Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2011

ANA RITA ASSUNÇÃO TEIXEIRA

TÉCNICAS BASEADAS EM SUBESPAÇOS E APLICAÇÕES

SUBSPACE-BASED TECHNIQUES AND APPLICATIONS

Universidade de Aveiro
2011
Departamento de Electrónica, Telecomunicações e Informática

ANA RITA ASSUNÇÃO TEIXEIRA

TÉCNICAS BASEADAS EM SUBESPAÇOS E APLICAÇÕES

SUBSPACE-BASED TECHNIQUES AND APPLICATIONS

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica, realizada sob a orientação científica da Doutora Ana Maria Perfeito Tomé, Professora Associada do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Apoio financeiro da FCT no âmbito do III Quadro Comunitário de Apoio.
(SFRH/BD/28404/2006)

Dedico este trabalho ao meu marido e aos meus pais.

o júri

Presidente	Professor Doutor José Abrunheiro da Silva Cavaleiro Professor Catedrático da Universidade de Aveiro
Vogais	Professor Doutor Elmar Wolfgang Lang Professor do Institute Biophysics da Universidade de Regensburg - Alemanha
	Professor Doutor Luís Filipe Barbosa de Almeida Alexandre Professor Associado da Universidade da Beira Interior
	Professora Doutora Bernardete Martins Ribeiro Professora Associada com Agregação da Faculdade de Ciências e Tecnologia da Universidade de Coimbra
	Professor Doutor Armando José Formoso de Pinho Professor Associado com Agregação da Universidade de Aveiro
	Professora Doutora Ana Maria Perfeito Tomé Professora Associada da Universidade de Aveiro

agradecimentos

Um trabalho de investigação nunca é um trabalho solitário, é sim a confluência do esforço e dedicação de muitos. Foi fundamental o apoio e contributo de diversas pessoas, sem as quais este trabalho nunca chegaria a bom termo. Quero agradecer a todos os que contribuiram de forma activa, especialmente,

À minha orientadora Professora Associada Ana Maria Tomé, pela sua profunda sabedoria, a qual nunca parou de me surpreender e suscitar admiração. As suas orientações e conhecimentos ímpares levaram-me a aspirar a um novo nível de proficiência nesta área.

Ao Professor Elmar Lang pelas diversas conversas, apoio e orientação dados ao longo destes anos.

Ao Professor António Martins da Silva e à técnica Mónica Quintas pela marcação dos sinais EEG e pela disponibilidade.

Aos meus queridos pais por todo o amor, dedicação e confiança que sempre depositaram em mim. Pela ajuda contínua ao longo de todo o percurso académico da minha vida. Sem eles não teria sido possível este trabalho.

Ao meu querido marido, pela paciência, dedicação e ajuda preciosa.
A ele devo um agradecimento sem fim, porque sempre me encorajou e apoiou desde a fase preliminar até à conclusão desta tese. Este trabalho também é dele!

À Daisy pelos momentos felizes que sempre me proporcionou.

À minha avó pelo carinho e admiração.

Ao meu tio Fernando e às minha primas Sara e Filipa que me acompanham activamente em todos os momentos da minha vida.

Aos meus amigos, Liliana, Sara, Ana Helena, Rita Simões, Mónica e Joana Campus que sempre me ouviram e foram um apoio fundamental nas fases menos boas deste percurso.

A todos o meu muito obrigada, do fundo do coração.

palavras-chave

Técnicas de Projecção, Modelo de Subspaço, Extração de características, Eliminação de ruído, Séries Temporais, EEG, PCA, Kernel PCA, Greedy KPCA, SSA, Local SSA, Nyström, Pre-imagem

resumo

Este trabalho focou-se no estudo de técnicas de sub-espaço tendo em vista as aplicações seguintes: eliminação de ruído em séries temporais e extracção de características para problemas de classificação supervisionada. Foram estudadas as vertentes lineares e não-lineares das referidas técnicas tendo como ponto de partida os algoritmos SSA e KPCA. No trabalho apresentam-se propostas para optimizar os algoritmos, bem como uma descrição dos mesmos numa abordagem diferente daquela que é feita na literatura. Em qualquer das vertentes, linear ou não-linear, os métodos são apresentados utilizando uma formulação algébrica consistente. O modelo de subespaço é obtido calculando a decomposição em valores e vectores próprios das matrizes de kernel ou de correlação/covariância calculadas com um conjunto de dados multidimensional.

A complexidade das técnicas não lineares de subespaço é discutida, nomeadamente, o problema da pre-imagem e a decomposição em valores e vectores próprios de matrizes de dimensão elevada. Diferentes algoritmos de pré-imagem são apresentados bem como propostas alternativas para a sua optimização. A decomposição em vectores próprios da matriz de kernel baseada em aproximações *low-rank* da matriz conduz a um algoritmo mais eficiente- o *Greedy* KPCA.

Os algoritmos são aplicados a sinais artificiais de modo a estudar a influência dos vários parâmetros na sua performance. Para além disso, a exploração destas técnicas é extendida à eliminação de artefactos em séries temporais biomédicas univariáveis, nomeadamente, sinais EEG.

keywords

Projective Techniques, Subspace Model, Feature Extraction, Denoising, Time Series, EEG, PCA, Kernel PCA, Greedy KPCA, SSA, Local SSA, Nyström, Pre-image

abstract

This work focuses on the study of linear and non-linear subspace projective techniques with two intents: noise elimination and feature extraction. The conducted study is based on the SSA, and Kernel PCA algorithms.

Several approaches to optimize the algorithms are addressed along with a description of those algorithms in a distinct approach from the one made in the literature. All methods presented here follow a consistent algebraic formulation to manipulate the data. The subspace model is formed using the elements from the eigendecomposition of kernel or correlation/covariance matrices computed on multidimensional data sets.

The complexity of non-linear subspace techniques is exploited, namely the pre-image problem and the kernel matrix dimensionality. Different pre-image algorithms are presented together with alternative proposals to optimize them.

In this work some approximations to the kernel matrix based on its low rank approximation are discussed and the Greedy KPCA algorithm is introduced.

Throughout this thesis, the algorithms are applied to artificial signals in order to study the influence of the several parameters in their performance. Furthermore, the exploitation of these techniques is extended to artefact removal in univariate biomedical time series, namely, EEG signals.

Contents

List of Figures

List of Tables

Notation

Upper-case bold letters denote matrices, for example $\mathbf{K}$. Vectors are denoted by lower-cases bold letters, for example $\mathbf{x}$, and are column vectors. Subscripting is used for indexing, thus x_i, denotes the ith element of vector $\mathbf{x}$ and K_{ij} represent the element in the ith row and in the jth column of $\mathbf{K}$.

$x[k]$	-	Input signal
$\hat{x}[k]$	-	Extracted signal
$y[k]$	-	Corrected signal
$\mathbf{S}$	-	Covariance matrix
$\mathbf{X}$	-	Trajectory matrix
$\mathbf{x}_n$	-	Column n of the trajectory matrix
$\mathbf{X}_r$	-	Toeplitz matrix
$\hat{\mathbf{X}}$	-	Denoised trajectory matrix
$\mathbf{D}$	-	Eigenvalues matrix of $\mathbf{S}$
$\mathbf{U}$	-	Eigenvectors matrix of $\mathbf{S}$
$\mathbf{K}$	-	Kernel matrix
$\mathbf{K}_r$	-	Kernel matrix of subset $\mathbf{\Phi}_r$
$\mathbf{K}_s$	-	Kernel matrix of subset $\mathbf{\Phi}_s$
$\mathbf{K}_{rs}$	-	Kernel matrix between the subset $\mathbf{\Phi}_r$ and $\mathbf{\Phi}_s$
$\mathbf{Y}$	-	Matrix of Projections
$\mathbf{I}$	-	Identity matrix
$\mathbf{L}$	-	Triangular matrix
$\mathbf{Q}$	-	Matrix of the neighbors in input space
$\mathbf{Q}_c$	-	Centered Matrix of the neighbors in input space
$\mathbf{M}$	-	Mixing matrix
$\phi(\mathbf{x})$	-	Map from data to feature space
$\mathbf{\Phi}$	-	Matrix of the data $\mathbf{X}$ in feature space
$\mathbf{\Phi}_R$	-	Matrix of the data $\mathbf{X}_r$ in feature space
$\mathbf{\Phi}_S$	-	Matrix of the data $\mathbf{X}_s$ in feature space
$\tilde{\mathbf{K}}$	-	Centered Kernel matrix
$\mathbf{k}$	-	Column vector of the Kernel matrix
K	-	Number of the time series samples
M	-	Embedding dimension - Integer window length
N	-	Number of samples

T	-	Number of samples of the training dataset
R	-	Number of non-negative eigenvalues
L	-	Number of selected eigenvalues
L_{c_i}	-	Number of selected eigenvalues in the cluster i
q	-	Number of clusters
Sc	-	Shur Complement
c	-	Cluster
f_s	-	Sampling frequency
f_r	-	Minimum frequency to be extracted
σ	-	RBF parameter
σ_x^2	-	Standard deviation of $\mathbf{x}$
$L(\hat{\theta})$	-	Log likelihood function
ϵ	-	Square error
λ_i	-	ith diagonal entry of matrix $\mathbf{D}$
d^2	-	Distance in input space
d_0^2	-	Distance of the neighbors to the origin
$\tilde{d}^2$	-	Distance in feature space
p	-	Pre-image
S	-	Number of neighbors
$\mathbf{u}_k$	-	k-th eigenvector of the covariance matrix
$\mathbf{v}_k$	-	k-th eigenvector of the kernel matrix
ct_{xy}	-	Correlation coefficient
cf_{xy}	-	Coherence function
μ_1	-	Test error average

Glossary

AIC *Akaike Information Criterion.*

BIC *Bayesian Information Criterion.*

BSS *Blind Source Separation.*

dB Decibel.

ECG Electrocardiogram.

EEG Electroencephalogram.

EMG Electromiogram.

EOG Electrooculogram.

ICA *Independent Component Analysis.*

ITC *Information Theoretic Criteria.*

KPCA *Kernel Principal Component Analysis.*

KICA *Kernel Independent Component Analysis.*

KFDA *Kernel Fisher Discriminant Analysis.*

GEVD *Generalized Eigenvalue Decomposition.*

EDF *European Data Format.*

RD *Raw Data.*

NN *One Nearest Neighbor.*

RL *Linear Discriminant Function.*

MDL *Minimum Description Length.*

MSE *Mean Square Error.*

NMR *Nuclear Magnetic Resonance.*

PCA *Principal Component Analysis.*

LDA *Linear Discriminant Analysis.*

PDF *Probability Density Function.*

PSD *Power Spectral Density.*

RBF *Radial Basis Function.*

SNR *Signal to Noise Ratio.*

SSA *Singular Spectrum Analysis.*

SVD *Singular Value Decomposition.*

SVM *Support Vector Machine.*

VE Explained Variance.

DFT Discrete Fourier Transformer.

FIT Finite Impulse Response.

Chapter 1

Introduction

"Reading, after a certain age, diverts the mind too much from its creative pursuits. Any man who reads too much and uses his own brain too little falls into lazy habits of thinking."
- Albert Einstein -

Contents

1.1 Motivation

The initial motivation for this work is to study different aspects of linear and non-linear projective subspace techniques in order to determine whether the available techniques can be applied in the denoising of time series, namely EEG signals.
EEG signals are often corrupted by high amplitude artifacts, like EOG. These artifacts complicate the EEG interpretation as, for instance, a seizure onset will be difficult to detect in epileptic data analysis. Eye movements and blinking are of larger amplitude than cortical EEG so, ocular artifacts pose a significant problem to the clinicians and neurologists either by the loss of the data or by masking significants events in the data. The goal is to apply projective subspace techniques to remove artifacts without distorting the underline brain

signals of interest.
Projective subspace techniques are not available for one dimensional time series, hence time series analysis techniques often rely on the embedding of a one dimensional sensor signal in a high dimensional space of time-delayed coordinates. As embedding can be regarded as a nonlinear signal manipulation, it is to be expected that the non-linear technique based on kernel methods should be even more appropriate to denoise time series. So, it will be of interest to explore these techniques and their ability to remove dominant artifacts and/or suppress noise.
The main advantage of kernel methods when compared to linear methods is that the number of feature space components which belong to the signal subspace are not limited by the dimension of the data. Its main drawbacks are related to the mapping from feature space to input space (the pre-image problem) and to the increasing complexity related to the number of samples in the training set. In order to reduce the computational complexity, a variant of kernel methods based on a Nyström extension will be presented, whose parameters are computed using the eigendecomposition of a low-rank approximation of the kernel matrix.
The second motivation is to find better representations of a given set of data with more informative features in order to improve the performance of a classifier. The kernel methods can be used to extract a relevant dataset into the feature space. The data reduction can be done in terms of features of the dataset considered. Redundant or highly dependent features can be replaced by features with smaller correlations capturing the entire information.

1.2 Summary of Novelties

The main goal of this work is to investigate, develop and apply linear and non-linear techniques to signal analysis. The embedding operation in time delayed coordinates leads to a multidimensional signal which presents non-linear characteristics. In that context, the extension of singular spectrum analysis to a local principal component analysis in the space of time-delayed coordinates seems to provide an efficient and simple tool for denoising. The other alternative is to perform principal component analysis in the high-dimensional spaces generated by kernel methods. With the latter methods, the computational complexity and the pre-image problem also need to be considered. The main contributions of this thesis can be summarized as follows:

- The choice of the parameters in Local SSA algorithm (the dimension of the embedding, number of clusters and number of directions).
- The interpretation of SSA as a bank of filters.
- With a KPCA algorithm the following issues will be discussed:
  - KPCA adapted to deal with one-dimensional signals by embedding them into the space of their delayed coordinates.

  - The pre-image problem. Two methods proposed in literature will be formulated using a coherent algebraic notation.
  - Techniques developed to reduce the computational complexity inherent to kernel methods.
  - Different Nyström approaches (orthogonal and non-orthogonal approaches) to compute the basis vectors in greedy KPCA algorithm.
  - Strategies to split the data based on incomplete Cholesky decomposition of the kernel matrix exploited to compute the Nyström extension, as well as the stopping criterion of the Cholesky decomposition.
  - The choice of the parameters in KPCA algorithms (σ parameter, number of directions).
  - The influence of centering the data in the KPCA and greedy KPCA algorithm's models. The models description are adapted to remove the mean of the data.

- Unichannel analysis to denoise a single EEG channel suffering from high amplitude interference using linear and non-linear projective techniques.

- Performance evaluation of the algorithms, in order to remove EOG artifacts from EEG signals in the presence of different EEG activities.

- New insight into unsupervised feature extraction techniques based on kernel methods. The data projected onto kernel subspace models are new data representations, which are more suitable for classification.

Study Cases Some of the issues described before, started to be studied in the context of noise reduction in the sense that only signal and noise are identified. The noise reduction was done in frontal EEG channels to extract the high-amplitude EOG signal from the EEG. The reconstructed EEG artifact-free signal can be recovered. All EEG signals were taken from a set of signals of epileptic patients created in Hospital Geral de Santo António. The signals were visually annotated with relevant information. Another case study to be considered is feature extraction of relevant components from the dataset using projective techniques. Two groups of data were considered: one constituted by thirteen benchmarks, section A.1.3 and the other by the USPS dataset, section A.1.2. The goal was to compare the classifiers performance using kernel features.

1.3 Chapter-by-chapter Overview

This thesis is organized into seven chapters which can be summarized as follows:

- **Chapter 1**
  In chapter 1 a review of the problems to be addressed in this thesis and the goals of the work are presented. The organization of the thesis and the publications done during this work are presented at the end of the chapter.

- **Chapter 2**

  Chapter 2 is a review of the basic theory that will be used throughout this work. In particular, it will set some basic notation and recall some notions from subspace techniques theory, eigendecomposition - primal and dual PCA (section 2.1), projective subspace techniques (section 2.2), subspace measures (section 2.3) and multivariate signal analysis - embedding and diagonal averaging (section 2.4). Work applications (denoising and feature extraction) are presented and discussed in section 2.5 and in the last section some conclusions are drawn.

- **Chapter 3**

  Chapter 3 deals with the SSA algorithm as well as the Local SSA algorithm and its differences. In addition, an example is given to illustrate the advantages of the Local SSA algorithm. The subspace distance is also used to better interpret the results. In section 3.3 the parameters of the Local SSA algorithm are discussed - embedding dimension and number of directions and clusters. In section 3.4 the application of SSA using a linear invariant system approach and its interpretation as a filter bank system is discussed and an example is presented to illustrate the main characteristics. In the end some conclusions are exposed.

- **Chapter 4**

  Chapter 4 focuses on non-linear subspace models: KPCA and greedy KPCA. First a brief review of the main steps of Kernel PCA algorithm was done in section 4.1. Some of the requirements treated in this chapter are kernel matrices and non-linear projections (section 4.1), reconstruction in feature space (section 4.1) and the pre-image problem - distance and fixed-point algorithm (section 4.2). Some illustrative examples were used to evaluate the pre-image performance. In section 4.3, a low rank approximation of kernel matrix based on Nyström approach is discussed. To compute the basis vectors, the orthogonal and non-orthogonal approaches are discussed (section 4.3). Two related strategies to split the dataset are reviewed, accomplished with some datasets examples, section 4.3.2. The three remainder sections discuss the RBF parameter selection (section 4.4), the centering problem in feature space using a complete and a reduced training sets (section 4.5) and finally the conclusions (section 4.6).

- **Chapter 5**

  In Chapter 5, the methodologies presented in the last chapters are now employed in time series analysis, to denoise the EEG signals. An introduction about the dataset used, arises from two distinct ways: artificial EEG mixed with different artifacts and real EEG contaminated with artifacts. An overview about the EEG signal and the artifacts is done (section 5.2). The data collection is presented in section 5.3. The parameters of evaluation in frequency and in time domains are presented (section 5.4) and some results of the algorithms are shown and discussed along this chapter. The real EEG applications and some results with Local SSA and Greedy KPCA are discussed and compared in section 5.6. Finally, some conclusions are drawn in section 5.7.

- **Chapter 6**
  In Chapter 6, the feature extraction application based in the linear and non-linear projective techniques described in chapters 3 and 4 is presented. Section 6.1 resumes the feature extraction and the classification methods used in this work. The numerical simulations compare the performance of classifiers using kernel features, principal component features and a direct classification of the raw data using two classifiers: the nearest neighbor (NN) and the linear discriminant function (RL). Furthermore, to evaluate the impact of the projective techniques, a comparative study, with the best results published, is presented and discussed (section 6.2). Greedy KPCA is used with a large dataset (USPS dataset of handwritten digits), something which is often used as a benchmark test dataset (section 6.3). Conclusions are presented in the last section.

- **Chapter 7**
  Chapter 7 presents the general conclusions of the thesis and proposes possible improvements and directions on future research work.

In appendix A, the datasets used in this work are described in detail in section A.1. The principal characteristics of the EEG and the EOG signals used in chapter 5 and the mixing model are present in section A.2. In the last section the Cholesky decomposition is described.

1.4 List of Publications

The main contributions of this work have been published.

1.4.1 Main Contributions

1. Ana Rita Teixeira, Ana Maria Tomé, Elmar W. Lang. Unsupervised feature extraction via kernel subspace techniques. Neurocomputing, vol. 74, no. 5, pp. 820-830, February 2011.

2. Ana Rita Teixeira, Ana Maria Tomé, M. Boehm, Carlos G. Puntonet, Elmar W. Lang. How to apply nonlinear subspace techniques to univariate biomedical time series. IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 8, pp. 2433-2443, August 2009.

3. Ana Rita Teixeira, Ana Maria Tomé, K. Stadlthanner, E. W. Lang. KPCA denoising and the pre-image problem revisited. Digital Signal Processing, vol. 18, no. 4, pp. 568-580, July 2008 (IF 2008: 1.486).

4. Ana Rita Teixeira, Ana Maria Tomé, E. W. Lang, P. Gruber, A. Martins da Silva. Automatic removal of high-amplitude artefacts from single-channel electroencephalograms. Computer Methods and Programs in Biomedicine, vol. 83, no. 2, pp. 125-138, August 2006.

5. A.R. Teixeira, A.M. Tomé, E. W. Lang. Feature Extraction Using Linear and Non-linear Subspace Techniques. Lecture Notes in Computer Science. Artificial Neural Networks, ICANN 2009, pp. 115-124, Volume 5769, 19th International Conference, Limassol, Cyprus.

6. A.R. Teixeira, A.M. Tomé, E. W. Lang and A. Martins da Silva. Subspace Techniques to Remove Artifacts From EEG: a quantitative analysis, Conference of the IEEE Engineering in Medicine and Biology Society, August 20-24, 2008 Vancouver, Canada.

7. A. R.Teixeira, A. Maria Tomé, E. W. Lang. Feature extraction using low-rank approximations of the kernel matrix. International Conference on Image Analysis and Recognition,ICIAR2008, June 25-27, 2008 Póvoa do Varzim Portugal.

8. A. R.Teixeira, A. Maria Tomé, E. W. Lang. Greedy KPCA in Biomedical Signal Processing, accepted in International Conference on Artificial Neural Networks, ICANN'2007, September 10-13, 2007 Porto, Portugal, 2007.

