1989-10, 15 pp., Dep. Elec. Eng., Katholieke Universiteit Leuven, Belgium; Numer. Anal. Proj. Manuscript NA-89-04, Dep. Comput. Sci., Stanford Univ., Stanford, CA., April 1989.

[74] A. W. Mueller and D. A. Hilton. Statistical comparisons of aircraft flyover noise adjustment procedures for different weather conditions. Technical report, May 1979.

[75] Johannes Müller-Gerking, Gert Pfurtscheller, and Henrik Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. Electroenc. Clin. Neurophys., 1999.

[76] H. Olofsson, M. Almgren, and M. Hookin. Transmitter diversity with antenna hopping for wireless communication systems. *proceedings IEEE Vehicular Technology Conference, VTC'97*, 3:1743–1747, May 1997.

[77] S. Onoe and S. Yasuda. Flexible re-use for dynamic channel assignment in mobile radio systems. *proceedings IEEE International Conference on Communications, ICC'89*, 1:472–476, 1989.

[78] K. Pahlavan and P. Krishnamurthy. *Principles of Wireless Networks - A Unified Approach.* Prentice Hall Upper, Saddle River, NJ,, 2002.

[79] C. C. Paige. A note on a result of sun ji-guang: sensitivity of the CS and GSV decomposition. *SIAM J. Numer. Anal.*, 21(1):186–191, Feb. 1984.

[80] C. C. Paige and M. A. Saunders. Towards a generalized singular value decomposition. *SIAM J. Numer. Anal.*, (18):398–405, 1981.

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

[82] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehab. Eng.*, 8:441–446, December 2000.

[83] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehab. Eng.*, 8(4):441–446, 2000.

[84] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, Upper Saddle River, NJ, 1996.

[85] S. R. Saunders. *Antennas and Propagation for Wireless Communication Systems.* John Wiley and Sons, New York, NY, 1999.

[86] B. Scholkopf, A.J. Smola, and K.-R.Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[87] S. Shahbazpanahi, A. B. Gershman, Luo Zhi-Quan, and Wong M. Kon. Robust adaptive beamforming for general-rank signal models. *IEEE Transactions on Signal Processing*, 51(9):2257–2269, Sept. 2003.

[88] P. Mohana Shankar. *Introduction to Wireless Systems.* John Wiley and Sons, New York, NY, 2002.

[89] J. M. Speiser and C. Van Loan. Signal processing computations using the generalized singular value decomposition. *SPIE Real time signal processing VII.*, 495, 1984.

[90] M. Stege, T. Ruprich, M. Bronzel, and G Fettweis. Channel estimation using long-term spatial channel characteristics. *WPMC'01*, September 2001.

[91] G. W. Stewart. Computing the CS decomposition of a partitioned orthonormal matrix. *Numer. Math.*, (40):297–306, 1982.

[92] P. Switzer and A. Green. Min/max autocorrelation factors for multivariate spatial imagery. Technical report, Stanford university, Department of statistics, Stanford university, Department of statistics, 1984.

[93] A. Taleb and C. Jutten. Source separation in post nonlinear mixtures. *IEEE Trans. On signal processing*, 47:2807–2820, 1999.

[94] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra.* SIAM, 1997.

[95] T. Tu, C. Lee, C. Chiang, and C. Chang. A visual disk approach for determining data dimensionality in hyperspectral imagery. *Proc. Natl. Sci. Counc. ROC(A)*, 25(4):219–231, 2001.

[96] D. Tufts and R. Kumaresan. Frequency estimation of multiple sinusoids: making linear prediction like maximum likelihood. *Proc IEEE 70*, pages 975–990, March 1983.

[97] R. J. Vaccaro, D. W. Tufts, and G. F. Boudreaux-Bartels. Advances in principal component signal processing. *SVD and signal processing E. F. Deprettere ed. Amsterdam: North-Holland*, pages 115–146, 1988.

[98] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer-Verlag Series in Statistics, 1982.

[99] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[100] V. N. Vapnik. *Statistical Learning Theory (Adaptive and Learning Systems for Signal Processing, Communication and Control).* Wiley & Sons, 1998.

[101] I. Viering, T. Frey, and G. Schnabl. Hybrid beamforming: Reduced eigen beamforming on beam signals. *IEEE Broadband communications*, pages 91–96, 2002.

[102] A. J. Viterbi. *CDMA, Principles of Spread Spectrum.* Addison-Wesley, Reading, MS, 1995.

[103] Yijun Wang, Shangkai Gao, and Xiaornog Gao. Common spatial pattern method for channel selelction in motor imagery based brain-computer interface. In *IEEE-EMBS 2005. 27th Annual International Conference of the Medicine and Biology Society*, pages 5392–5395, 2005.

[104] Yijun Wang, Shangkai Gao, and Xiaornog Gao. Common spatial pattern method for channel selelction in motor imagery based brain-computer interface. In *IEEE-EMBS 2005. 27th Annual International Conference of the Medicine and Biology Society*, pages 5392–5395, 2005.

[105] P.-Å. Wedin. On angles between subspaces. In B. Kågström and A. Ruhe, editors, *Matrix Pencils*, pages 263–285, New York, 1983. Springer.

[106] M. H. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigen faces. *in Proceedings of the 2000*, pages 37–40, 2000.

[107] Yokoo, B. W. Knight, and L. Sirovich. An optimization approach to signal extraction from noisy multivariate data. *NeuroImage*, 14(6):1309–1326, 24th Sep. 2001.

[108] S. A. Zekavat and C. R. Nassar. Smart antenna arrays with oscillating beam patterns: Characterization of transmit diversity using semi-elliptic-coverage geometric-based stochastic channel modeling. *IEEE Transactions on Communications*, 50(10):1–8, Oct. 2002.

[109] H. Zha. *The Singular Value Decompositions Theory, Algorithms and Applications.* PhD thesis, Pennsylvania State University, University Park, PA 16802, March 1993.

[110] P. Zhang, J. peng, and C. Domeniconi. Kernel pooled local subspaces for classification. *IEEE transaction on systems, man., and Cybernetics*, 35(3):489–502, June 2005.

[111] Q. T. Zhang. Bridging the gap between dynamic and static methods for cell planning. *IEEE Transactions on Vehicular Technology*, 50(5):1224–1230, Sep. 2001.