9. A. M. Tomé, A. R. Teixeira, E. W. Lang, and A. Martins da Silva. Greedy KPCA Applied to Single- Channel EEG Recordings, In 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2007), Lyon, 2007.

10. A. R. Teixeira, A. M. Tomé, and E. W. Lang. Exploiting Low-Rank Approximations of Kernel Matrices in Denoising Applications, In IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2007), Thessaloniki, Greece, 2007.

11. A. R. Teixeira, N. Alves, A. M. Tomé, M. Bohm, E. W. Lang, and C. G. Puntonet. Single-Channel electroencephalogram analysis using non-linear subspace techniques, In IEEE International Symposium on Intelligent Signal Processing (WISP 2007), Madrid, Spain, 2007.

12. A. R. Teixeira, A. M. Tomé, E.W.Lang, P. Gruber, and A. M. da Silva. Extraction and separation of high-amplitude artifacts in electroencephalograms from epileptic patients, Fourth IASTED International Conference on Biomedical Engineering- BIOMED2006 (C. Ruggiero, ed.), pp. 270-275, IASTED, Innsbruck, Austria, 2006.

13. A. R. Teixeira, A. M. Tomé, E. W. Lang, and K. Stadlthanner. Nonlinear Projective Techniques To Extract Artifacts in Biomedical Signals. In EUSIPCO2006, Florence, Italy, 2006.

14. A. R. Teixeira, A. M. Tomé, E. W. Lang, R. Schachtner, and K. Stadlthanner. On the Use of KPCA to Extract Artifacts in One-Dimensional Biomedical Signals. In Séan McLoone, Jan Larsen, Marc Van Hulle, Alan Rogers, and Scott C. Douglas (Editors), Machine Learning for Signal Processing, MLSP 2006, pp. 385-390, Dublin, 2006.

15. A. M. Silva, A. M. Tomé, A. R. Teixeira. Analyzing Single Channel EEG Recordings to Extract EOG Artifacts. Third International Conference on Neural Networks (ICNN2006), Barcelona, October 27-29, 2006.

1.4.2 Other Contributions

1. A. M. Tomé, A. R. Teixeira, N. Figueiredo, I. M. Santos, P. Georgieva, E. W. Lang. SSA of biomedical signals: a linear invariant systems approach. Statistics and Its Interface, vol. 3, no. 3, pp. 345-355, 2010.

2. A. M. Tomé, A. R. Teixeira, E. W. Lang. Subspace Techniques and Biomedical Time Series Analysis. Recent Advances in Biomedical Signal Processing, in print, 2010.

3. A. M. Tomé, A. R. Teixeira, N. Figueiredo, P. Georgieva, I. M. Santos and E. Lang. Clustering Evoked Potential Signals Using Subspace Methods. In 32th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2010), Buenos Aires, Argentina, 2010.

4. Nuno Figueiredo, Filipe Silva, Pétia Georgieva, Ana Tomé and Ana Teixeira. Subspace Techniques and Biomedical Time Series Analysis. RECPAD 2010. UTAD University, Vila Real, October 29th, 2010.

5. Ana Teixeira, Nuno Figueiredo, A. M. Tomé and P. Georgieva and Isabel Santos. Enhancement of Evoked Potential Response Signal Using Singular Spectrum Analysis. RECPAD 2009. IEETA / University of Aveiro, Portugal, October 23rd, 2009.

6. A. R.Teixeira, A. Maria Tomé, E. W. Lang. Subspace based methods and artefact elimination. Poster presentation in NeuroMath Workshop Leuven, Katholieke Universiteit Leuven, Belgium, 12-13 March 2009.

7. Ana Rita Assunção Teixeira. Extract Artefacts in EEG signals. Poster presentation in Summer School, Neurotrain, Ofir Portugal, 21-27 June 2007.

Chapter 2

Subspace Techniques

"In the middle of difficulty lies opportunity."
- Albert Einstein -

Contents

Subspace techniques have been used frequently in digital signal processing in connection with, e.g., spectrum estimation [1], system identification [2] and digital speech processing [3, 4]. Subspace methods not only provide a new insight into such problems, but they also offer a good trade off between achieved performance, computational complexity and memory usage. They can be considered low-cost alternatives to tackle larger problems that are too expensive

for methods that work in the entire space. Many studies have shown that the estimation and detection tasks in many signal processing applications can be significantly improved by using subspace-based methodology [5]. The fundamental idea of the subspace methods is to find proper subspaces for particular classes using covariance-correlation analysis. In this work, subspace methods were used in linear and non-linear eigendecomposition problems, in input and feature space, respectively. In literature, different linear and nonlinear subspace techniques were used. Examples of linear subspace techniques include principal component analysis (PCA) and linear discriminant analysis (LDA) [6, 7], bayesian algorithm using probabilistic subspace measures [8, 9] and independent component analysis (ICA) [10]. Examples of nonlinear subspace techniques include linear methods using the kernel trick [11, 12].

In this chapter, the subspace techniques and their principles are introduced focusing in unidimensional applications for denoising and feature extraction. At first, some signal processing tools that will be used later on, like singular values decomposition, principal component analysis and subspace distance are presented. The transformation of unidimensional time series into multidimensional will be exposed and two steps are described, embedding and diagonal averaging. In the end, the work applications of subspace techniques done in this thesis will be presented.

2.1 Eigendecomposition versus Basis Vectors

Singular value decomposition (SVD) and principal component analysis (PCA) are the most common multivariate data analysis tools in signal processing. These techniques were introduced by [13] and [14, 15, 16] respectively. SVD and PCA are widely used in multivariate statistical analysis for data reduction [17]. These methods have been successfully employed in noise reduction [18], data processing and compression [17], feature extraction [17], molecular dynamics [19], signal-to-noise enhancement [20] and in waveform morphologies classification of biological signals, like ECG [21] and EEG signals [22]. SVD/PCA serves as a powerful intermediate step when addressing problems related to dimensionality reduction and pattern recognition.
In the following subsections the primal and dual PCA decompositions will be exposed as well as the SVD decomposition.

2.1.1 Primal PCA

PCA is mathematically defined as an orthogonal linear transformation. Thus, in PCA a data vector is represented in an orthogonal basis system so that the projected data has maximal variance [23]. PCA can be performed by eigenvalue decomposition of the data covariance matrix. The orthogonal transformation is obtained by diagonalizing the correlation matrix of the dataset,

$$\begin{aligned} \mathbf{S} &= \mathbf{X}\mathbf{X}^T \\ &= \mathbf{U}\mathbf{D}\mathbf{U}^T \end{aligned} \tag{2.1}$$

where $\mathbf{X}$ is matrix $M \times N$ with $N > M$, $\mathbf{U}$ are the eigenvectors $M \times M$ and $\mathbf{D}$ is the diagonal eigenvalue matrix, $M \times M$ with ordered eigenvalues ($\lambda_1 > \lambda_2 > ... > \lambda_L > ... > \lambda_M$). The eigenvectors are orthogonal, i.e, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, and it is possible to identify the components which keep the largest variance of the raw data. It is assumed that the dataset $\mathbf{X}$ is centered.

2.1.2 SVD

A different approach to obtain the same principal components is through singular value decomposition (SVD). Singular value decomposition is a factorization technique for rectangular matrices widely used in signal processing and pattern recognition. A non-square data matrix $\mathbf{X}$ of size $M \times N$ with $N > M$ can be factorized into three matrices $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ using singular value decomposition as shown in eqn. 2.2.

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{2.2}$$

where $\mathbf{U}$ is a $M \times M$ matrix and represents the eigenvectors of the correlation matrix (out products) $S = \mathbf{X}\mathbf{X}^T$; $\mathbf{\Sigma} = \begin{bmatrix}\mathbf{D}^{1/2} & \mathbf{0}_{N-M}\end{bmatrix}$ is a $M \times N$ diagonal matrix with ordered singular values ($\sqrt{\lambda_1} > \sqrt{\lambda_2} > ... > \sqrt{\lambda_L} > ... > \sqrt{\lambda_M}$) and $\mathbf{V}$ is a $N \times N$ matrix and represents the eigenvectors of the matrix of inner products $\mathbf{X}^T\mathbf{X}$.
Compared to PCA, SVD is more embracing, because SVD simultaneously provides the principal components in both row and column spaces.

2.1.3 Dual PCA

In this section, it is assumed that the dimensionality M of the $M \times N$ matrix of data $\mathbf{X}$ is large (i.e., $M >> N$). The singular value decomposition described above allows the formulation of the principal components algorithm entirely in terms of dot products between data points. This limits the direct dependence on the original dimensionality M. This fact will become important in chapter 4. Consider the eigendecomposition of the matrices of inner, eqn. 2.3 and outer products, eqn. 2.4,

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}^T\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{V}^T \tag{2.3}$$

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}^T\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{U}^T \tag{2.4}$$

By the last two equations it is possible to conclude that the matrices of outer and inner products have the same nonzero eigenvalues $N = min(M, N)$. Assuming that $\mathbf{D}$ is a $N \times N$ matrix it is possible to find the following relation:

$$\begin{aligned} \mathbf{U}\mathbf{D}^{1/2} &= \mathbf{X}\mathbf{V} \\ \mathbf{U} &= \mathbf{X}\mathbf{V}\mathbf{D}^{-1/2} \end{aligned} \tag{2.5}$$

The last relation shows that each eigenvector $\mathbf{u}_j$, columns of $\mathbf{U}$, can be represented as a linear combination of the data vectors $\mathbf{X}$ - Dual PCA

$$\mathbf{u}_j = \mathbf{X}\mathbf{v}_j\lambda_j^{-1/2} \tag{2.6}$$

where $j = 1, ..., N$. Note that in the space of dimension M it is possible to find N eigenvectors.

2.2 Projective Subspace Techniques

Projective techniques are used to generate an alternative representation of the data that can be more easily interpreted. The projective models are described by a matrix (or a couple of matrices) and generally comprise three steps: the projection of the data, the selection of the relevant components and the reconstruction.
The goal of projective subspace techniques is to describe the data with reduced dimensionality by extracting meaningful components while still retaining the structure of the raw data. Only then the projections on the directions corresponding to the most significant eigenvalues L of the kernel or covariance matrices need to be computed. The L columns of the projecting matrix $\mathbf{U}$, which represent basis vectors in an M - dimensional space, will transform the input data vector $\mathbf{x}$ by

$$\mathbf{y} = \mathbf{U}^T\mathbf{x} \tag{2.7}$$

where $L < M$ and $\mathbf{y} = (y_1, ..., y_L)$, constitutes a new representation of the data [24]. Different techniques to compute $\mathbf{U}$ lead to different subspace methods. The most widely used techniques to compute $\mathbf{U}$ are: principal component analysis (PCA), blind source separation (BSS), kernel methods (KPCA) and independent component analysis (ICA). In the first case, the projection matrix has orthogonal columns, i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, and it is possible to identify the components which keep the largest variance of the raw data. The goal of ICA is a decomposition into statistically independent components and the projecting matrix is usually a non-orthogonal matrix. BSS methods also achieve a non-orthogonal matrix but often only use second-order statistics to estimate the components of the data. However, after identifying the relevant (or irrelevant) components in $\mathbf{y}$, all methods are used in a similar way to obtain reconstructed data without the influence of undesired components. The reconstruction is defined as

$$\hat{\mathbf{x}} = (\mathbf{U}^T)^\dagger\mathbf{y} \tag{2.8}$$

where $\dagger$ denotes the pseudo-inverse. In PCA $(\mathbf{U}^T)^\dagger = \mathbf{U}$. It should be noted that only L directions of the dataset contribute to the reconstruction of $\hat{\mathbf{x}}$.

2.3 Subspace Measures

The subspace measures are used to analyze the similarity between two subspaces. Recently, there has been a growing interest in the design and analysis of similarity/distance measures over subspaces. In the literature, different subspace measures were studied like geodesic

distance [25] as a measure based on principal angles [26, 27]; chordal distance [26]; Hausdorff distance [28, 29] and subspace distance [30]. All these subspace methods base themselves on choosing a subset of eigenvectors which span the multidimensional space.

2.3.1 Principal Angles

Let $\mathscr{U}_A$ and $\mathscr{U}_B$ be subspaces in $\Re^n$. Supposing that a subspace $\mathscr{U}_A$ with p eigenvectors $\mathbf{A}$ is computed for a dataset and another subspace $\mathscr{U}_B$ with q eigenvectors $\mathbf{B}$ is computed for another dataset, where $q \geq p$.
The principal or canonical angles

$$0 \leq \theta_1 \leq \theta_2 \leq \ldots \leq \theta_p \leq \frac{\pi}{2} \tag{2.9}$$

between the two subspaces $\mathscr{U}_A$ and $\mathscr{U}_B$ are uniquely recursively defined as the minimal angles between any two vectors of the subspaces $k = 1, 2, ..., p$:

$$\begin{aligned} cos(\theta_k) &= \underbrace{max}_{\mathbf{a}\in A}\underbrace{max}_{\mathbf{b}\in B}\mathbf{a}^T\mathbf{b} \\ &= \mathbf{a}_k^T\mathbf{b}_k \end{aligned} \tag{2.10}$$

subject to

$$\begin{aligned} \|\mathbf{a}\| &= \|\mathbf{b}\| = 1 \\ \mathbf{a}^T\mathbf{a}_i &= 0 \qquad i = 1, 2, ..., k-1 \\ \mathbf{b}^T\mathbf{b}_i &= 0 \qquad i = 1, 2, ..., k-1 \end{aligned}$$

where p is the minimum of the dimensions of $\mathscr{U}_A$ and $\mathscr{U}_B$.
The vectors $\mathbf{a}_1$, $\mathbf{a}_2$,..., $\mathbf{a}_q$ and $\mathbf{b}_1$, $\mathbf{b}_2$,..., $\mathbf{b}_p$ are called principal vectors between the subspaces $\mathscr{U}_A$ and $\mathscr{U}_B$. Note that $\mathbf{a}_i$ and $\mathbf{b}_i$ represent the ith pair of principal vectors. Intuitively, the first pair of principal vectors corresponds to the most similar modes of variation of two linear subspaces. To obtain the second angle θ_2, the subspaces that are orthogonal are searched, respectively for $\mathbf{b}_1$ and $\mathbf{a}_1$. Continuing in this manner, always searching in subspaces orthogonal to principal vectors that have already been found, the complete set of principal angles and principal vectors is obtained.
The singular value decomposition is used to compute the principal angles by [31]. If the columns of $\mathbf{A}$ and the columns of $\mathbf{B}$ are orthogonal bases of two spaces, the canonical angles can be computed using the SVD of $\mathbf{A}^T\mathbf{B}$ as follows:

$$cos(\theta_k) = \sigma_k(\mathbf{A}^T\mathbf{B}), \quad k = 1, 2, ..., p \tag{2.11}$$

where $\sigma_k(\mathbf{O})$ denotes the singular value of the matrix $\mathbf{O}$. The associated principal vectors of the subspaces are also obtained using the eigenvectors of the singular values decomposition of the matrices of the orthogonal bases [31].

2.3.2 Similarity and Distance Measures

Given the matrices $\mathbf{A}$ and $\mathbf{B}$ which are orthogonal bases for subspaces $\mathscr{U}_A$ and $\mathscr{U}_B$ respectively and considering the projection matrices

$$\mathbf{P} = \mathbf{A}\mathbf{A}^T \qquad \mathbf{Q} = \mathbf{B}\mathbf{B}^T \tag{2.12}$$

It is possible to define their corresponding subspaces uniquely and the distance can be computed as

$$d^2(\mathscr{U}_A, \mathscr{U}_B) = \left\|(\mathbf{A}\mathbf{A}^T - \mathbf{B}\mathbf{B}^T)\right\|_F^2 \tag{2.13}$$

Manipulating the last equation, the distance can be formulated as

$$\begin{aligned} d^2(\mathscr{U}_A, \mathscr{U}_B) &= trace((\mathbf{P}-\mathbf{Q})^T(\mathbf{P}-\mathbf{Q})) \\ &= trace(\mathbf{P}) + trace(\mathbf{Q}) - 2trace(\mathbf{P}\mathbf{Q}) \\ &= p + q - 2\sum_{i=1}^{p}\sigma_i^2(\mathbf{A}^T\mathbf{B}) \end{aligned} \tag{2.14}$$

In this work the distance measure used to compare two subspaces was the distance proposed by [29, 30] and defined as:

$$\begin{aligned} d(\mathscr{U}_A, \mathscr{U}_B) &= \sqrt{max(p,q) - \sum_{i=1}^{p}\sum_{j=1}^{q}(\mathbf{a}_i^T\mathbf{b}_j)^2} \\ &= \sqrt{max(p,q) - tr(\mathbf{A}^T\mathbf{B}\mathbf{B}^T\mathbf{A})} \\ &= \sqrt{max(p,q) - \|\mathbf{A}^T\mathbf{B}\|_F^2} \end{aligned} \tag{2.15}$$

where $\|\|_F$ represents the Frobenius norm. The exposed distance has several properties. First, it is invariant to the choice of the orthonormal basis for the subspaces $\mathscr{U}_A$ and $\mathscr{U}_B$. Furthermore, it is symmetric and non negative, in particular $d(\mathscr{U}_A, \mathscr{U}_B) = 0$ if and only if $\mathscr{U}_A \equiv \mathscr{U}_B$. The upper bound for the subspace distance is given by $d(\mathscr{U}_A, \mathscr{U}_B) \leq \sqrt{max(p,q)}$ and corresponds to the orthogonality condition $\mathscr{U}_A \perp \mathscr{U}_A$. Finally, as proved in [32], the subspace distance satisfies the triangle inequality.
The criterion used in this work to define the similarity between two subspaces is

$$\frac{d(\mathscr{U}_A, \mathscr{U}_B)}{\sqrt{max(p,q)}} \leq \frac{1}{2} \tag{2.16}$$

The subspace distance was introduced as a measure between two linear subspaces. A crucial observation of eqn. 2.15 is that the subspace distance can be expressed in terms of inner products, therefore it is possible to generalize this measure to the nonlinear case.

2.4 Multivariate Signal Analysis

In this section some signal processing operations for multivariable signals will be discussed. The projective subspace techniques discussed so far are clearly not available for one-dimensional time series to suppress noise contributions, but many signal processing applications rely on one-dimensional signals, like biomedical signals.
In multi-sensor signal processing the data vector $\mathbf{x}[n] = (x_1 x_2 ... x_M)^T$ is naturally formed with samples from different sensors. The projective techniques can be applied directly by forming a data matrix with N samples where each column represents a time sample of the multichannel recording,

$$\mathbf{X} = \begin{bmatrix} x_1[0] & x_1[1] & \dots & x_1[N-1] \\ x_2[0] & x_2[1] & \dots & x_2[N-1] \\ x_3[0] & x_3[1] & \dots & x_3[N-1] \\ \vdots & \vdots & & \vdots \\ x_M[0] & x_M[1] & \dots & x_M[N-1] \end{bmatrix} \tag{2.17}$$

However, projective subspace techniques can also be applied to single sensor signals (unichannel analysis) by forming vectors with windows of the signal. In the following sections, the unichannel analysis will be formalized with two steps: embedding and diagonal averaging.

2.4.1 Embedding

Projective subspace techniques can not be directly applied in one dimensional time series, therefore time series analysis techniques often rely on the embedding of a one dimensional sensor signal in a high dimensional space of time delayed coordinates [33, 3, 34]. Note that space of time delayed coordinates is also called embedding space and phase space. The embedding strategy is used in many signal processing applications to obtain a multidimensional signal. Embedding is a standard procedure for time series analysis. This method is used in chaotic time series prediction to capture the full dynamical system [35, 36], in singular spectrum analysis [33] or in statistics for signals with finite decaying memory [37]. The embedding transformation can be regarded as a mapping that transforms a one-dimensional time series $x = (x[0], x[1], ..., x[K-1])$ into a multidimensional sequence of lagged vectors. Let M be a integer window length with $M < K$. The embedding procedure forms $N = K - M + 1$ lagged vectors $\mathbf{x}_n$

$$\mathbf{x}_n = [x[n-1+M-1], \dots, x[n-1]]^T,\ n = 1, \dots, N \tag{2.18}$$

The lagged vectors $\mathbf{x}_n$ lie in a space of dimension M and constitute the columns of the $M \times N$ trajectory matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]$,

$$\mathbf{X} = \begin{bmatrix} x[M-1] & x[M] & \dots & x[K-1] \\ x[M-2] & x[M-1] & \dots & x[K-2] \\ x[M-3] & x[M-2] & \dots & x[K-3] \\ \vdots & \vdots & & \vdots \\ x[1] & x[2] & \dots & x[K-M+1] \\ x[0] & x[1] & \dots & x[K-M] \end{bmatrix} \tag{2.19}$$

Note that the trajectory matrix $\mathbf{X}$ is a Toeplitz matrix, a matrix that has identical entries along its diagonals. There are other alternatives to form the data matrix via embedding the signal into M-dimensional space that lead to an Hankel structured matrix which then has identical elements along the anti-diagonals [33, 38]. The embedding step requires the assignment of the window length, M parameter. The size of the embedding window M (number of rows of the trajectory matrix) should be large enough to capture the global behavior of the dataset. If no further knowledge of the data is available, M should be chosen approximately as half of the segment length (K) [33]. To extract periodic signal components, M should be close to their periodicity [33]. A more general strategy is used in [39] where a lower bound is suggested according to the frequency resolution contained in every column of the trajectory matrix.

A similar criterium was used in [40] to find the embedding dimension of an algorithm based on independent component analysis. Another method to determine M is to use the point where mutual information between the first and the last column of the trajectory matrix reaches the first minimum [41]. However, the last method carries a computational penalty.

Computing the covariance matrix, variations of the window length only stretch or compress the spectrum of the eigenvalues, leaving the relative magnitudes of the eigenvalues unchanged [42]. The further processing of the data matrix $\mathbf{X}$ can be performed by different algorithms considering each column a point in a dimensional space M. After the embedding step, the signal $\mathbf{x}_n$ is projected onto the directions (eigenvectors) related to the largest eigenvalues of the covariance matrix or the kernel matrix.

2.4.2 Diagonal Averaging

After applying the algorithms to each column of the trajectory matrix ($\mathbf{X}$), a new matrix of the data is obtained ($\hat{\mathbf{X}}$). New points then form the columns of $\hat{\mathbf{X}}$, the "new trajectory matrix". However, in general this matrix does not possess the characteristic Toeplitz structure, i.e. the elements along each descending diagonal of $\hat{\mathbf{X}}$ will not be identical, as it was the case of the original trajectory matrix $\mathbf{X}$. This can be fixed, however, by replacing the entries in each diagonal by their average, obtaining a Toeplitz matrix $\mathbf{X}_r$. This procedure assures that the Frobenius norm of the difference ($\mathbf{X}_r - \hat{\mathbf{X}}$) attains its minimum value among all possible solutions to get a matrix with all of its diagonals equal [33].

2.5 Work Applications

Subspace techniques can be used to different ends, as described before. In this work, subspace techniques were applied to denoise and to extract features on different signals. In subspace methods, denoising or classification is achieved by projecting the data onto basis vectors.
The algorithms were implemented in MATLAB using the toolbox provided in [43], where basic pattern recognition tools and kernel methods can be found. This toolbox was used as a support basis for this work, however several functions were included to implement the algorithms and methods proposed in this thesis.

2.5.1 Denoising

In many biomedical signal processing applications, a sensor signal is contaminated with noise, as well as signal artifacts of substantial amplitude. The artifacts can sometimes be the most prominent signal component registered. Noise signals are frequently modeled as being additive, normally distributed and uncorrelated to the signals of interest. Signal to noise ratios (SNR) are often quite low. Therefore, to recover the signals of interest, the task consists on removing both artifactual components as well as the superimposed noise contributions [39].
The objective of noise reduction techniques is to improve noisy signals. Projective subspace techniques can be used favorably to get rid of most of the noise contributions to multidimensional signals [44]. The goal of subspace methods is to project the noisy signal onto two subspaces: the signal plus noise subspace, or simply signal subspace (since the signal dominates this subspace), and the noise subspace. Hence an estimate of the clean multidimensional signal can be made by removing or nulling the components of the signal in the noise subspace, retaining only the components in the signal subspace. The decomposition of the space into two subspaces can be done using either singular values decomposition (SVD) or principal component analysis (PCA). Both strategies are achieved by estimating those directions, corresponding to the L largest eigenvalues or singular values, which can be associated to the eigenvectors spanning the signal subspace. The remaining orthogonal directions can then be associated with the noise subspace. Reconstructing the multidimensional signal using only those L dominant components can result in a substantial noise reduction of the recorded signals.
One of the applications of this work is the use of algorithms in EEG signals. The availability of digital EEG recordings allowed the study of different procedures trying to remove the artifact from the recorded brain signals.
The projective subspace techniques referred earlier can be used to separate the artifacts from the "pure" signals.
The input of the algorithms is the one-dimensional EEG signal contaminated with artifact $x[k]$, $k = 1, ..., K$. The output of the algorithms is the one-dimensional signal, $\hat{x}[k]$ obtained by reverting the embedding. If the $\hat{x}[k]$ corresponds to the high amplitude artifact, then the correct EEG signal, $y[k]$, is computed as

$$y[k] = x[k] - \hat{x}[k] \tag{2.20}$$

An example of the procedure described before by eqn. 2.20 is seen in figure 2.1. Signals 1, 2

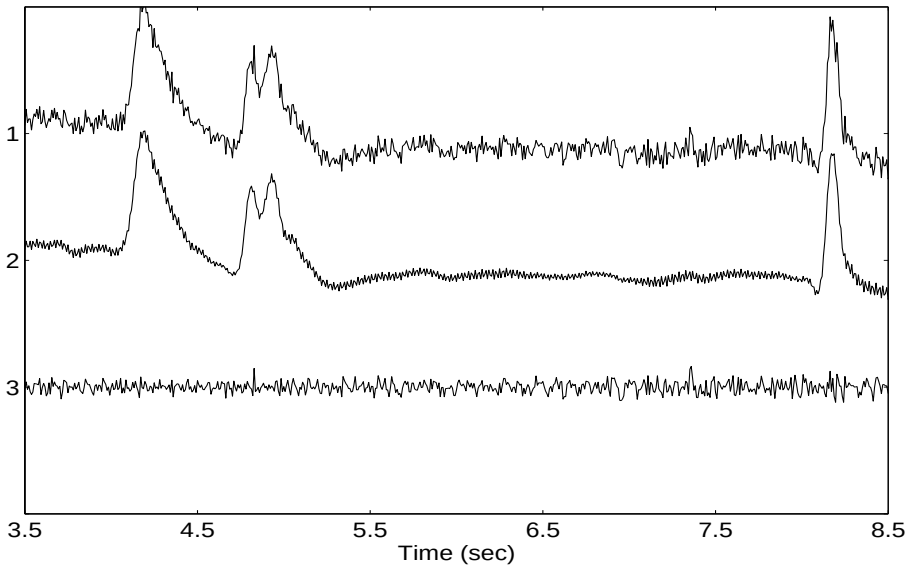


Figure 2.1: 1-Original EEG ($x[k]$), 2-Extracted EEG ($\hat{x}[k]$), 3-Corrected EEG ($y[k]$).

and 3 represented in figure 2.1 correspond to the original EEG, the extracted EEG artifact and to the corrected EEG, respectively.

2.5.2 Feature Extraction

Feature extraction is an essential pre-processing step to pattern recognition and machine learning problems. During the feature extraction process, the dimensionality of data is reduced. This is necessary most of the times, due to the technical limits in memory and computation time. A good feature extraction scheme should maintain and enhance the features of the input data that make distinct pattern classes separate from each other. In this work, feature extraction was achieved using subspace techniques. Features are the result of the projections of the data onto the coordinate's space, created by linear and non-linear functions, eqn. 2.7. The feature extraction was performed on the basis vectors computed in input and feature spaces. The aim is to describe the data by extracting meaningful components while maintaining the structure of the raw data. The effectiveness of the features extraction strategies were evaluated resorting to different classifiers. The study considered the influence of noise in the feature extraction process as well as the performance of the classifiers using different datasets.

2.6 Conclusion

In this chapter subspace models concepts were introduced and casted in a concise presentation by the use of the dual form for the linear models. Two ways to compute the subspace models were presented using L directions. So, the subspace model is described by the matrix $\mathbf{U}$ whose L columns are the basis vectors of the new representation.

It is noteworthy that there are two ways to calculate the subspace models, the dual and the primal. The dual form is used when the data dimension (M) is larger than the number of examples dimension (N). It will also be seen that it is the starting point of the kernel models. The main applications of the subspace techniques used in this work, denoising and feature extraction, were described. The subspace models will be experimentally evaluated using different datasets, as will be seen along this work. Some notation, useful in the next chapters, was introduced.

Note that subspace models rely on a multidimensional representation of the data. However, projective subspace techniques can be applied to single sensor signals by forming vectors with sliding windows of the signal. In this chapter the embedding step that transforms univariate signals into multidimensional signal vectors is introduced as well as the diagonal averaging step which is the reverse of embedding.

Bibliography

[1] Z. Leonowicz, T. Lobos, and J. Rezmer, "Advanced spectrum estimation methods for signal analysis in power electronics," *IEEE Transaction on Industrial Electronics*, vol. 50, no. 3, pp. 514–519, 2003.

[2] A. Chiuso and G. Picci, "Consistency analysis of some closed-loop subspace identification methods," *Automatica*, vol. 41, pp. 377–391, 2005.

[3] Y. Ephraim and H. L. Van Trees, "A Signal Subspace Approach for Speech Enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, 1995.

[4] J. Huang and Y. Zhao, "A DCT-Based fast signal subspace technique for robust speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 747–751, 2000.

[5] K. Berberidis, B. Champgne, G. V. Moustakides, H. V. poor, and P. Stoica, "Advances in subspace-based techniques for signal processing and communications," *Eurasip Journal on Advances in Signal Processing*, 2007.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[7] X. Wang and X. Tang, "A Unified Framework for Subspace Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1222–1228, 2004.

[8] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian Face Recognition," *Pattern Recognition*, vol. 33, 2000.

[9] B. Moghaddam, "Principal Manifolds and Probabilistic Subspaces for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 780–788, 2002.

[10] P. Comon, "Independent Component Analysis: A new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[11] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[12] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller, "Fisher Discriminant Analysis with Kernels," in *Proceedings of the 1999 IEEE Signal Processing Society Workshop*, (Madison, WI, USA), pp. 41–48, 1999.

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

[14] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[15] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *J. Educ. Psychol*, vol. 24, pp. 417–441, 1933.

[16] H. Hotelling, "Simplified Calculation of Principal Components," *Psychometrika*, pp. 27–35, 1936.

[17] M. E. Wall, A. Rechtsteiner, and L. Rocha, "Singular Value Decomposition and Principal Component Analysis," in *In A Practical Approach to Microarray Data Analysis*, pp. 91–109, 2003.

[18] P. K. Sadasivan and D. N. Dutt, "SVD based technique for noise reduction in electroencephalographic signals," *Signal Processing*, vol. 55, pp. 179–189, 1996.

[19] T. D. Romo, J. B. Clarage, D. C. Sorensen, and G. N. J. Phillips, "Automatic identification of discrete substates in proteins: singular value decomposition analysis of time-averaged crystallographic refinements.," *Proteins: Structure, Function, and Bioinformatics*, vol. 22, no. 4, pp. 311–321, 1995.

[20] M. Bekara and M. Bann, "Local Singular Value Decomposition for signal enhancement of seismic data," *Geographysics*, vol. 72, no. 2, pp. 59–65, 2007.

[21] F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. M. Roig, "Principal component analysis in ECG signal processing," *Eurasip Journal on Advances in Signal Processing*, vol. 2007, p. 21, 2006.

[22] F. Foresta, F. C. Maorabito, B. Azzerboni, and M. Ipsale, "PCA and ICA for extraction of EEG dominant components in cerebral death assessment," in *Proceedings of International Joint Conference on Neural Networks*.

[23] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks, Theory and Applications*. Wiley, 1996.

[24] A. R. Teixeira, A. M. Tomé, and E. W. Lang, "Exploiting Low-Rank Approximations of Kernel Matrices in Denoising Applications," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2007)*, (Thessaloniki, Greece), 2007.

[25] Y. Wong, "Differential geometry of grassmann manifolds," in *Proceedings of the National Academy of Science*, vol. 57, pp. 589–594, 1967.

[26] T. Chin and D. Suter, "A New Distance Criterion for Face Recognition using Image Sets," in *Lecture Notes in Computer Science*, vol. 3851, pp. 549–558, 2006.

[27] A. Galántai and Hegedüs, "Jordan's principal angles in complex vector spaces," *Numerical Linear Algebra with Applications*, vol. 13, pp. 589–598, 2006.

[28] P. Selwyn, "The Hausdorff Distance Measure for Feature Selection in Learning Application," in *Proceedings of the 32nd Hawaii International Conference on System Sciences* (IEEE, ed.), (Hawaii), 1999.

[29] L. Wang, X. Wang, and J. Feng, "Intrapersonal subspace analysis with application to adaptive Bayesian algorithm for face recognition," *Pattern Recognition*, vol. 38, no. 4, pp. 617–621, 2005.

[30] L. Wang, X. Wang, and J. Feng, "Subspace distance analysis with application to adaptive Bayesian algorithm for face recognition," *Patter Recognition*, vol. 39, no. 3, pp. 456–464, 2006.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computation*, vol. Baltimore. The John Hopkins University Press, 1989.

[32] X. Sun, L. Wang, and J. Feng, "Further results on the subspace distance," *Pattern Recognition*, vol. 40, pp. 328–329, 2007.

[33] N. Golyandina, *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman & Hall/CRC, 2001.

[34] C. H. You, S. N. Koh, and S. Rahardja, "Signal subspace speech enhancement for audible noise reduction," in *ICASPP 2005*, vol. I, (Philadelphia, USA), pp. 145–148, IEEE, 2005.

[35] J. Zhang, H. Shu-Hung Chung, and W.-L. Lo, "Chaotic Time Series Prediction using a Neuro-Fuzzy System with Time-Delay Coordinates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 956 – 964, 2008.

[36] A. K. Alparslan, M. Sayar, and A. R. Atilgan, "State-space prediction model for chaotic time series," *Physical Review E*, vol. 58, no. 2, pp. 2640–2643, 1998.

[37] I. W. Sandberg and L. Xu, "Uniform approximation of multidimensional myoptic maps," *IEEE Transactions on Circuits and Systems I*, vol. 44, pp. 477–485, 1997.

[38] P. C. Hansen and S. H. Jensen, "Subspace-based noise reduction for speech signals via diagonal and triangular matrix decompositions: Survey and analysis," *Eurasip Journal on Advances in Signal Processing*, vol. 2007, 2007.

[39] A. R. Teixeira, A. M. Tomé, E. Lang, P. Gruber, and A. M. Silva, "Automatic removal of high-amplitude artifacts from single-channnel electroencephalograms," *Computer Methods and Programs in Biomedicine*, vol. 83, no. 2, pp. 125–138, 2006.

[40] C. J. James and D. Lowe, "Extracting multisource brain activity from a single electromagnetic channel," *Artificial Intelligence in Medicine*, vol. 28, pp. 89–104, 2003.

[41] A. M. Fraser, *Using Mutual Information to Estimate Metric Entropy.* Dimensions and Entropies in Chaotic Systems, Berlin: G.Mayer-Kress (ed.), Springer, 1986.

[42] J. B. Elsner and A. A. Tsonis, *Singular spectrum analysis: a new tool in time series analysis.* Plenum Press, 1997.

[43] V. Franc and V. Hlavac, "Statistical Pattern Recognition Toolbox," *http://cmp.felk.cvut.cz/ xfrancv/stprtool/*, 2005.

[44] A. R. Teixeira, A. M. Tomé, and E. W. Lang, "Greedy KPCA in biomedical signal processing," in *LNCS 4669- International Conference on Artificial Neural Networks-ICANN 07*, (Porto, Portugal), pp. 486–495, 2007.

Chapter 3

Linear Subspace Techniques

"Essentially, all models are wrong,
but some are useful."
- George E.P. Box -

Contents

Linear subspace methods have become rather ubiquitous in a wide range of problems arising in computer vision [1], pattern recognition [1], noise reduction [2] and speech enhancement [3], where the high-dimensional representations of certain structures are approximately low dimensional. Linear projective techniques applied to time series datasets, can be found in the literature under several names depending on the application domain: Singular Spectrum Analysis (for instance in climate time series analysis) [2, 4, 5] and SVD (for instance in speech enhancement and pattern recognition) [3, 6, 1]. The aim of SSA is to achieve a decomposition of the original time series into a sum of small number of interpretable components like

oscillatory and noise components, while the aim of speech enhancement methods is simply to eliminate noise which is usually considered additive and normally distributed. There are other linear subspace methods mentioned in literature such as Principal Component Analysis (PCA) [4], Linear Discriminant Analysis (LDA) [7], Independent Component Analysis (ICA) [8], Canonical Correlation Analysis (CCA) [9], Fisher Linear Discriminant (FLD) [10] and Partial Least Squares (PLS) [11].
This work is focused in the SSA technique to denoise time series. The basic SSA analysis will be discussed and a new modified version of singular spectrum analysis called Local SSA will be introduced. The Local SSA algorithm was developed, studied and described in detail in the master thesis [12]. Some works were published along this work focusing on noise elimination in EEG signals [2, 13] and the analysis of protein NMR spectra [14]. It should be noted that the free parameters of the algorithm remained open. Some methods and heuristics were found and discussed to automatically determine the parameters of the algorithm that make it efficient in practical applications.
Another issue discussed is the comparison of subspaces, providing a new insight to the proposed signal processing approach to projective subspace techniques. The SSA is presented using a linear invariant system terminology. The goal is to show that the basis vectors of the multivariate feature space can be interpreted as filter banks extracting different frequency contents of the original time series. The projections and reconstruction step are discussed as a filter bank interpretation.

This chapter is organized as follows: Sections 3.1 and 3.2 resume the main steps of SSA and Local SSA respectively. Here a sinusoidal example was used to show the performance of the algorithms as well as the application of subspace distance measure. Section 3.3 discusses the Local SSA parameters: embedding, number of clusters and number of directions to optimize the algorithm. Section 3.4 discusses the SSA algorithm as a filter bank and the last section presents some conclusions. Note that all the results present in this chapter are published already in [15, 2].

3.1 Singular Spectrum Analysis

In many signal processing applications, the sensor signals are contaminated with noise which is assumed to be additive and non-correlated to the source signals. The general purpose of SSA analysis is the decomposition of a time series into additive components that can be interpreted as "trends", "oscillatory" and "noise" components [4]. This decomposition initializes forecasting procedures for both the original time series and its components. It is a powerful and useful tool of time series analysis in meteorology, hydrology, geophysics, climatology, economics, biology, physics, medicine and other sciences [16, 4, 5, 2, 15]. It can be used for series that are short and long, one-dimensional and multidimensional, stationary and non-stationary, almost deterministic and noisy. The basic SSA algorithm has two stages: decomposition and reconstruction.

- **Decomposition**

  1. Embedding
     Transformation of the unidimensional time series $x[k]$ into a multidimensional time series, $\mathbf{x}_n$ using a M-dimensional window, section 2.4.1. Note that the embedding can be interpreted as a non-linear transformation of the signal.
  2. SVD/PCA decomposition
     In input space the covariance matrix is calculated and its decomposition into eigenvectors and eigenvalues is done.

- **Reconstruction**

  1. Grouping
     Splitting the elementary matrices $\mathbf{X}$ into several groups and summing the matrices within each group.
  2. Diagonal averaging
     Transformation of the multidimensional signal into a unidimensional signal, section 2.4.2.

In the SSA algorithm two news steps were introduced (clustering and unclustering). The goal was to cluster the dataset and project the data locally in the principal directions. The next section will describe in detail the main steps of the basic SSA analysis and introduce the additional steps of the modified version of singular spectrum analysis, called Local SSA.

3.2 Local SSA

Local SSA basically introduces a clustering step into the SSA technique [2] and operates in time delayed coordinates. The formulation follows the steps considered in SSA methods: embedding, SVD, grouping and diagonal averaging. The clustering step was introduced after the embedding phase of those methods, hence applying them locally only. With Local SSA, after embedding, the column vectors $\mathbf{x}_n, n = 1, \ldots, N$ of the trajectory matrix, (eqn. 2.19) are clustered. After clustering, using any clustering algorithm (like k-means [17]) the set of indexes of the columns of $\mathbf{X}$ is subdivided into q disjoint subsets $c_1, c_2, \ldots, c_q$. Thus a sub-matrix $\mathbf{X}^{(c_i)}$ is formed with N_{c_i} columns of the matrix $\mathbf{X}$ which belongs to the subset c_i of indexes. The number N_{ci} of columns in each sub-matrix obeys

$$\sum_{i=1}^{q} N_{c_i} = K - M + 1 \tag{3.1}$$

Note that the model parameter q is naturally upper bounded by the number of data available. However, any reliable estimate needs a sufficient number of data points in each cluster limiting the number of clusters that need to be much less than the number of available data. Normal SSA is obtained by skipping the clustering step, i.e choosing $q = 1$.
In the Local SSA the following steps $(1 - 4)$ need to be repeated for every $i = 1 \ldots q$:

1. A covariance matrix is computed in each cluster using zero mean data obtained via

$$\mathbf{X}_c = \mathbf{X}^{(c_i)}(\mathbf{I} - \frac{1}{N_{c_i}}\mathbf{j}_{c_i}\mathbf{j}_{c_i}^T) \tag{3.2}$$

where $\mathbf{j}_{c_i} = [1, 1, \ldots, 1]^T$ is a vector with dimension $N_{c_i} \times 1$, and $\mathbf{I}$ is a $N_{c_i} \times N_{c_i}$ identity matrix.

2. Next, the eigenvalue decomposition of the covariance matrix is computed, i.e.

$$\mathbf{S}^{(c_i)} = \frac{1}{N_{c_i}}\mathbf{X}_c\mathbf{X}_c^T = \frac{1}{N_{c_i}}\mathbf{S}_c = \mathbf{U}\mathbf{D}\mathbf{U}^\mathbf{T} \tag{3.3}$$

Afterwards denoising can be achieved by projecting the multidimensional signal into the subspace spanned by the eigenvectors corresponding to the $L_{c_i} < M$ largest eigenvalues.

3. The number of significant directions can be found by using a maximum likelihood estimation of the parameter vector of the covariance matrix $\mathbf{S}^{(c_i)}$ of each cluster. This parameter vector comprises the most significant eigenvalues and corresponding eigenvectors and also the variance of the noise which is estimated by the average over the discarded eigenvalues.

4. The eigenvectors related to the largest eigenvalues are used in the reconstruction process. Considering the matrix $\mathbf{U}$ with L_{c_i} eigenvectors in its columns, the reconstructed vectors in each cluster are obtained by

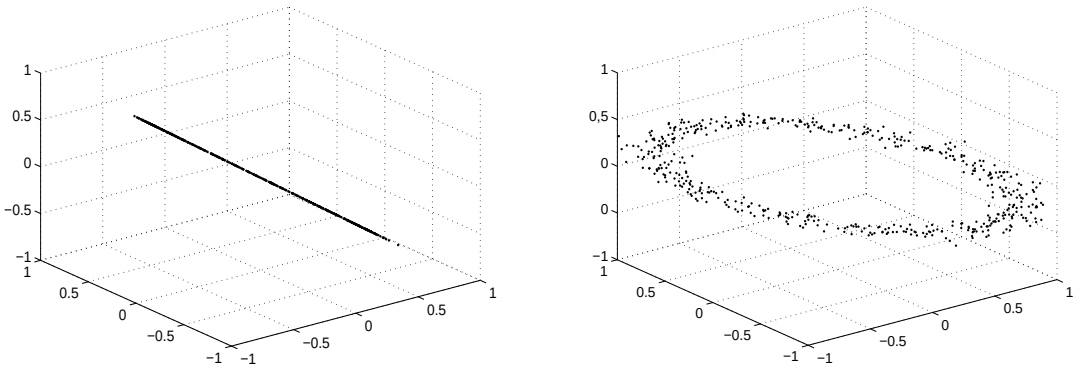
$$\hat{\mathbf{X}}^{(c_i)} = \mathbf{U}\mathbf{Y}^{(c_i)} + \frac{1}{N_{c_i}}\mathbf{X}^{(c_i)}\mathbf{j}_{c_i}\mathbf{j}_{c_i}^T \tag{3.4}$$

where $\mathbf{Y}^{(c_i)} = \mathbf{U}^T\mathbf{X}^{(c_i)}$ corresponds to the projections of the dataset $\mathbf{X}^{(c_i)}$ onto the basis vectors $\mathbf{U}$ with L_{c_i} eigenvectors in its columns. Thus $\hat{\mathbf{X}}^{(c_i)}$ represents the reconstructed version of the original dataset in each cluster. If all M eigenvectors are selected, the original $\mathbf{X}^{(\mathbf{c_i})}$ is recovered because $\mathbf{U}\mathbf{U}^T = \mathbf{I}$. This reconstruction has to be done for each cluster separately.

After reconstructing all sub-matrices $\hat{\mathbf{X}}^{(c_i)}$, $i = 1, ..., q$, the clustering process must be reverted to obtain an estimate $\hat{\mathbf{X}}$ of the reconstructed, noise-free trajectory matrix using the columns of the extracted sub-matrices, $\hat{\mathbf{X}}^{(c_i)}$, $i = 1, ..., q$, according to the contents of subsets c_i - unclustering step.
Then, one-dimensional sensor signals are embedded in the space of their time-delayed coordinates [18] to form a trajectory matrix, whose column vectors span the so called embedding space. To revert the embedding process, the last step is to transform the denoised trajectory matrix, whose columns are formed by the images of the reconstructed (=denoised) vectors, into a one-dimensional time series with K samples in time domain - diagonal averaging.

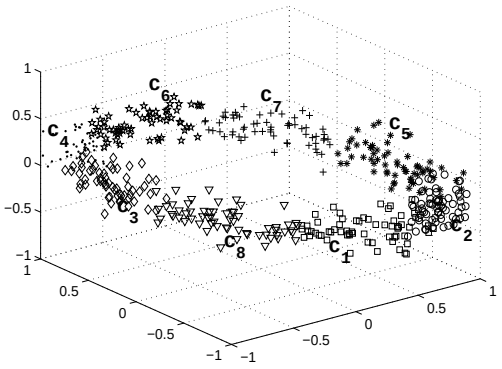
3.2.1 Illustrative Example

The Local SSA algorithm was applied in real and artificial datasets [19, 15, 14]. In all applications the goal was to remove noise without distorting the signal. The algorithm was applied directly to the signal without selecting any parameter.
A noisy sinusoidal signal (20dB) embedded in 3D is used to illustrate the performance of the algorithm, section A.1 .

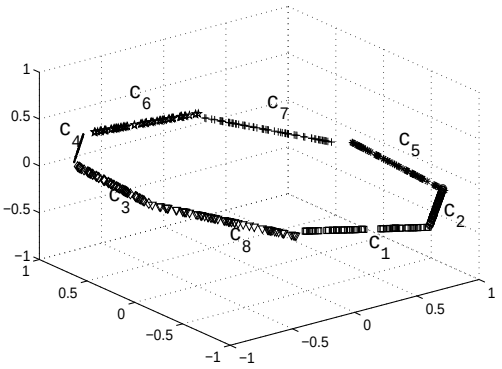


(a) SSA - $L = 1$

(b) SSA - $L = 2$



(c) Clustering



(d) Local SSA

Figure 3.1: Illustrative example: Reconstruction using SSA with $L = 1$ (a), $L = 2$ (b), Clustering the data: $q = 8$ (c), Local SSA: reconstruction with $L = 1$ in each cluster(d).

Using plain SSA, the data can be projected into $L = 1$ or $L = 2$ principal directions to achieve the denoising signal. With $L = 1$ a straight line will result, figure 3.1(a) and with $L = 2$ the reconstructed version is similar to the noisy version, figure 3.1(b). Using Local SSA, which incorporates a clustering step in SSA, an improved solution is achieved.
Applying Local SSA using $q = 8$ clusters, figure 3.1 (c) and projecting the data in each cluster onto the direction related to the largest eigenvector ($L = 1$), the underlying trajectory in phase space could be reconstructed satisfactorily in a piecewise linear way, figure 3.1 (d).

Subspace Distance - Interpretation

Consider the illustrative example of Local SSA, figures 3.1 (c), (d). In each cluster, a covariance matrix ($\mathbf{S}^{(c_i)}$) was computed and an eigendecomposition was performed. The eigenvectors were considered and the subspace distances between the subspace models computed in each clusters $c_i, i = 1, ..., 8$ are calculated, figure 3.2. The distance values range between $[0 \;\; 1[$, where $d = 0$ occurs when the distance is computed between the same subspace models. The results show that the minimum distance was found when the clusters had the same principal direction, i.e. the same trajectory. For example, when the principal direction of the cluster c_1 is parallel to the principal direction of the cluster c_6, the subspace distance is minimal, ($d = 0.13$). To understand the influence of the clusters in the reconstruction process, two

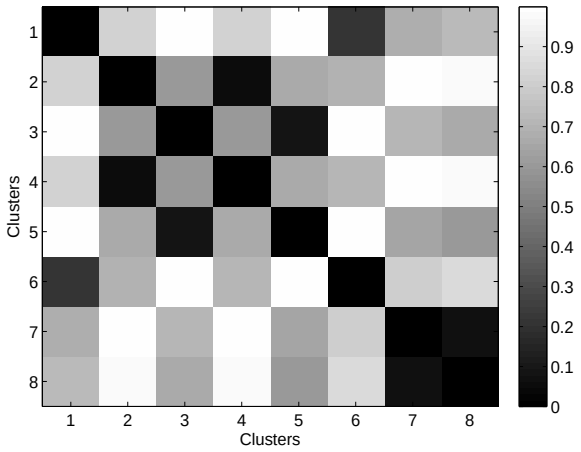


Figure 3.2: Subspace distance between the subspace models for each cluster represented in figure 3.1.

experiments were done. For each cluster represented in figure 3.1 (c), the mean was removed. The SVD was applied to the new dataset using two strategies:

1. Using all the dataset. The reconstructed dataset represents only one direction, figure 3.3 (a). In the end, the mean of each cluster was added to the reconstructed dataset and the result is presented in figure 3.3 (b). The trajectory of the dataset was lost because all the directions in each cluster are parallels with different centers.

2. Splitting the data into four clusters using the subspace distance information, i.e. cluster1=$c_1 \cup c_6$; cluster2=$c_2 \cup c_4$; cluster3=$c_3 \cup c_5$ and cluster4=$c_7 \cup c_8$. The mean of clusters was added to the reconstructed data and the result shows that the trajectory of the original dataset is maintained.

It can be concluded by the results presented in figure 3.3, that the subspace distance allows the consistent grouping of the dataset. Furthermore, splitting the data into clusters is necessary to find the different directions to reconstruct the original trajectory.

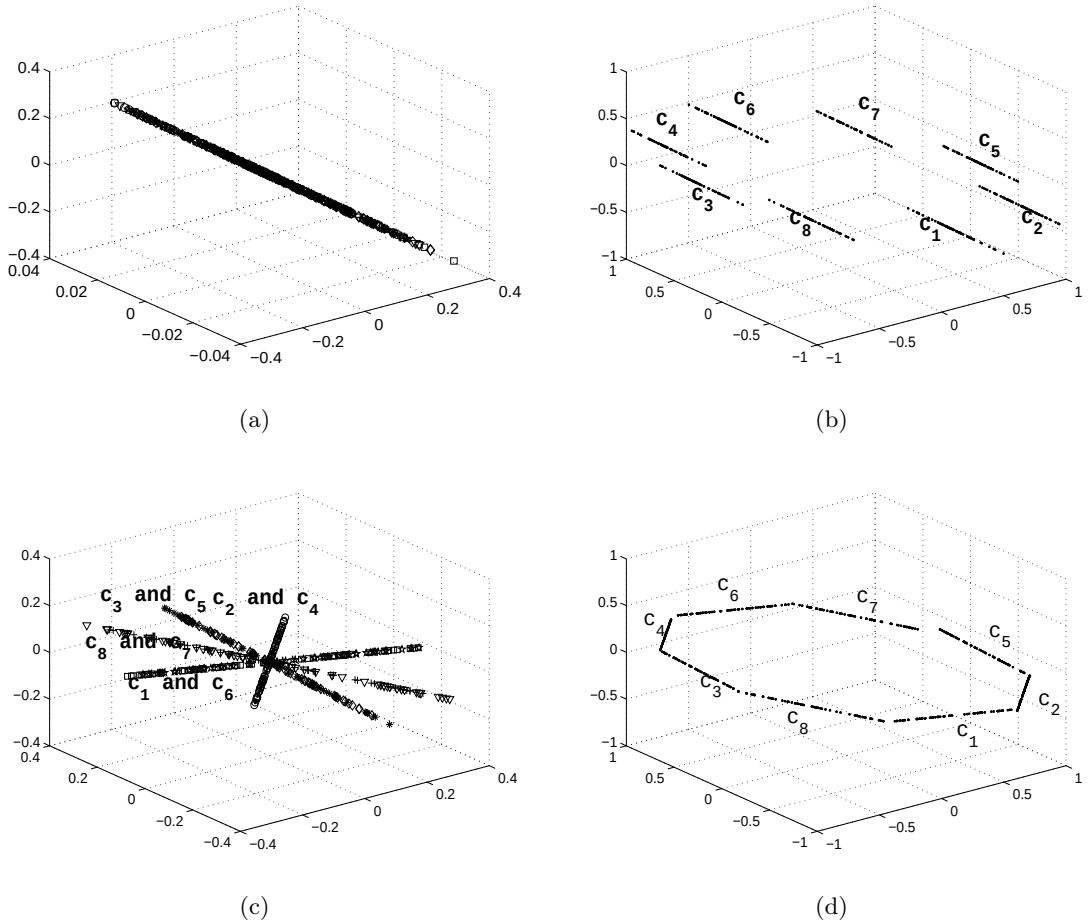


Figure 3.3: The mean of each cluster was removed from the dataset of figure 3.1 (c). Reconstruction using (a) SSA without the mean of the cluster ($L = 1$); (b) SSA added the mean of each cluster; (c) Local SSA $q = 4$ and ($L = 1$) without the mean of the cluster; (d) Local SSA added the mean of the cluster.

3.3 The Parameters of the Local SSA Algorithm

The implementation of the algorithm is achieved by following the steps already described after assignment of the following parameters: the embedding dimension (M) and the number of clusters (q) to split the columns of the trajectory matrix. A third parameter can be chosen by the user or automatically assigned using the MDL criterion: the number of significant directions L_{c_i} to project and reconstruct the multidimensional vectors.

3.3.1 Embedding Dimension and Number of Clusters

In SSA applications, the embedding dimension is discussed and it is recommended to be approximately half of the segment length (K) [4]. The embedding dimension should also be close to the periodicity of the signal to be extracted [4]. A similar strategy is followed in [20] where a lower bound chosen according to the resolution in frequency is recommended so that

each column of the trajectory matrix should have as dimension:

$$M > \frac{f_s}{f_r} \tag{3.5}$$

where f_s represents the sampling frequency and f_r the minimum frequency to be extracted. Along this work the last criterion was chosen to select the embedding dimension in denoising applications. Beyond the embedding dimension, the number of clusters in each cluster (N_{c_i}) also need to be assigned by the user. The number of samples N constitute the natural threshold once the eigendecomposition performance in each cluster is related to the number of points to perform the estimates. In particular the cardinality of each cluster can not be lower than the embedding dimension,

$$\#(c_i) > M \tag{3.6}$$

The number of clusters is automatically assigned to each signal using an heuristic rule that aims to prevent overfitting by MDL but simultaneously uses the maximum number of clusters. The number of clusters starts with a maximum value q_{max} checking afterwards if all clusters end up with a cardinality higher than M and if the number of directions in each cluster is $L < \frac{M}{2}$. If both criteria are not met, then the number q of clusters is decreased and the process is repeated until a reliable decomposition in each cluster is achieved.

3.3.2 Number of Directions

In [12, 2] the performance of MDL and AIC were studied in the selection of the number of directions. In the literature several studies [21, 22, 23, 24, 25] present MDL and AIC as selection criteria to estimate the dimension of subspace signals in many applications. Different experiments were done changing the dataset size and the SNR added. Comparing the MDL and AIC criteria, the results showed that the MDL estimates the size of the subspace more consistently, while Akaike's criterion tends to achieve an even stronger overfitting, as referred in [21]. So, in this work, the determination of the number of significant directions is based on the MDL criterion. This method was originally proposed in [26] and became popular in the signal processing analysis [27].
The number of directions is obtained by the maximum likelihood estimation of the parameter vector of the covariance matrix of each cluster, which is

$$\boldsymbol{\theta} = (\lambda_1, \lambda_2, \ldots, \lambda_L, \sigma^2, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L) \tag{3.7}$$

where $\lambda_i,\ i = 1, \ldots, L$ are the L largest eigenvalues of the covariance matrix, $\mathbf{u}_i$ the corresponding eigenvectors and σ^2 corresponds to the mean of the discarded eigenvalues [21].
Using the maximum likelihood estimate of $\widehat{\boldsymbol{\theta}}$, L will be the value that minimizes the following expression

$$MDL(L) = -\ln f(\mathbf{X}^{c_i}|\widehat{\boldsymbol{\theta}}) + \frac{1}{2} P \ln N_{c_i},\ L = 0, \ldots, M-1 \tag{3.8}$$

where N_{c_i} is the number of observations available to estimate the covariance matrix and $f(\mathbf{X}^{c_i}|\widehat{\boldsymbol{\theta}})$ denotes the conditional probability density parameterized by $\widehat{\boldsymbol{\theta}}$. The log likelihood function $L(\widehat{\boldsymbol{\theta}}) = \ln f(\mathbf{X}^{c_i}|\widehat{\boldsymbol{\theta}})$ represents the accuracy of the data with the parameter vector $\widehat{\boldsymbol{\theta}}$ and depends on the discarded eigenvalues $M - L$

$$L(\widehat{\boldsymbol{\theta}}) = N_{c_i}(M-L)\ln\left[\frac{\prod_{i=L+1}^{M}\lambda_i^{1/(M-L)}}{\frac{1}{M-L}\sum_{i=L+1}^{M}\lambda_i}\right] \tag{3.9}$$

The negative log-likelihood $-L(\widehat{\boldsymbol{\theta}})$ is recognized to be a standard measure of the training error. However it has been reported that the simple maximization of this term tends to result in the phenomenon of overfitting. Thus the second term in eqn. 3.8 was added as a regularization term to penalize complexity. The value of P is related to the number of parameters in $\boldsymbol{\theta}$ and to the complexity of its estimation. Considering real value signals, the value of P is computed according to

$$P = L + 1 + ML - \frac{L^2}{2} - \frac{L}{2} = ML - \frac{L^2}{2} + \frac{L}{2} + 1 \tag{3.10}$$

Substituting the last P value in equation 3.8, the general expression for the MDL criteria is obtained by:

$$MDL(L) = -\ln f(\mathbf{X}^{c_i}|\widehat{\boldsymbol{\theta}}) + \frac{1}{2}(ML - \frac{L^2}{2} + \frac{L}{2} + 1)\ln N_{c_i},\ L = 0, \ldots, M-1 \tag{3.11}$$

The number of directions L is determined by the value $L \in \{0, 1, ..., M-1\}$ that minimizes the MDL criterion. A simple alternative to elaborate the model order, is to fix the number of relevant directions instead. In some applications even a single direction $L = 1$ suffices. Another very popular alternative seen in the literature to find L is based on explained variance [28]. Assuming the eigenvalues ordered by decreasing order, one possible strategy to find L is defining a threshold th in the range of $80\% - 95\%$ and computing the following inequality

$$\frac{\lambda_1 + \lambda_2 + ... + \lambda_L}{\lambda_1 + \lambda_2 + ... + \lambda_L + ... + \lambda_M} * 100 > th \tag{3.12}$$

Therefore L is computed by defining the percentage of variance of the data that should be kept.

3.4 SSA and Filter Banks

In this section, the SSA algorithm is discussed using linear invariant systems terminology. The extracted components are considered outputs of linear invariant systems with finite response filter characteristics. The number of systems is determined by the embedding dimension and the selection of informative components is discussed in terms of frequency response of the systems. The goal is to show that the basis vectors of the embedding space can be interpreted

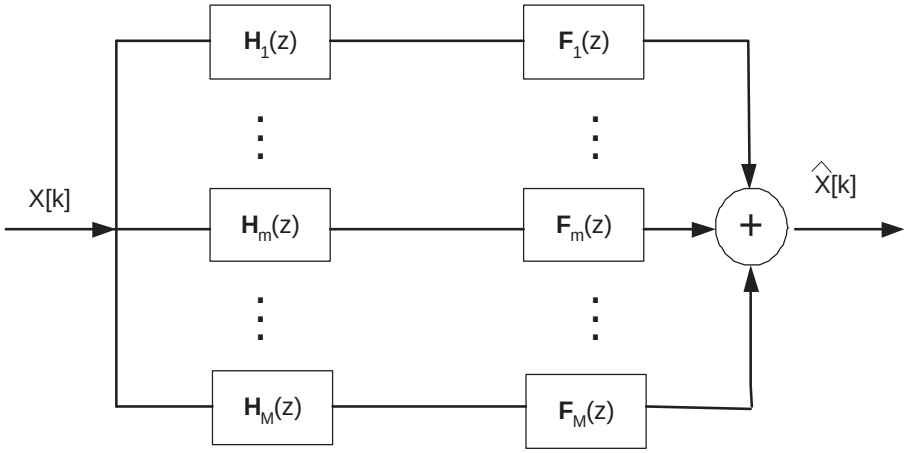


Figure 3.4: Filter bank description of the processing chain: $H_m(z)$ are analysis transfer functions and $F_m(z)$ are synthesis transfer functions.

as filter banks extracting different frequency contents of the original time series.
In the framework of linear invariant system theory, the filter bank structure needed to achieve the output time series $\hat{x}[k]$ should be provided by the input time series $x[k]$ instead of the trajectory matrices. Hence, it was proposed an approach based on filter responses and related transfer functions rather than on matrix manipulation. The approach proposed in this work is summarized in figure 3.4. The figure represents the block diagram of the filter bank system where the transfer functions $H_m(z)$ of the analysis filter are related with the projection step onto the eigenvectors of the subspace model and the transfer functions $F_m(z)$ of the synthesis filters are related with the reconstruction and diagonal averaging step. The reconstruction and projection steps described in the last chapter by eqn. 2.7 and eqn. 2.8 can be written as

$$\begin{aligned} \hat{\mathbf{X}} &= \mathbf{u}_1\mathbf{u}_1^T\mathbf{X} + \mathbf{u}_2\mathbf{u}_2^T\mathbf{X} + ... + \mathbf{u}_p\mathbf{u}_p^T\mathbf{X} + ... + \mathbf{u}_M\mathbf{u}_M^T\mathbf{X} \\ &= \mathbf{u}_1\mathbf{Y}_1 + \mathbf{u}_2\mathbf{Y}_2 + ... + \mathbf{u}_M\mathbf{Y}_M \\ &= \sum_{m=1}^{M} \hat{\mathbf{X}}_m \end{aligned} \tag{3.13}$$

In the last equation, each eigenvector $\mathbf{u}_i$ is a M dimensional vector. Note that in noise reduction only a small number of directions is used ($L \leq M$), so the eigenspectrum of $\hat{\mathbf{X}}$ is a truncated version of the original eigenspectrum of the data.
In the next sections, the projections and the reconstruction step are going to be presented as well as the filtering interpretation.

3.4.1 Projections

The projections in each eigenvector represent distinct frequency contents of the signal. Selecting the largest eigenvalues, the projections on the corresponding eigenvalues will extract the components whose frequency contents contribute the most to the energy of the signal. Each row of the projected data $\mathbf{Y}$ can be considered a filtered version of the original data

sequence. Each row Y_m, $m = 1, 2, ..., M$, is obtained via

$$\mathbf{Y}_m = \mathbf{u}_m^T \mathbf{X} \tag{3.14}$$

where $\mathbf{X}$ is $M \times N$ data matrix and $\mathbf{u}_m^T$ is $1 \times M$ vector, eqn. 2.7. Thus, each element of the matrix $\mathbf{Y}_m$ is the dot product between the $m-th$ eigenvector and one of the columns of the data matrix. This operation, however, can be formulated as well as the weighted sum of a sequence of samples of the time series,

$$y_m[k] = \sum_{i=1}^{M} u_{im} x[k - i + 1] \tag{3.15}$$

where $(M-1) \leq k < K$, then all the columns of the dataset $\mathbf{X}$ are manipulated to obtain an output sample. The element $y_m[k]$ of the row $\mathbf{Y}_m$ of the matrix $\mathbf{Y}$ is taken by natural order. The row vector $\mathbf{Y}_m$ has N samples of the time series $y_m[k]$, starting with time index $(M-1)$, much like the first row of the trajectory matrix $\mathbf{X}$. The entries of the vector $\mathbf{u}_m$ are the coefficients of the finite impulse response (FIR) filter. The transfer function of the analysis step, $H_m(z)$, can be computed by substituting in eqn. 3.15 every delay operation by the corresponding z transform. Therefore mapping $x[k]$ to $X(z) = \sum_{-\infty}^{+\infty} x[k] z^{-k}$, $x[k \pm d]$ to $z^{\pm d} X(z)$ and $y_m[k]$ to $Y_m(z)$ [29], where z is a complex number. The filtering operation can be formulated as

$$Y_m(z) = H_m(z) X(z) \tag{3.16}$$

whereby

$$\begin{aligned} H_m(z) &= \tfrac{Y_m(z)}{X(z)} \\ &= \sum_{i=1}^{M} u_{im} z^{-(i-1)} \\ &= u_{1m} + u_{2m} z^{-1} + ... + u_{Mm} z^{-(M-1)} \end{aligned} \tag{3.17}$$

The transfer function $H_m(z)$, $m = 1, ..., M$ is an output-input ratio and constitutes the analysis block as it decomposes the input time series $x[k]$ into several components time series $y_m[k]$, $m = 1, ..., M$. After filtering out the subspace projections at the analysis step, these projections can be further processed by selecting only the problem-relevant components and discarding the rest. Afterwards the remaining problem-relevant projections need to be combined to the reconstructed signal at the synthesis step as will be explained in the next section.

3.4.2 Reconstruction

The extracted components of the time series are related to the filtered versions of $x[k]$. The m component is obtained by

$$\hat{\mathbf{X}}_m = \mathbf{u}_m \mathbf{Y}_m = \begin{bmatrix} u_{1m}y_m[M-1] & u_{1m}y_m[M] & \dots & u_{1m}y_m[K-1] \\ u_{2m}y_m[M-1] & u_{2m}y_m[M] & \dots & u_{2m}y_m[K-1] \\ u_{3m}y_m[M-1] & u_{3m}y_m[M] & \dots & u_{3m}y_m[K-1] \\ \vdots & \vdots & & \vdots \\ u_{Mm}y_m[M-1] & u_{Mm}y_m[M] & \dots & u_{Mm}y_m[K-1] \end{bmatrix} \tag{3.18}$$

As it can be seen, each row is a scaled version of the same component time series $y_m[k]$. Obviously, the resulting matrix does not have the Toeplitz structure of the original trajectory matrix. But by replacing the entries in each diagonal of $\hat{\mathbf{X}}_m$ by their average, a Toeplitz matrix is obtained again. Interestingly, the diagonal averaging can equally be formulated as a linear filtering operation

$$\hat{x}_m[k] = \frac{1}{M_d} \sum_{i=l}^{s} u_{im} y_m[k+i-1] \tag{3.19}$$

where the values M_d, l and s have values according to the number of elements in the diagonals of the matrix defined in eqn 3.18. More specifically, the response can be sub-divided into a transient and a steady state response according to the following distinction:

- with M elements, the eqn. 3.19 represents the steady state response of the filter which corresponds to $(M-1) \leq k \leq (K-M)$ and $M_d = M$, $l = 1$ and $s = M$.
- with $< M$ elements, the eqn. 3.19 represents the transitory response of the filter in case of
  - if $0 \leq k \leq (M-2)$ (lower left corner of the matrix) then $M_d = k+1$, $l = M - M_d$ and $s = M$.
  - if $(K-M+1) \leq k \leq (K-1)$ (upper right corner of the matrix) then $M_d = K-k$, $l = 1$ and $s = M - M_d$.

Whatever the case, the synthesis filter is an anti-causal FIR filter, each output at time index k depends on the input samples with the time indices $k+1, ..., k+M$. Both cases can be unified by formally setting $y_m[k] = 0$ for time indices $0 \leq k \leq (M-2)$ and $K \leq k \leq (K+M-2)$ and compute eqn. 3.19 as in the steady-state case. Therefore, the synthesis transfer function is given by:

$$\begin{aligned} F_m(z) &= \frac{\hat{X}_m(z)}{Y_m(z)} \\ &= \frac{1}{M} \sum_{i=1}^{M} u_{jm} z^{i-1} \\ &= \frac{1}{M_d}(u_{1m} + u_{2m}z^1 + ... + u_{Mm}z^{M-1}) \end{aligned} \tag{3.20}$$

where $1 \leq M_d \leq M$ are the number of samples in the diagonals. Notice that the analysis and synthesis transfer functions differ by a scale factor $(1/M_d)$ and by the sign of the powers of z. Therefore the magnitudes of the frequency response of both filters are related by a scale factor $(1/M_d)$ and their phase are symmetric.
The transfer function of the global system is a cascade formed by the projection step (analysis filter) followed by the reconstruction and diagonal averaging step (synthesis filter). Hence, the product of the two transfer functions reads

$$\begin{aligned} T_m(z) &= \frac{\hat{X}_m(z)}{X(z)} \\ &= F_m(z)H_m(z) \\ &= \sum_{k=-M+1}^{M-1} t_{km} z^k \end{aligned} \tag{3.21}$$

The coefficients t_{km} are the coefficients of the product of two polynomials with the same coefficients but with symmetric powers. It can be easily shown that the impulse response is a sequence $t_m[k]$ with non-zero values for $k = -M+1, -M+2, ..., 0, ..., M-2, M-1$ formed by the coefficients t_k of the eqn. 3.21. The impulse response is non-causal and symmetric $t_m[k] = t_m[-k]$, thus a zero phase filter. Therefore, the frequency response $T_m(e^{jw})$ can be obtained by substituting $z = e^{jw}$, where $j = \sqrt{-1}$, in eqn. 3.21 which then leads to

$$T_m(e^{jw}) = t_{0m} + \sum_{k=1}^{M-1} 2t_{km} cos(kw) \tag{3.22}$$

The frequency response is a periodic real function, with period equal to $w = 2\pi$ (the normalized sampling rate), so it corresponds to a zero-phase filter. This means that each extracted component $\hat{x}_m[k]$ is then always in-phase with its related original time series $x[k]$. Furthermore the sequences $y_m[k]$, $m = 1, ..., M$ corresponding to the outputs of $\mathbf{u}_m$, $m = 1, ..., M$ and having as input $x[k]$, the sequences are orthogonal, which represent filtered versions of the input. This aspect can be verified using the SVD decomposition of the original data, i.e. $\mathbf{X} = \mathbf{U}\mathbf{D}^{1/2}\mathbf{V}^T$, and substituting into eqn. 3.14. This yields

$$\begin{aligned} \mathbf{Y}_m &= \mathbf{u}_m^T \mathbf{X} \\ &= \lambda^{1/2} \mathbf{v}_m^T \end{aligned} \tag{3.23}$$

The sequence $y_m[k]$ is related to the mth eigenvector $(\mathbf{v}_m)$ of the inner products $(\mathbf{X}^T\mathbf{X})$ of the data and consequently is orthogonal. Several components can be extracted also by projecting and afterwards reconstructing the data onto several directions corresponding to the addition of the corresponding components $\hat{x}_m[k]$. Furthermore it is possible to conclude that the eqn. 3.14 corresponds to a filter bank with blocks T_m in parallel, figure 3.4. The resulting output $\hat{x}[k]$ is a sum of the selected signals $\hat{x}_m[k]$, i.e., the outputs of the cascaded filter pairs formed by $H_m(z)$ and $F_m(z)$.

3.4.3 Illustrative Example

To illustrate the concepts discussed above, a numerical simulation was performed. Three sinusoids with different angular frequencies ($w = 0.1\pi, 0.2\pi, 0.4\pi$) were added resulting in a new input signal to SSA algorithm, figure 3.5. The models were computed for distinct embedding dimension, taking $K = 500$ samples of the input signal $x[k]$. The goal was to show the influence of the M parameter in the filter banks described above. The spectrum

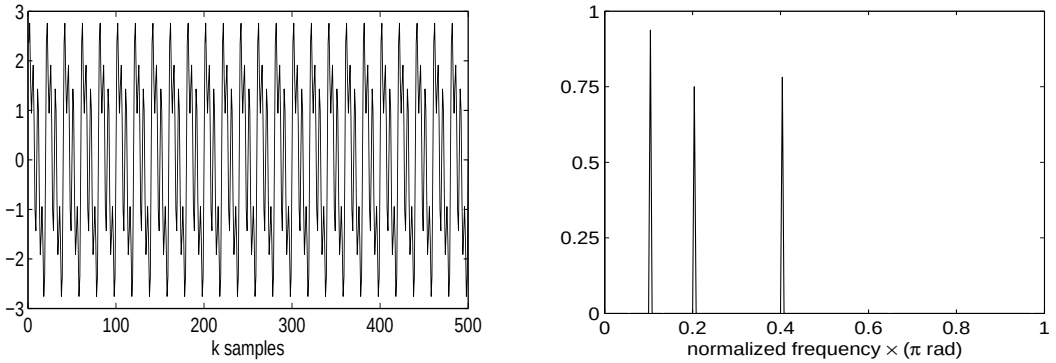


Figure 3.5: The sinusoidal time series and its Discrete Fourier Transformer (DFT).

of the time series exhibits three large peaks. The basis vectors $\mathbf{U}$ of the space of time-delayed coordinates correspond to eigenvalues by decreasing value. Figure 3.6 illustrates the frequency response of the global system $T_m(z)$ for $M = 3$, $M = 8$, $M = 15$ and $M = 50$ distinct embedding dimensions. In the example, for $M = 3$ three directions are available. The filter associated to the largest eigenvalue is centered at $w = 0$ and covers the low frequency band of the signal. The filter associated to the second eigenvalue is centered at $w = \frac{\pi}{2}$ and the third filter associated to the third eigenvalue is centered at $w = \pi$. Although each filter is centered in different energy bands, there is an overlap of themselves, figure 3.6. All the three angular frequencies are associated to the first filter. For $M = 8$ and $M = 15$ the first two angular frequencies are presented in the two first filters. Using $M = 50$ each filter is associated to each angular frequency (sinusoidal signal) and the order of the filters is according to the energy of the sinusoidal signals. Increasing the amount of M, the bandwidth of the filters decreases, figure 3.6. According to equation 3.5, to extract $w = 0.1\pi$, the M value should be $M > \frac{2}{0.1} \Rightarrow M > 20$. Using $M = 21$, the result is analog to the result obtained with $M = 50$. Changing the value of the embedding dimension, the profile of the filter also changes and the number of systems is determined by the embedding dimension. In conclusion there is a relation between the embedding dimension and the eigenvalues of the model and the selection of informative components is related to the frequency response of the filter.

3.5 Conclusion

In this chapter the application of linear subspace techniques in time series was presented and discussed. The main steps of the SSA method were presented and explained. This method

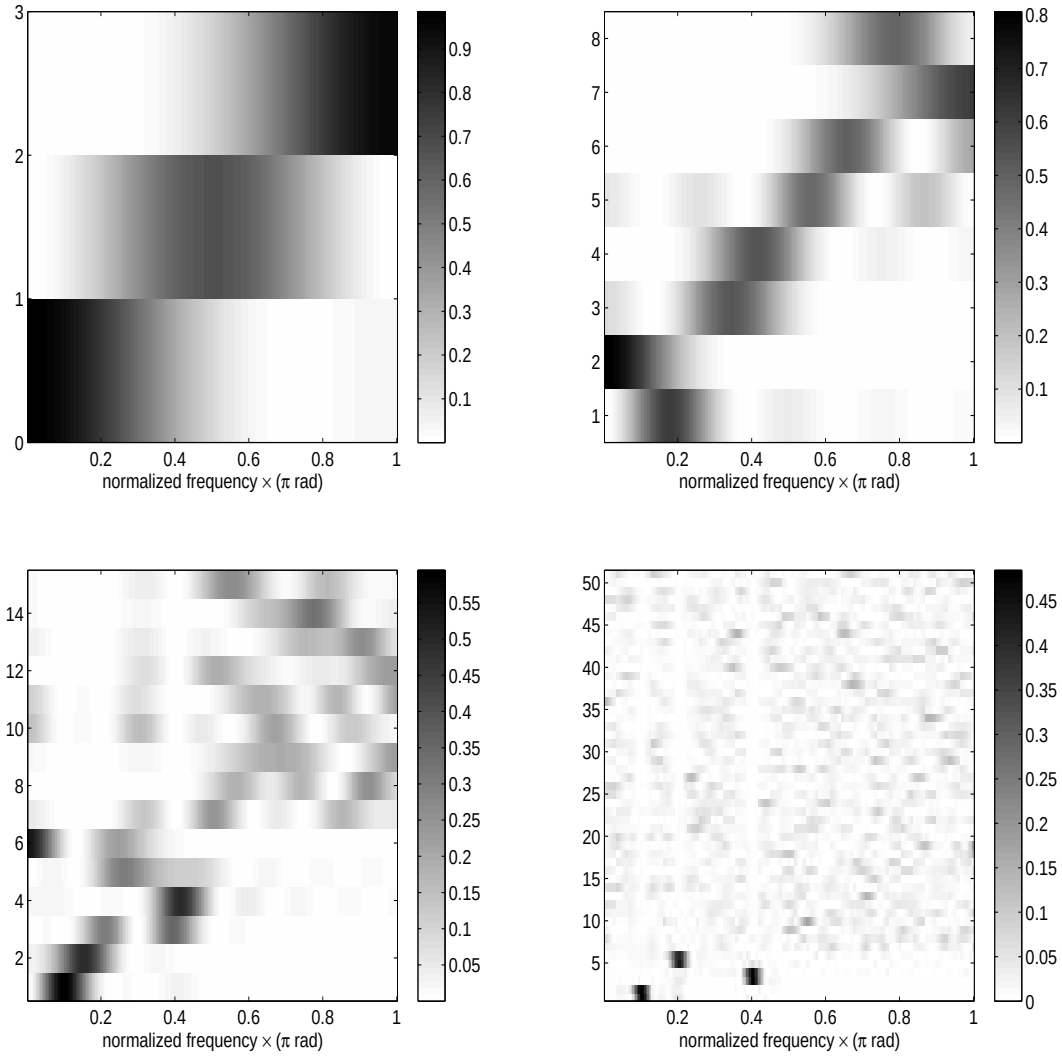


Figure 3.6: Frequency response $T_m(e^{jw})$. (a) M=3 filters, (b) M=8 filters, (c) M=15 filters and (d) M=50 filters for the sinusoidal time series.

is clearly a linear approach to multidimensional representation of the data and to time series components as they are filtered versions of the original time series. In this chapter it was shown that the basis vectors of the embedding space can be interpreted as filter banks extracting different frequency contents of the original time series. This interpretation can be useful to attain a more clear-cut insight into the outcomes of the method. By the frequency responses of the filter bank, corresponding to the basis vectors of the subspace, the frequency content of different components can be easily attained. SSA filters are data adaptive and the relevance of one component to the energy of the input signal is deducted from the corresponding eigenvalue. Moreover, the frequency profile of each component is determined only at the projection step. In order to get a component in phase with the input signal, the diagonal averaging is required. The possibility of having outputs that are in phase with the inputs is an important aspect in applications where measures have to be taken using a small number

of samples.

After considering the multidimensional representation of the data (the trajectory matrix) and its characteristics, a new algorithm called Local SSA was introduced. This algorithm is an extension of SSA that incorporates a clustering step (i.e a grouping of the trajectory matrix in sub-matrices). After clustering, SSA was used locally in each cluster. Then, the reconstruction and the reverting clustering were done to obtain the denoised dataset.

Another key point to consider about is that the Local SSA algorithm presented has a fully automated choice of the parameters. Some heuristics were found to compute the parameters. These are adjusted according to the input of the algorithm (number of clusters and the dimension of the embedding). Furthermore, the Local SSA does not require any user intervention to select the components of the reconstruction. The MDL criterion is used as default to select the relevant directions.

Another issue discussed in this chapter is the application of a similarity measure to compare subspace models.

Bibliography

[1] P. Chen and D. Suter, "An Analysis of Linear Subspace Approaches for Computer Vision and Pattern Recognition," *International Journal of Computer Vision*, vol. 68, pp. 83–106, June 2006.

[2] A. R. Teixeira, A. M. Tomé, E. Lang, P. Gruber, and A. M. Silva, "Automatic removal of high-amplitude artifacts from single-channnel electroencephalograms," *Computer Methods and Programs in Biomedicine*, vol. 83, no. 2, pp. 125–138, 2006.

[3] Y. Ephraim and H. L. Van Trees, "A Signal Subspace Approach for Speech Enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, 1995.

[4] N. Golyandina, *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman & Hall/CRC, 2001.

[5] M. Ghil, M. Allen, M. D. Dettinger, K. Ide, and e. al, "Advanced spectral methods for climatic time series," *Reviews of Geophysics*, vol. 40, no. 1, pp. 3.1–3.41, 2002.

[6] P. C. Hansen and S. H. Jensen, "Subspace-based noise reduction for speech signals via diagonal and triangular matrix decompositions: Survey and analysis," *Eurasip Journal on Advances in Signal Processing*, vol. 2007, 2007.

[7] J. R. Price and T. F. Gee, "Face recognition using direct, weighted linear discriminant analysis and modular subspaces," *Pattern Recognition*, vol. 38, no. 2, pp. 209–219, 2005.

[8] F. J. Theis, A. Jung, C. G. Puntonet, and E. W. Lang, "Linear geometric ICA: Fundamentals and Algorithms," *Neural Computation*, vol. 15, pp. 1–21, 2002.

[9] K. E. Muller, "Understanding Canonical Correlation Through the General Linear Model and Principal Components," *The American Statistician*, vol. 36, no. 4, p. 342, 1982.

[10] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[11] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," *Subspace, Latent Structure and Feature Selection Techniques, Lecture Notes in Computer Science*, pp. 34–51, 2006.

[12] A. Teixeira, *Técnicas não lineares baseadas em componentes principais no estudo de séries temporais*. Departamento de Electrónica e Telecomunicações, Aveiro University, Master Thesis, 2005.

[13] A. R. Teixeira, A. M. Tomé, E. W. Lang, P. Gruber, and A. M. Silva, "On the use of clustering and local singular spectrum analysis to remove ocular artifacts from electroencephalograms," in *IJCNN2005, IEEE*, (Montréal, Canada), pp. 2514–2519, 2005.

[14] P. Gruber, K. Stadlthanner, M. Böhm, F. J. Theis, E. W. Lang, A. M. Tomé, A. R. Teixeira, C. G. Puntonet, and J. M. Gorriz Saéz, "Denoising using Local Projective Subspace Methods," *Neurocomputing*, vol. 69, no. 13-15, pp. 1485–1501, 2006.

[15] A. R. Teixeira, A. M. Tomé, E. W. Lang, P. Gruber, and A. M. Silva, "Extraction and separation of high-amplitude artifacts in electroencephalograms from epileptic patients," in *Fourth IASTED International Conference on Biomedical Engineering- BIOMED2006* (C. Ruggiero, ed.), (Innsbruck, Austria), pp. 270–275, IASTED, 2006.

[16] A. S. Sharma, D. Vassiliadains, and K. Papadopoulos, "Reconstruction of low-dimensional magnetospheric dynamics by Singular Spectrum Analysis," *Geophysical Research Letters*, vol. 20, no. 5, pp. 335–338, 1993.

[17] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995.

[18] C. H. You, S. N. Koh, and S. Rahardja, "Signal subspace speech enhancement for audible noise reduction," in *ICASPP 2005*, vol. I, (Philadelphia, USA), pp. 145–148, IEEE, 2005.

[19] A. R. Teixeira, A. M. Tomé, E.W.Lang, P. Gruber, and A. M. Silva, "On the use of clustering and local singular spectrum analysis to remove ocular artifacts from electroencephalograms," in *IJCNN2005*, (Montréal, Canada), pp. 2514–2519, IEEE, 2005.

[20] C. J. James and D. Lowe, "Extracting multisource brain activity from a single electromagnetic channel," *Artificial Intelligence in Medicine*, vol. 28, pp. 89–104, 2003.

[21] A. P. Liavas and P. A. Regalia, "On the behaviour of information thereotic criteria for model order selection," *IEEE Trans. Signal Process*, vol. 49, no. 8, pp. 1689–1695, 2001.

[22] Z. Leonowicz, J. Karvanen, T. Tanaka, and J. Rezmer, "Model order selection criteria: comparative study and applications," in *International Workshop Computational Problems of Electrical Engineering*, 2004.

[23] P. Gruber, K. Stadlthanner, A. M. Tomé, A. R. Teixeira, F. J. Theis, C. G. Puntonet, and E. W. Lang, "Denoising using local ICA and a generalized eigendecomposition with time-delayed signals," in *ICA 2004* (C. G. Puntonet and A. Prieto, eds.), LNCS 3195, (Granada- Spain), pp. 993–1000, Springer, 2004.

[24] P. Stoica and Y. Selén, "Model-Order Selection-A Review of Information Criterion Rules," *IEEE Signal Processing Magazine*, vol. 21, pp. 36–46, July 2004.

[25] Z. Leonowicz, J. Karvanen, T. Tanaka, and J. Rezmer, "Model Order Selection Criteria: Comparative Study and Applications," in *Computational Problems of Electric Engineering (CPEE2004)*, (Zakopane, Poland), pp. 193–196, 2004.

[26] G. Schwartz, "Estimating the Dimension of a Model," *Journal of Artificial Intelligence Research*, vol. 6, pp. 461–464, 1978.

[27] M. Wax and T. Kailath, "Detection of Signals by Information Theoretic Criteria," *IEEE Transactions on Acousticas, Speech and Signal Processing*, vol. ASSP-33, pp. 387–392, April 1985.

[28] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[29] L. B. Jackson, *Digital Filters and Signal Processing*. Klumer Academic Publishers, 1996.

Chapter 4

Non-Linear Subspace Techniques

"Everything of importance has been said before
by somebody who did not discover it."
- Alfred North Whitehead -

Contents

In many real world applications, work in feature space can increase the overall performance of the algorithms. Non-linear subspace methods using kernel functions have received increasing attention in recent years. Support Vector Machine (SVM) [1], Kernel Principal Component Analysis (KPCA) [2, 3, 4], Greedy KPCA [5], Kernel Independent Component Analysis (KICA) [6] and Kernel Fisher Discriminant Analysis (KFDA)[7] are some examples of such techniques. The advantage of using these techniques focuses on two points. First it is possible to extract a number of principal components that exceed the dimensionality of the input data. Notice that having $N \geq M$ examples of data with dimension M working in input space, the maximum number of nonzero eigenvalues will also be M, as can be seen by either computing the covariance matrix or the matrix of dot products. In KPCA, the kernel matrix in feature space will have instead, a $N \times N$ size and the number of non-zero eigenvalues can often be higher than M. So, it is possible to spread the information of the data in a higher number of directions, allowing the separation of noise and signal more effectively. The second point is that nonlinear principal components afford better recognition rates than corresponding numbers of linear principal components [3]. KPCA has been suggested for various signal processing tasks, e.g, denoising and compression [8, 9]. In some applications, KPCA is used as a pre-processing step before applying an SVM, [10], however in [11] a new algorithm called kernel Projection Machine is proposed and the preliminary results show that this algorithm reaches the same performance as the SVM.

In this chapter the KPCA methodology is reviewed. The KPCA and Greedy KPCA algorithms are reformulated under a unifying algebraic notation, underlying the differences between both approaches. Two methods of computing the pre-image discussed in the literature are summarized in section 4.2,Some alternatives are exposed to optimize the complexity of the pre-image problems which render the algorithms much more fast and efficient. Some experiments with sinusoidal and EEG signals are done to evaluate and to compare the pre-image methods performance in the denoising task. In section 4.3 the Greedy KPCA algorithm is explained. Different Nyström approaches (orthogonal and non-orthogonal) to compute the basis vectors in Greedy KPCA algorithm were discussed in a common algebraic framework. Numerical simulations concerning the subset selection in input space are discussed and some results are presented. The discussion is supported with some artificial and real datasets to better understand the influence of different parameters in the algorithms. The selection of the RBF parameter is discussed in section 4.4 and the centering problem is discussed in section 4.5 considering a complete training set as well as a reduced training set to compute the kernel matrix. Finally some conclusions are presented, section 4.6. Note that all the results present in this chapter are published already in [12, 13, 14, 15, 16, 17, 18].

4.1 Kernel PCA

Kernel subspace techniques are projective methods in a feature space created by a non-linear transformation of the data. The data is mapped into an high (and possible infinite) dimensional space defined by a nonlinear function. However, the mapping into feature space is avoided by using kernel functions which implicitly define a dot product in feature space computed using data in input space, known as the "kernel trick" [2]. Afterwards, every data manipulation (or every algorithm) can be efficiently computed as long as it can be translated into a sequence of dot products.

4.1.1 Main Steps of Kernel PCA

The main steps of the Kernel PCA algorithm are:

1. Mapping the data into a high dimensional space by mapping Φ and computing the kernel matrix
   - $\mathbf{\Phi} = [\phi(\mathbf{x}_1)\phi(\mathbf{x}_2)\dots\phi(\mathbf{x}_N)]$ represents the mapping.
   - $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ represents the kernel matrix, $N \times N$.
2. PCA in feature space
   - Eigendecomposition of the kernel matrix ($\mathbf{K} = \mathbf{VDV}^T$) to compute the dual form, eqn. 2.5.
   - Computing the projections in feature space ($\mathbf{Y} = \mathbf{U}^T\mathbf{\Phi}$), where $\mathbf{U}$ are the basis vectors.
3. Reconstruction in feature space
4. Reconstruction in input space - Pre-Image problem.

4.1.2 Kernel Matrices and Non-linear Projections

Kernel Principal Component Analysis (KPCA) relies on a non-linear mapping of a given data to a higher dimensional space, called feature space. In KPCA a multi-dimensional signal $\mathbf{x}_n$ with $n = 1, \dots, N$, is envisaged to be mapped through a non-linear function $\phi(\mathbf{x}_n)$ into a feature space yielding the mapped dataset. This procedure results in an increase of the dimension of the original problem. Nevertheless, the kernel subspace techniques are projective methods in feature space created by a nonlinear transformation

$$\mathbf{\Phi} = [\phi(\mathbf{x}_1)\phi(\mathbf{x}_2)\dots\phi(\mathbf{x}_N)] \tag{4.1}$$

In feature space the matrix of inner products is computed, called kernel matrix, $\mathbf{K}$

$$\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi} \tag{4.2}$$

Note that the kernel matrix has $N \times N$ dimension. The eigendecomposition of the kernel matrix is performed

$$\mathbf{K} = \mathbf{V}\mathbf{D}\mathbf{V}^T \tag{4.3}$$

The matrix $\mathbf{V}$ is formed by the eigenvectors and $\mathbf{D}$ is a diagonal matrix with the corresponding eigenvalues, both resulting from the eigendecomposition of the kernel matrix, $\mathbf{K}$. Usually the eigenvectors, i.e. the columns of $\mathbf{V}$, are in decreasing order according to the value of the corresponding eigenvalues.

$$\lambda_1 > \lambda_2 > ... > \lambda_L > ... > \lambda_N \tag{4.4}$$

The number (L) of selected eigenvectors can be chosen according to the percentage of variance of the data to be kept in the new representation. In feature space a linear PCA is then performed estimating the eigenvectors and eigenvalues of a matrix of *outer* products, called a covariance matrix, given by $\mathbf{S} = \mathbf{\Phi}\mathbf{\Phi}^T$ [13, 2]. It can be shown that these eigenvectors and eigenvalues are related to those of kernel matrix, section 2.1. In order to avoid an explicit mapping of the data, the basis vector matrix must be written in its dual form

$$\begin{aligned} \mathbf{U} &= \mathbf{\Phi}\mathbf{V}\mathbf{D}^{-1/2} \\ &= \mathbf{\Phi}\mathbf{A} \end{aligned} \tag{4.5}$$

where $\mathbf{A} = \mathbf{V}\mathbf{D}^{-1/2}$, as in eqn. 2.6. The projections of the mapped dataset $\mathbf{\Phi}$ are then obtained by computing the dot product between the mapped training set and the basis vectors $\mathbf{U}$

$$\mathbf{Y} = \mathbf{U}^T\mathbf{\Phi} \tag{4.6}$$

The columns of the matrix $\mathbf{U}$ form a basis of feature space. Assuming the $N \times L$ eigenvectors matrix $\mathbf{V}$ and the corresponding $L \times L$ eigenvalues diagonal matrix $\mathbf{D}$, the new representation of each example has dimension L. Using eqn. 4.5 and eqn. 4.6, the column vectors $\mathbf{y}_j$ of $\mathbf{Y}$ can be written as dot products of the dataset

$$\begin{aligned} \mathbf{y}_j &= \mathbf{A}^T\mathbf{\Phi}^T\phi(\mathbf{x}_j) \\ &= \mathbf{A}^T\mathbf{k}_{\mathbf{x}_j} \end{aligned} \tag{4.7}$$

Nonetheless, all data manipulation is achieved by dot products and the kernel trick is applied where $\mathbf{\Phi}^T(X)\phi(\mathbf{x}_j)$ represents a component vector which can be computed as

$$\mathbf{k}_{xj} = [k(\mathbf{x}_1, \mathbf{x}_j), k(\mathbf{x}_2, \mathbf{x}_j), ..., k(\mathbf{x}_N, \mathbf{x}_j)]^T \tag{4.8}$$

where $\mathbf{k}_{xj}$ represents a vector $N \times 1$ of the kernel matrix.
Kernel methods rely on the kernel trick, which is an efficient implementation of dot products. In feature space, dot products are evaluated by kernel functions like the radial basis function (RBF) using the data in input space. In this work a radial basis function (RBF) kernel is used, and the dot product between a vector $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ is computed using a kernel function

that only depends on the input space, according to

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) &= \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j) \\ &= exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}) \end{aligned} \tag{4.9}$$

where σ^2 is a free parameter related to the width of the kernel. The choice of the optimal value for σ and its influence in the algorithm are some issues of the KPCA algorithm that will be discussed in this chapter. By eqn. 4.9 it is very easy to compute the $N \times N$ matrix of the dot products $\mathbf{K}$. Each entry (i, j) of the kernel matrix is the result of the dot product between a pair (i, j) of examples of the training set. It is important to point out that $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d^{(2)}$ represents the euclidean distance of the dataset in input space,

$$d^{(2)} = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2x_j^T x_i \tag{4.10}$$

The last relation will be useful in the pre-image problem. In the next sections the reconstruction and pre-image steps will be discussed. Some algorithms will be exposed and new alternatives will be presented to optimize their performance.

4.1.3 Reconstruction in Feature Space

There are many applications (for instance classification) where the projections provide necessary and sufficient information to characterize the problem [19]. However, in denoising applications, for example, it is necessary to reconstruct any data point in feature space from its noisy version employing the L principal components [12]. The reconstructed point in feature space is obtained by:

$$\begin{aligned} \hat{\phi}(\mathbf{x}_j) &= \mathbf{U}\mathbf{y}_j \\ &= \mathbf{\Phi}\mathbf{V}\mathbf{D}^{-1/2}\mathbf{y}_j \\ &= \mathbf{\Phi}\mathbf{g} \end{aligned} \tag{4.11}$$

where $\mathbf{g} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{y}_j$ is a $N \times 1$ vector. If the value of L is high enough to project the data in all directions associated to the eigenvalues different from zero, then $\hat{\phi}(\mathbf{x}_j) = \phi(\mathbf{x}_j)$. In practice, the above equation is never explicitly used, but is necessary for the algebraic manipulation of the data when it intends to do the inverse mapping for the input space - Pre-Image Problem.

4.2 Pre-Image Problem

In KPCA, it is only possible to obtain an approximate pre-image for a given dataset projected on the feature space. This problem can be tackled by many approaches, some of them of iterative nature, and some with closed-form solutions. Typically those solutions make use of the fact that distances in feature space are related to distances in input space. Thus, those solutions try to achieve an optimal transformation that can embed those feature points in input space respecting those distance relationships.
As referred in [20, 21, 22, 23] the pre-image problem is useful in two principal applications:

denoising and compression. Another application of pre-image techniques is in reduced set methods in [24] and hyper-resolution using kernels methods [9]. Obviously, solving pre-image problems corresponds to an inverse problem, $\mathbf{p} = \hat{\phi}^{-1}(\mathbf{x}_j)$ because solving a pre-image problem requires the inversion of the feature map. Unfortunately, in the most common cases it turns out that the problem of inverting Φ belongs to the class of ill-posed problems. Since such a point $\mathbf{x}$ might not exist, the pre-image is ill posed. A way to solve this problem is to look for an approximate pre-image, i.e. a point $\mathbf{p}$ such that $\phi(\mathbf{p})$ is close as possible to $\hat{\phi}(\mathbf{x})$. Beyond the existence, uniqueness is another problem in computing the pre-image. Exact solutions rarely exist. In order to avoid working with the mapped dataset $\mathbf{\Phi}$, pre-image estimation methods described in the literature combine the reconstruction in feature space and the estimation of its pre-image in input space in one step. This goal is achieved by using the Euclidean or $L2$ - norm. The square of the Euclidean distance can be written using dot products which in turn can be substituted by kernel evaluations. Considering a point $\mathbf{p}$ in input space, the distance of its image $\phi(\mathbf{p})$ in feature space to the reconstructed point $\hat{\phi}(\mathbf{x}_j)$ is defined by

$$\begin{aligned} \tilde{d}^{(2)} &= \|\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}_j)\|^2 \\ &= (\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}_j))^T(\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}_j)) \end{aligned} \tag{4.12}$$

where $\tilde{d}^{(2)}$ represents the distance in feature space. Substituting $\hat{\phi}(\mathbf{x}_j)$ in the last equation by the expression given in eqn. 4.11, the dot product can be replaced by kernel values

$$\tilde{d}^{(2)} = k(\mathbf{p}, \mathbf{p}) - 2\mathbf{g}^T\mathbf{k}_p + \mathbf{g}^T\mathbf{K}\mathbf{g} \tag{4.13}$$

where $\mathbf{k}_p$ represents a vector whose entries are computed as the dot product of $\phi(\mathbf{p})$ with images $\mathbf{\Phi}$ of the set of training data $\mathbf{x}_n$, $n = 1, ..., N$ according to eqn. 4.8 identifying $\mathbf{p} \equiv \mathbf{x}_j$. Note that the term $\mathbf{g}^T\mathbf{K}\mathbf{g}$ is independent of $\mathbf{p}$.
Two methods were studied to compute the pre-image based upon the definition of Euclidean distance within different strategies. Consequently the input space point $\mathbf{p}$ must be chosen accordingly to them.
In the following sections these different strategies will be discussed to solve the approximation pre-image problem and new proposals are going to be made to optimize the algorithms.

4.2.1 Distance Method

Recent works [20] to estimate the pre-image of a given point in feature space are based on the fact that it is possible to compute the coordinates of a new point if its distances to a set of known points are known [25]. Hence, the distance vector of the reconstructed point $\hat{\phi}(\mathbf{p})$ to all mapped points of the training set $\mathbf{x}_n,\ n = 1, \ldots, N$ is obtained by

$$\tilde{\mathbf{d}}^{(2)} = diag(\mathbf{K}) - 2\mathbf{g}^T\mathbf{K} + \mathbf{g}^T\mathbf{K}\mathbf{g} \tag{4.14}$$

where $\tilde{\mathbf{d}}^{(2)}$ is a vector of distances in feature space and

$$diag(\mathbf{K}) = [\phi^T(\mathbf{x}_1)\phi(\mathbf{x}_1) \ \ \phi^T(\mathbf{x}_2)\phi(\mathbf{x}_2) \ \ \ldots \phi^T(\mathbf{x}_N)\phi(\mathbf{x}_N)] \tag{4.15}$$

represents the diagonal of the kernel matrix. With certain kernels it is possible to evaluate the distance in input space knowing the corresponding distance in feature space. If, for example, a RBF kernel is considered, there is a distinct relation between an input space distance $\mathbf{d}^{(2)}$ and the corresponding feature space distance [26, 20]. Assuming that $k(\mathbf{x}_i, \mathbf{x}_j)$ is an isotropic kernel, the vector of distances in feature space can be computed as

$$\begin{aligned} \tilde{\mathbf{d}}^{(2)} &= diag(\mathbf{K}) + k(\mathbf{p},\mathbf{p})\mathbf{j}_N - 2exp(-\tfrac{\mathbf{d}^{(2)}}{2\sigma^2}) \\ &= 2\mathbf{j}_N - 2exp(-\tfrac{\mathbf{d}^{(2)}}{2\sigma^2}) \end{aligned} \tag{4.16}$$

where $\mathbf{j}_N = [1, 1, ..., 1]$ is N dimensional vector, $k(\mathbf{x}_j, \mathbf{x}_j) = 1$, $diag(\mathbf{K}) = \mathbf{j}_N$ and $exp(-\frac{\mathbf{d}^{(2)}}{2\sigma^2})$ represents the RBF function, eqn. 4.9. The corresponding vector of distances in input space is then given by

$$\mathbf{d}^{(2)} = -2\sigma^2 \ln(\mathbf{j}_N - \frac{1}{2}\tilde{\mathbf{d}}^{(2)}) \tag{4.17}$$

Following this, a subset $\mathcal{S}$ of neighbors of the reconstructed point $\hat{\phi}(\mathbf{x}_j)$ is considered, i.e. those S points from the training set with the smallest distance $\tilde{\mathbf{d}}^{(2)}$ are chosen, and the corresponding points $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_S]$ in the input space are selected. After centering the set of neighbors a SVD is computed

$$\begin{aligned} \mathbf{Q}_c &= \mathbf{Q}(\mathbf{I} - \tfrac{1}{S}\mathbf{j}_S^T\mathbf{j}_S) \\ &= \mathbf{E}\mathbf{\Sigma}\mathbf{F}^T \\ &= \mathbf{E}\mathbf{W} \end{aligned} \tag{4.18}$$

where $\mathbf{j}_S = [111...1]$ is a S dimensional vector and $\mathbf{I}$ is a identity matrix $M \times S$. The columns of $\mathbf{W} = \mathbf{E}^T\mathbf{Q}_c$ represent the new coordinates of the points $\mathbf{Q}_c$. Their distance to the origin, is obtained as $\mathbf{d}_0^{(2)} = [\| \mathbf{w}_1 \|^2, \| \mathbf{w}_2 \|^2, \ldots, \| \mathbf{w}_S \|^2]$.
It is assumed that the image $\mathbf{p}$ is contained in the space of the considered neighbors. Usually, the pre-image does not exist in input space and, therefore, the solutions that meet these conditions do not exist. In [25] a method to calculate the $\mathbf{p}$ value is described. Then, the coordinates of the point $\tilde{\mathbf{p}}$ (a pre-image that one searches in the new space of dimension $M \times S$) are given by

$$\mathbf{W}^T\tilde{\mathbf{p}} = -\frac{1}{2}(\mathbf{d}^{(2)} - \mathbf{d}_0^{(2)}) \tag{4.19}$$

Manipulating the last equation, the pre-image $\mathbf{p}$ of the reconstructed point $\hat{\phi}(\mathbf{x}_j)$ is finally obtained as

$$\begin{aligned} \mathbf{p} &= \mathbf{E}\tilde{\mathbf{p}} + \tfrac{1}{S}\mathbf{Q}\mathbf{j}_S \\ &= \mathbf{E}\tilde{\mathbf{p}} + \mathbf{p}_0 \end{aligned} \tag{4.20}$$

where $\mathbf{p}_0$ represents the mean of the selected neighbors. This method is usually applied considering that the number S of neighbors is less than the dimension M of the points in input space [20]. In that case $M - S$ components of the point $\tilde{\mathbf{p}}$ vanish. Hence, the covariance matrix of the set of points $\mathbf{Q}$ can have at most S non-zero eigenvalues. Also note that the SVD of eqn. 4.19 represents the minimum norm solution. In that case, the second term of eqn. 4.20 representing the mean of the neighbors, might constitute the dominant term for

solving the pre-image estimation of the reconstructed point $\hat{\phi}(\mathbf{x}_j)$. So, the new alternative to find the pre-image $\mathbf{p}$ is

$$\begin{aligned} \mathbf{p} &= \tfrac{1}{S}\mathbf{Q}\mathbf{j}_S \\ &= \mathbf{p}_0 \end{aligned} \tag{4.21}$$

using the mean of the neighbors subset.

4.2.2 Fixed-point Method

The central idea of the fixed-point method [22] consists in computing the unknown pre-image $\mathbf{p}$ which minimizes the Euclidean distance in feature space by setting to zero the gradient of eqn. 4.13

$$\frac{\partial \tilde{d}^{(2)}}{\partial \mathbf{p}} = \frac{\partial k(\mathbf{p},\mathbf{p})}{\partial \mathbf{p}} - 2\frac{\partial \mathbf{g}^T\mathbf{k}_p}{\partial \mathbf{p}} \tag{4.22}$$

Substituting the RBF kernel function, the first term of the previous equation is zero because $k(\mathbf{p},\mathbf{p}) = 1$. Hence the zeros of the gradient are obtained by

$$\begin{aligned} \sum_{i=1}^{N} g_i(\mathbf{x}_i - \mathbf{p}) \exp\left(\frac{\|\mathbf{x}_i - \mathbf{p}\|^2}{\sigma^2}\right) &= \\ \mathbf{X}(\mathbf{g}\lozenge\mathbf{k}_p) - \mathbf{p}\mathbf{g}^T\mathbf{k}_p &= 0 \end{aligned} \tag{4.23}$$

where $\lozenge$ represents the Hadamard product. The zeroes can thus be computed iteratively by the fixed-point algorithm

$$\mathbf{p}_{t+1} = \frac{\mathbf{X}(\mathbf{g}\lozenge\mathbf{k}_{\mathbf{p_t}})}{\mathbf{g}^T\mathbf{k}_{\mathbf{p_t}}} \tag{4.24}$$

The iterative procedure stops when $|\mathbf{p}_{t+1} - \mathbf{p}_t|$ is less than a threshold and/or a maximum number of iterations t is achieved. The starting point $\mathbf{p}_0$ can be chosen randomly but often leads to a slow convergence. When SNR is high, an option is to start with the noisy point [27]. However and since the denominator is the dot product between the denoised point $\hat{\phi}(\mathbf{x}_j)$ and $\phi(\mathbf{p}_t)$, the starting point can be chosen using the nearest neighborhood strategy. That way the numerical instability of having a very small or negative denominator on the first iteration can be avoided once $\mathbf{p}_0$ is chosen according to the maximum dot product criterion. This strategy is more efficient than starting with the value computed by the distance method, eqn. 4.20, as suggested in [9]. Note that with RBF kernel, the identification of the closest neighbors can be achieved with the dot products thus avoiding the computation of Euclidean distances in feature space. Computing the vector $\mathbf{r}$ of dot products of $\hat{\phi}(\mathbf{x}_j)$ with the training set $\mathbf{\Phi}$ yields

$$\mathbf{r} = \mathbf{g}^T\mathbf{K} \tag{4.25}$$

As the dot product of every mapped data point with itself is normalized to one, the closest neighbors are obtained by identifying the set S of maximal dot products $(\hat{\phi}^T(\mathbf{x}_j)\phi(\mathbf{x}_i)), i = 1,\ldots,N$. The S closest neighbors, i.e. the ones that exhibit the largest dot products, are chosen. Selecting the corresponding points $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_S]$ in input space, the fixed-point

iteration should start with

$$\mathbf{p}_0 = \frac{1}{S}\mathbf{Q}\mathbf{j}_S \tag{4.26}$$

In the next section some results with artificial and non-artificial signals will be shown to address the pre-image problem using different approaches.

4.2.3 Evaluation Results

In this section the pre-image methods for denoised one-dimensional signals was studied. The aim was to see which method described before is more effective in the pre-image problem. The distance and fixed-point methods were used as well as the variants of these methods proposed before. To estimate the pre-image of every denoised point in feature space, the methods need a set of neighbors of a denoised point to be selected from the training set. Pre-images were estimated varying the number of neighbors S and using:

- **Distance**: the distance method as proposed by [20], eqn. 4.20.
- **Mean**: the mean of the nearest neighbors only within the distance method, which corresponds to considering $\mathbf{p}_0$ as the pre-image, eqn. 4.21.
- **FPM**: the fixed-point method, eqn. 4.24, initialized with $\mathbf{p}_0$, eqn. 4.26.
- **FPR**: the fixed-point method initialized with a random point.

In the following subsections the pre-image methods and their variants discussed above will be first applied to denoise a sinusoidal time series in $3D$ in order to illustrate the denoising performance of the modified KPCA in what concerns the estimation of pre-images. The second example refers to extraction of an EOG artifact from single channel EEG recordings. These methods were also applied to USPS dataset as presented in [12, 13].

Sinusoidal Signal

The KPCA was applied to a noisy 3D signal, appendix A.1.1. The reconstruction in feature space was achieved using $L = 1$ principal components. The mean-square error was computed between the denoised sequence and the original sequence. Table 4.1 shows the mean-square error between the reconstructed and the original signals for the two methods and their variants of estimating the pre-image. By the present results, given an embedding dimension $M = 3$, the number of neighbors S has a direct influence on the estimation methods. The strongest impact is seen in the distance method. Fig. 4.1 (a) and (c) illustrate the results in the multidimensional space using the mean only, while figure 4.1 (b) and (d) show results obtained using the distance method, eqn. 4.20. Obviously the latter yields no significant improvement. Note that in [20], S was chosen always less than M. Further, considering the situation $S = 5 > M = 3$, the results obtained using the mean $\mathbf{p}_0$ are now significantly better than the results using $\mathbf{p}$ from eqn. 4.20. The latter result is still noisy and shows too many outliers, figure 4.1 (d). In this example, the pre-image mean variant is a strategy to find the best match of $\phi(\hat{\mathbf{x}}_j)$ in the training set based on single neighbor ($S = 1$) or by doing a kind of

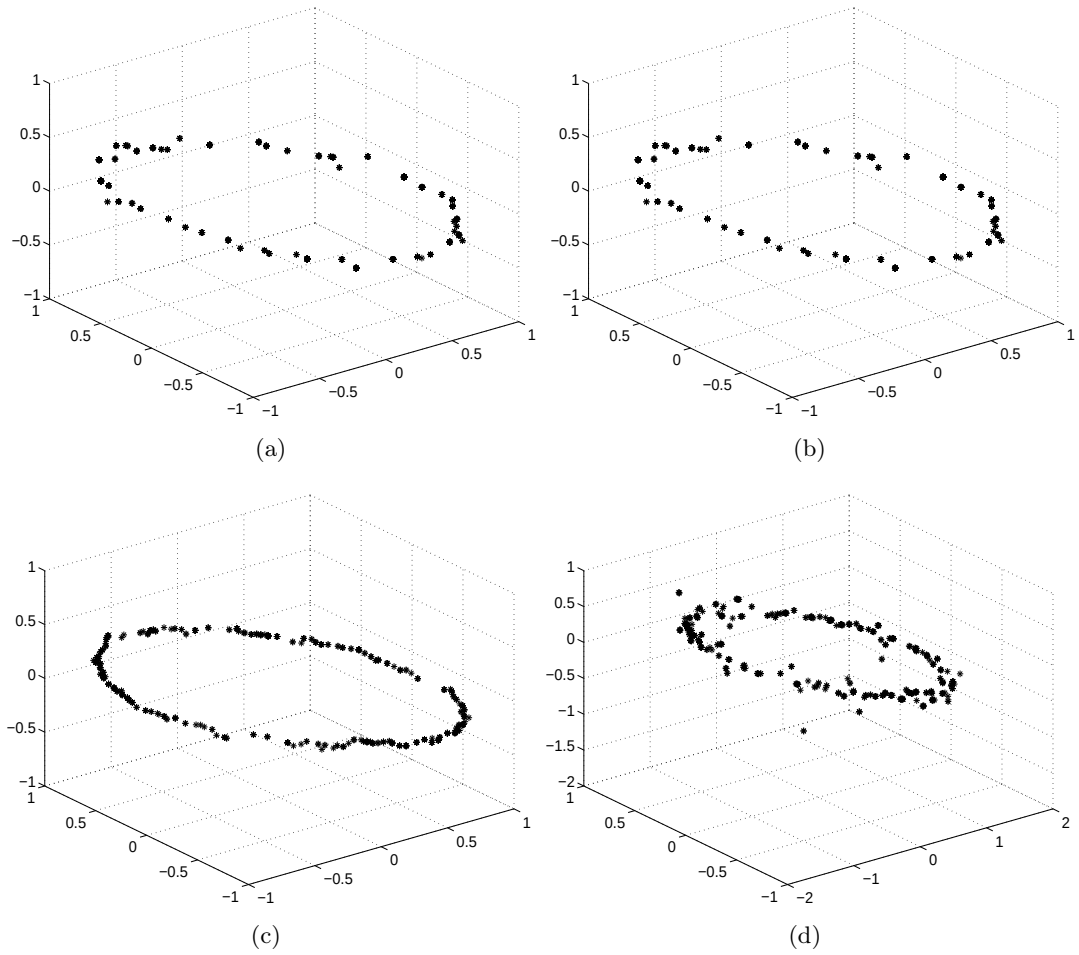


Figure 4.1: Distance Method Analysis. (a) Using nearest neighbor $S = 1$; (b) Distance method with $S = 1$; (c) Mean with $S = 5$; (d) Distance method with $S = 5$.

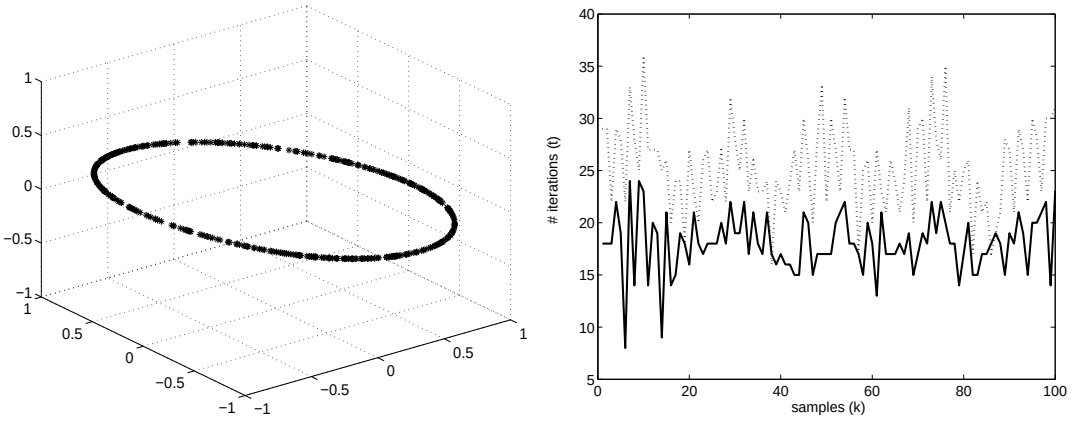
adaptive clustering ($S > 1$) in feature space. The denoised point in input space is obtained as the mean of points in input space which correspond to the nearest neighbors in feature space. Note that MSE decreases with an increasing S (second column of table 4.1). The iterative fixed-point algorithm exhibits a more robust performance as it is not dependent on the starting point (two last column of table 4.1). Also notice that the underlying trajectory of the signal is smoother than what is obtained by the mean method (compare figure 4.2 (a) and figure 4.1 (a) and (c)). The initialization of fixed-point algorithm with the mean of the neighbors, turns the algorithm faster (figure 4.2 (b)) and avoids very low values in the denominator of the eqn. 4.24.

EEG Data

A frontal (Fp1-Cz) EEG channel of $12s$ of duration containing high-amplitude EOG artifacts was considered and divided into 4 sub-segments with $K = 384$ samples. KPCA was applied separately to each sub-segment. The one-dimensional signal was embedded in $M = 11$ dimensions, but only $L = 4$ principal components were used for the reconstruction in the feature

S	**Distance**	**Mean**	**FPM**	**FPR**
1	0.0843	0.0843	0.0843	0.0843
2	0.0813	0.0800	0.0843	0.0843
3	0.0813	0.0789	0.0843	0.0843
4	0.1138	0.0771	0.0843	0.0843
5	0.5381	0.0753	0.0843	0.0843

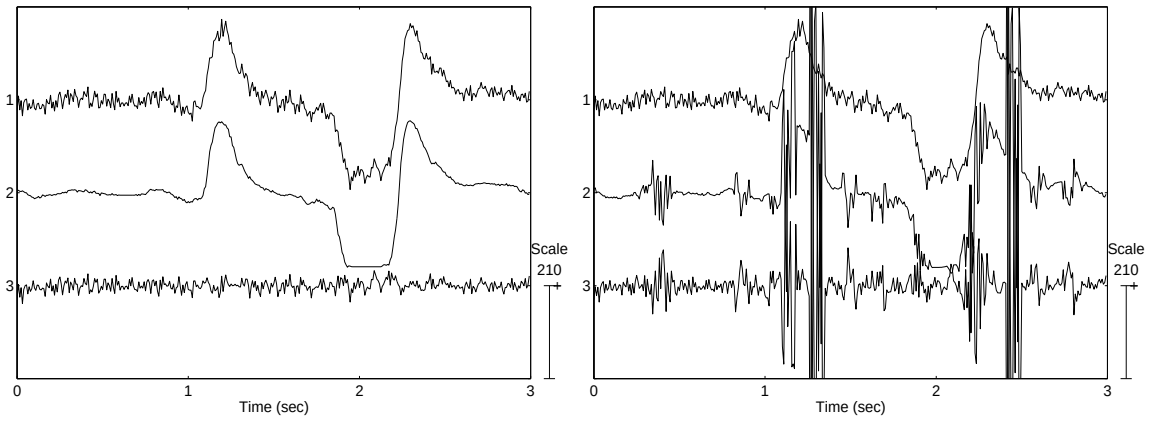
Table 4.1: Mean square error: original versus the denoised signal with KPCA using different algorithms to compute pre-image.



(a) (b)

Figure 4.2: Fixed-point method: using nearest neighbor $S = 1$ as starting (a); number of iterations using random FPR (dotted line) or mean of neighbors initialization FPM (full line) (b).

space. Visual inspection of the extracted signals confirmed that the results strongly depend



(a) Fixed-Point method ($S = 12$) (b) Distance method ($S = 12$)

Figure 4.3: Segment of signal processed by KPCA using either (a) the fixed-point or (b) the distance method to estimate the pre-image (*top*: the original EEG, *middle*: the extracted EOG signal, *bottom*: the corrected EEG).

on the method to estimate the pre-image corroborating results obtained by the sinusoidal example. Further experiments showed that the performance of the distance method is strongly dependent on the number S of neighbors yielding, in some cases, unreliable solutions, figure 4.3 (b). The fixed-point algorithm, on the contrary, leads to more stable solutions whatever the number of neighbors used as a starting point, figure 4.3 (a). To provide a global overview of the performance of the methods, the correlation coefficients between a reference signal and signals resulting from changing, either the method of estimating the pre-image or varying the number of neighbors S, are calculated. Figure 4.4 shows the results considering as reference

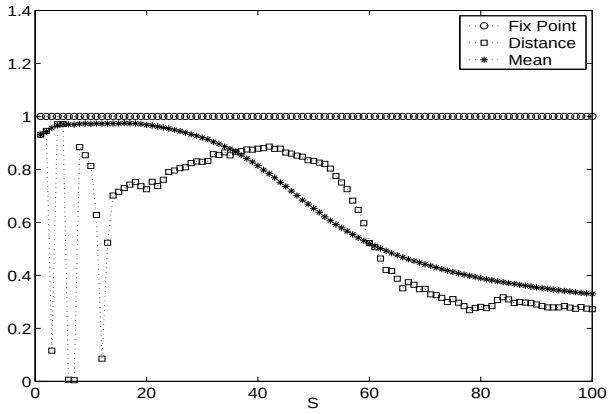


Figure 4.4: Correlation coefficient between a reference signal and all the signals resulting from changing the pre-image method and/or varying S.

the signal obtained with the fixed-point method initialized with the best match in the training set ($S = 1$). Note that with $S = \{3, 6, 7, 12\}$ neighbors, the distance method does not yield a reliable solution, figure 4.3 (b) but also if $S >> M$ the correlation coefficients are low. If $S < 20$, the result of the mean is very close to the fixed-point algorithm.
The result of the fixed-point initialized with the mean of neighbors, FPM, is shown in figure 4.5. Usually, the number of iterations decreases by approximately a factor of two if the starting point is the mean of neighbors instead of a random choice. To compare the FPM and the Mean performance in EEG signal, the power spectral density for the extracted and corrected EEGs using $S = 12$ was done. Figure 4.6 shows that the power spectral density is identical in the corrected EEG signals while the extracted EOG shows no visible difference between the two methods. The same analysis was done to denoise USPS dataset. The results with the USPS dataset confirm the results obtained with other datasets [12]. The distance method is the most sensitive concerning the number of nearest neighbors selected. With the fixed-point method $S = 1$, a reliable solution is achieved and it was verified that the algorithm converges faster than when it is initialized randomly, as shown by figure 4.5.

4.3 Greedy KPCA

Computing the eigendecomposition of the kernel matrix in huge datasets, can be prohibitive as it involves a large matrix ($N \times N$). The size of the kernel matrix represents a computational

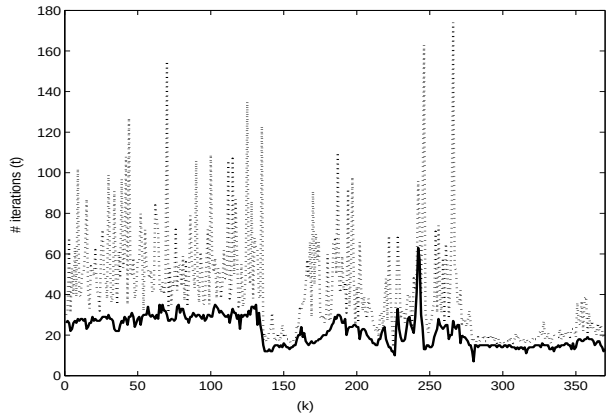


Figure 4.5: The number of iterations needed to denoise the EEG segment using the algorithms FPR (dotted line) and FPM (full line)

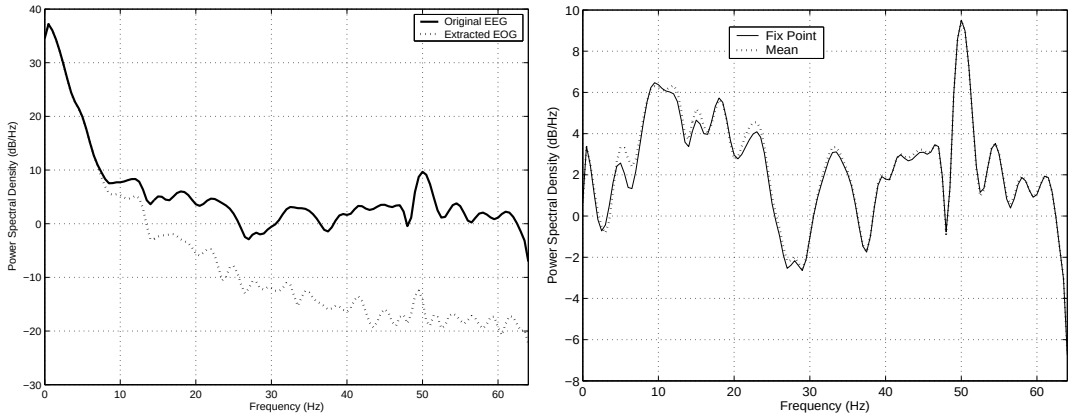


Figure 4.6: Power Spectral density of *left*: the original EEG and extracted EOG by FPM and Mean algorithm, *right*: the corrected EEG comparing the Fixed-point vs Mean algorithm ($S = 12$).

burden. Nevertheless, only the largest eigenvalues and corresponding eigenvectors need to be computed. In practice, the goal of projective subspace techniques is to describe the data with reduced dimensionality by extracting meaningful components while still retaining the structure of the raw data. Then, only the projections on the directions corresponding to the most significant eigenvalues of the kernel need to be computed. Therefore, apart the σ parameter, the drawbacks of Kernel PCA methods are:

- the size of kernel matrix $\mathbf{K}$. The eigendecomposition of large matrices can be unfeasible in practical applications requiring the manipulation of large datasets.
- the dual form of the model, eqn. 4.5. In classification problems this form needs the storage of the training set, even during the test phase, to compute the projections of any new point $\phi(\mathbf{x_j})$ into the model. This is due to the fact that the mapping is never explicitly computed but simultaneously obtained with dot product - the so called kernel trick.

In this section a method is suggested to deal with both problems - Greedy KPCA. The kernel matrix of the complete training set is not computed and the eigendecomposition is performed with matrices of smaller size ($R < N$). The description of the model is also based on a subset of the training dataset.

The exploitation of the methods to achieve a low rank eigendecomposition is a strategy that has been considered in different applications. Greedy approaches have been applied in several fast algorithms to approximate the kernel matrix. In [28] the kernel matrix $\mathbf{K}$ is approximated by the subspace spanned across a subset of columns. The basis vectors are chosen incrementally to minimize an upper bound of the approximation error of the kernel matrix. In [29, 6] the greedy approach for low rank approximation based in the incomplete Cholesky decomposition is proposed. Another greedy sampling scheme is mentioned in [30] based on how well a sample point can be represented by a (constrained) linear combination of the current subspace basis in feature space. The exploitation of methods like Nyström to achieve the low rank eigendecomposition is another class of low rank approximation algorithms considered in [31, 2]. In [32] the Nyström method was developed to approximate the solutions of integral equations. The most popular sampling scheme for Nyström method is random sampling proposed in [33] as well as a fast version proposed in [31]. In [34, 35] the method used to choose the landmark points is not based on greedy or in probabilistic sampling as in [36], but based instead on k-means cluster centers. This proposed alternative was compared to the greedy approach and the results show that k-means is consistently better than all known variants of Nyström [**?**]. Those techniques achieve a solution without the manipulation of the full matrix.

In the next sections it will be shown how the Nyström method can be applied to KPCA leading to what is usually known as Greedy KPCA. Two alternatives to compute the basis vectors based on orthogonal and non-orthogonal approach will be exposed. A new alternative to select the pivots will be explained for the incomplete Cholesky decomposition and some results will be exposed and discussed.

4.3.1 Nyström Approach

Without loosing generality, let's assume that the dataset is centered and splitted into two parts yielding the mapped dataset

$$\begin{aligned} \mathbf{\Phi} &= [\phi(\mathbf{x}_1)\ \phi(\mathbf{x}_2)\ \ldots\ \phi(\mathbf{x}_R)\ \phi(\mathbf{x}_{R+1})\ \ldots\ \phi(\mathbf{x}_N)] \\ &= [\mathbf{\Phi}_R\ \mathbf{\Phi}_S] \end{aligned} \tag{4.27}$$

where the first R elements constitute the subset $\mathbf{\Phi}_R$ and the remaining $N - R$ elements form the subset $\mathbf{\Phi}_S$. The kernel matrix can be written in block notation [38, 31],

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \\ \mathbf{K}_{rs}^T & \mathbf{K}_s \end{bmatrix} \tag{4.28}$$

where the $\mathbf{K}_r$ is the kernel matrix within the subset $\mathbf{\Phi}_R$, $\mathbf{K}_{rs}$ is the kernel matrix between subset $\mathbf{\Phi}_R$ and $\mathbf{\Phi}_S$, $\mathbf{K}_s$ is the kernel matrix within the subset $\mathbf{\Phi}_S$. In Greedy KPCA a low-

rank approximation of the kernel matrix is considered. This leads to the eigendecomposition of matrices with reduced size. The low-rank approximation is written using the upper blocks of matrices $\mathbf{K}_r$ and $\mathbf{K}_s$ of the original matrix [38, 39]

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_r \\ \mathbf{K}_{rs}^T \end{bmatrix} \mathbf{K}_r^{-1} \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix} \tag{4.29}$$

It can be verified that the lower block is approximated by

$$\mathbf{K}_s \approx \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs} \tag{4.30}$$

If the rank of $\mathbf{K}$ is higher than R or the first R rows are not linear independent, the quality of the approximation can be quantified by the norm of the Schur complement

$$Sc = \|\mathbf{K}_s - \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs}\| \tag{4.31}$$

In a recent work [36] the advantage of using the pseudo-inverse ($\mathbf{K}_r\dagger$) in eqn. 4.29 is discussed. The Nyström extension for the R eigenvectors $\mathbf{V}$ corresponding to the R eigenvalues are obtained by

$$\mathbf{V} = \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix}^T \mathbf{H} \tag{4.32}$$

Matrix $\mathbf{H}$ is computed using eigendecomposition of $R \times R$ matrices, where R is the size of subset $\mathbf{\Phi}_R$. Different approaches were considered to form the $R \times R$ matrices. In [38] only the block $\mathbf{K}_r$ is considered, while in [6] a matrix related to both upper blocks of the kernel matrix is additionally computed $\mathbf{K}_{rs}$. The main difference between both approaches is that eigenvectors are non-orthogonal [38] or orthogonal [6]. Consequently, the low rank approximation of the kernel matrix has distinct characteristics. The strategies are described in the following sections.

Non-orthogonal Approach

In this case, a non-orthogonal matrix $\mathbf{V}$ is computed using the eigendecomposition of the upper left block of the kernel matrix $\mathbf{K}_r = \mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T$ [38]. Considering that the eigenvalues of $\mathbf{K}$ and $\mathbf{K}_r$ are related by a common scale factor (N/R), matrix $\mathbf{H}$ is

$$\mathbf{H} = \mathbf{V}_r \mathbf{D}^{-1} \tag{4.33}$$

where $\mathbf{V}$ is a non-orthogonal matrix obtained by

$$\mathbf{V} = \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix}^T \mathbf{V}_r \mathbf{D}^{-1} \tag{4.34}$$

Manipulating eqn. 4.34, eqn. 4.5 and eqn. 4.6 the basis vector matrix is

$$\mathbf{U} = \mathbf{\Phi}_R \mathbf{V}_r \mathbf{D}^{-1/2} \tag{4.35}$$

Orthogonal Approach

The alternative approach considers the kernel matrix decomposed as $\tilde{\mathbf{K}} = \mathbf{C}^T\mathbf{C}$, where $\mathbf{C}$ has dimension $R \times N$ and is computed as follow

$$\mathbf{C} = \begin{bmatrix} \mathbf{L} & \mathbf{L}^{-T}\mathbf{K}_{rs} \end{bmatrix} \tag{4.36}$$

where $\mathbf{L}$ can be computed using the Cholesky decomposition [6, 40] or the square root [31] of $\mathbf{K}_r$

- $\mathbf{K}_r = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L}$ is a triangular matrix. Note that if the matrix is symmetric positive definite, there exists an unique $R \times R$ triangular matrix that accomplishes the decomposition without any pivoting scheme. Alternatively, an incomplete Cholesky decomposition of the full matrix $\mathbf{K}$ can be performed [6].
- $\mathbf{L} = \mathbf{K}_r^{1/2} = \mathbf{V}_r\mathbf{D}_r^{1/2}\mathbf{V}_r^T$, which is a symmetric matrix.

The low rank approximation of $\tilde{\mathbf{K}} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ is based on the eigendecomposition of an $R \times R$ matrix defined by

$$\begin{aligned} \mathbf{Q} &= \mathbf{C}\mathbf{C}^T \\ &= \mathbf{V}_q\mathbf{D}\mathbf{V}_q^T \end{aligned} \tag{4.37}$$

The result of this eigendecomposition as well as the decomposition of $\mathbf{K}_r$ leads to

$$\mathbf{H} = \mathbf{L}^{-1}\mathbf{V}_q\mathbf{D}^{-1/2} \tag{4.38}$$

It has to be noticed that the matrix $\mathbf{K}_{rs}$ contributes to $\mathbf{Q}$ and not only to the upper left block, as in [38]. The eigenvectors of $\tilde{\mathbf{K}}$ are computed as

$$\mathbf{V} = \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix}^T \mathbf{L}^{-T}\mathbf{V}_q\mathbf{D}^{-1/2} \tag{4.39}$$

It can be verified that the previous eigenvectors are orthogonal ($\mathbf{V}^T\mathbf{V} = \mathbf{I}$) while the former eqn. 4.34 is not. By manipulation of eqn. 4.39, eqn. 4.5 and eqn. 4.6 the basis vector matrix is

$$\mathbf{U} = \mathbf{\Phi}_R\mathbf{L}^{-1}\mathbf{V}_q \tag{4.40}$$

Both approaches imply that the data is projected into the space spanned by the subset $\mathbf{\Phi}_R$. Given the eigenvectors (and corresponding eigenvalues) of the kernel matrix, the projections are computed by

- Using the orthogonal eigenvectors eqn. 4.39

$$\begin{aligned} \mathbf{Y} &= \mathbf{D}^{1/2}\mathbf{V}^T \\ &= \mathbf{V}_q^T\mathbf{L}^{-1}\mathbf{\Phi}_R^T \begin{bmatrix} \mathbf{\Phi}_R & \mathbf{\Phi}_S \end{bmatrix} \end{aligned} \tag{4.41}$$

- Using eqn. 4.34 and considering that the eigenvalues of $\mathbf{K}_r$ and $\mathbf{K}$ are related also by a

scaling factor (R/N), the projections are

$$\mathbf{Y} = \mathbf{D}_r^{-1/2}\mathbf{V}_r^T\mathbf{\Phi}_R^T \left[\begin{array}{cc} \mathbf{\Phi}_R & \mathbf{\Phi}_S \end{array} \right] \tag{4.42}$$

The first terms of the previous equations suggest that the data is projected into the space spanned by the subset $\mathbf{\Phi}_R$ yielding to

$$\begin{array}{rcl} \mathbf{Y} & = & \mathbf{U}^T\mathbf{\Phi} \\ & = & (\mathbf{\Phi}_R\mathbf{A})^T\mathbf{\Phi} \end{array} \tag{4.43}$$

Each column of the matrix $\mathbf{U}$ is computed as a linear combination of the subset $\mathbf{\Phi}_R$, where $\mathbf{A}$ is a matrix of the linear coefficients by

$$\mathbf{u}_i = \sum_{j=1}^{R} \mathbf{a}_{ij}\mathbf{\Phi}_R(\mathbf{x}_j), i = 1 \ldots l < R \tag{4.44}$$

where the vector $\mathbf{a}_i$ can be calculated by two ways:

- $\mathbf{a}_i = \lambda_i^{-1/2}\mathbf{v}_{ri}$. The pair $(\mathbf{v}_{ri}, \lambda_i)$ is respectively the eigenvector and eigenvalues of $\mathbf{K}_r$, considering that $\lambda_1 > \lambda_2 > ... > \lambda_i > ... > \lambda_R$, eqn. 4.35.
- $\mathbf{a}_i = \mathbf{L}^{-T}\mathbf{v}_{qi}$. The vector $\mathbf{v}_{qi}$ is the i $(< R)$ eigenvector of the matrix $\mathbf{Q}$ associated to the i eigenvalue (ordered by decreasing value), eqn. 4.40. It has to be noticed that

  $$(\mathbf{\Phi}_R\mathbf{L}^{-T})^T(\mathbf{\Phi}_R\mathbf{L}^{-T}) = \mathbf{I} \tag{4.45}$$

  which corresponds to an orthogonalization of the subset $\mathbf{\Phi}_R$ [41].

4.3.2 Splitting the Dataset

In the previous section, it was assumed that the training set is splitted into two groups. In what concerns the Nyström approach it is said that the first R rows should represent the linear independent rows of the kernel matrix. Usually, R rows randomly chosen are used to organize the upper block of the kernel matrix. This strategy is also suggested by the majority of published works [38, 42, 31] for huge datasets considering that there is an high probability of the randomly chosen subset still might represent the training set distribution. However, the quality of the approximation is ruled by the norm of Schur's complement, eqn. 4.31. There are two more strategies besides the random strategy to select the subset R of the training set:

1. First, the incomplete Cholesky decomposition (Appendix A.3) can be applied to the full kernel matrix using symmetric pivoting, as described in [6], [43]. Naturally R must be chosen a-priori, but in [29] it is suggested that the values of the pivoting could be used as a stop condition. Using this strategy to select the subset R having as input the kernel matrix, can be a disadvantage if a huge dataset is involved. A very efficient implementation for the incomplete Cholesky decomposition algorithm exists, accessible in [44], having as input: the training dataset $\mathbf{X}$, σ of the RBF kernel and a threshold to

control the approximation error of the decomposition. As described in [6], the matrix $\mathbf{C}$ is formed iteratively, starting with one row up to R when the error is less than the threshold. The outputs of the algorithm are the index of the pivoting scheme and matrix $\mathbf{C}$. The former allows the identification of the subset Φ_R which will contribute to form R orthogonal basis vectors. The total error ϵ is approximated as [44]

$$\epsilon \approx tr(\mathbf{K}_s - \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs}) \tag{4.46}$$

A similar approximation error was defined in other algorithms, [5, 45, 46], to choose the subset of the training set.

2. Second, the Naïve algorithm is used. This implementation considers the global dataset $\mathbf{\Phi}$ and defines iterative procedures to form the subset $\mathbf{\Phi}_R$ [45, 47]. This is based on the assumption that every element of subset $\mathbf{\Phi}_S$, $\phi(\mathbf{x}_s)$ can be described as a linear combination of the elements of $\mathbf{\Phi}_R$ [45, 47].

$$\hat{\phi}(\mathbf{x}_s) = \mathbf{\Phi}_R \mathbf{w}_s \tag{4.47}$$

where $\mathbf{w}_s$ represents a vector of the linear coefficients. The square error of this representation is

$$\begin{aligned} \varepsilon_{x_s} &= \|\phi(\mathbf{x}_s) - \hat{\phi}(\mathbf{x}_s)\|^2 \\ &= \|\phi(\mathbf{x}_s) - \mathbf{\Phi}_R \mathbf{w}_s\|^2 \\ &= k(\mathbf{x}_s, \mathbf{x}_s) - 2\mathbf{w}_s^T \mathbf{k}(\mathbf{X}_R, \mathbf{x}_s) + \mathbf{w}_s^T \mathbf{K}_r \mathbf{w}_s \end{aligned} \tag{4.48}$$

where $\mathbf{k}(\mathbf{X}_R, \mathbf{x}_s)$ represents a vector of dot products between all points Φ_R and the point $\phi(\mathbf{x}_s)$ mapped using the kernel function. The zeroes of the gradient of the error allows the computation of the $\mathbf{w}_s$

$$\begin{aligned} \frac{\partial \varepsilon_{x_s}}{\mathbf{w}_s} &= 0 \\ \mathbf{w}_s &= \mathbf{K}_r^{-1} \mathbf{k}(\mathbf{X}_R, \mathbf{x}_s) \end{aligned} \tag{4.49}$$

Based on the last equation, an iterative algorithm to form the subset $\mathbf{\Phi}_R$ is proposed. Starting by one element (randomly selected in the training set) authors propose iterative solutions. The steps of [5] are summarized:

   (a) Compute every element of $\mathbf{\Phi}_S$ as a linear combination of the actual $\mathbf{\Phi}_R$. For each element of $\mathbf{\Phi}_S$ the optimal coefficient values are given by eqn. 4.49.

   (b) Compute the errors ε_{x_s} of the new representations of all elements of $\mathbf{\Phi}_S$ eqn. 4.48.

   (c) Identify the element $\phi(\mathbf{x}_s)$ that achieves the larger error.

   (d) Move the element $\phi(\mathbf{x}_s)$ of $\mathbf{\Phi}_S$ to the set $\mathbf{\Phi}_R$.

The two strategies described before are related. The first is based on the total error, eqn. 4.46, while the other is based on the error for each point $\phi(\mathbf{x}_s)$. Substituting eqn. 4.49 in eqn. 4.47 it is possible to deduct that the two strategies to select the subset R are the same

and the errors of eqn. 4.48 and eqn. 4.46 are related by

$$\varepsilon = \sum \varepsilon_{x_s} \quad \forall x_s \in S \tag{4.50}$$

Note that the iterative algorithms described before to find a subset R do not need the kernel matrix as input. The two ways described above to select the subset R of the training set are repeated up to the achievement of some criteria:

- Size of set $\mathbf{\Phi}_R$ - in [15] an error threshold to stop the Cholesky decomposition was the maximum number of pivots.
- Trace of the matrix - The error ϵ is approximated as $\epsilon \approx tr(\mathbf{K}_s - \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs})$, [5, 45, 46]. The process stops when the trace of the matrix corresponding to the actual approximation is less than a threshold. Note that using an RBF function, the trace is obtained as $tr(\mathbf{K}) = N$ where N denotes the size of the dataset. Then the criterion to choose the threshold can be a fraction of the dataset (N).
- Rank of $\mathbf{K}_r$ - the stopping condition used in [45]. If the matrix $\mathbf{K}_r$ is not well conditioned the process stops.

The outputs of the algorithm are the matrix $\mathbf{C}$ and the set of pivoting indexes which allow the choice of the subset $\mathbf{\Phi}_R$. Note that in those works the computation of the eigenvectors of $\mathbf{K}$ is according to eqn. 4.34 [5] or to eqn. 4.39 [47] with $\mathbf{L}$ obtained as the square root of $\mathbf{K}_r$ [31].

4.3.3 Experimental Study - EEG Signal

In this work, different strategies were studied to select the subset R. The first strategy was based on incomplete Cholesky decomposition computing the full kernel matrix. In [15, 14] an hybrid approach which leads to the choice of three subsets of data was considered. Firstly the data is splitted into two datasets: the training set which T vectors and the testing set which contains the remaining data to be processed. Two strategies were considered to form the training set: choosing T vectors randomly or, choosing the T vectors that correspond to a subsegment of the segment to be processed.
The subset R of the training set is chosen using the Cholesky decomposition with the size of set $\mathbf{\Phi}_R$ chosen a priori. This methodology was applied to the EEG signal to remove the proeminent EOG artifact. The results show that $R = 20$ pivots when T is selected randomly which is enough to denoise the signal, figure 4.7.

Choosing the T vectors that correspond to a subsegment of the signal, it is necessary to ensure that the artifact to be eliminated is in the subsegment used, otherwise it is impossible to extract the artifact from the signal [15]. This strategy was applied to a multidimensional EEG dataset to extract the artifacts [48].

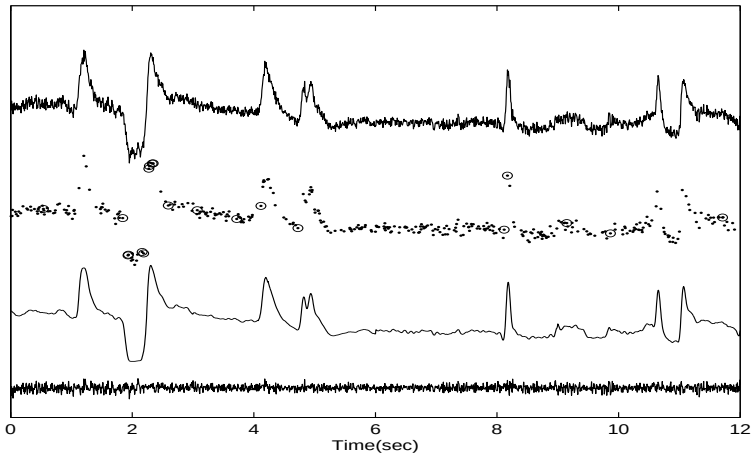


Figure 4.7: Illustration of the Greedy KPCA in different steps. Top to bottom: Corrupted EEG; random selected training set with the selected pivots with a circle; extracted EOG signal and corrected EEG signal.

4.3.4 Numerical Simulations

In this section some numerical simulations were done to study the impact of the projection approaches and its relation with the choice of subset $\mathbf{\Phi}_R$. For convenience of the exposition, a subdivision of the algorithms is made to deal with the computation of the parameters of the model. They are the following:

- **Chol**- Orthogonal approach using the incomplete Cholesky decomposition with symmetric pivoting. The subset $\mathbf{\Phi}_R$ is chosen according to the set of pivoting indexes and the matrix of basis vectors is computed using eqn. 4.40.

- **Cholr**- Orthogonal approach using a random selection of the subset $\mathbf{\Phi}_R$ followed by the Cholesky decomposition of $\mathbf{K}_r$. The matrix $\mathbf{C}$ and the matrix of basis vectors are computed using eqn. 4.36 and eqn. 4.40, respectively.

- **Nort**- Non-orthogonal approach using a random selection of $\mathbf{\Phi}_R$ using the eigendecomposition of $\mathbf{K}_r$ to compute the matrix of basis vectors as described by eqn. 4.35.

This procedure was validated in three different sorts of datasets: USPS dataset, sinusoidal signals and EEG signals to denoise them.
In this work only the USPS dataset and the sinusoidal signals results will be discussed. The EEG results are discussed in [13]. All experiments shown in the following sections were carried out with RBF kernel, eqn. 4.9, and with the fixed-point algorithm initialized with $\mathbf{p}_0$, eqn. 4.26. This algorithm is proposed, discussed and chosen in section 4.2.

USPS Dataset

For each type of digit, appendix A.1, the kernel matrix is computed with the total number of elements. This matrix is a full rank matrix (the minimum eigenvalues are $\simeq 0.17$).

The dataset for each digit type has different number of elements so it was considered a fixed percentage of 5% and 30%, to constitute the subset $\mathbf{\Phi}_R$ of the available data. Figure 4.8 shows a set of digits and its noisy versions.

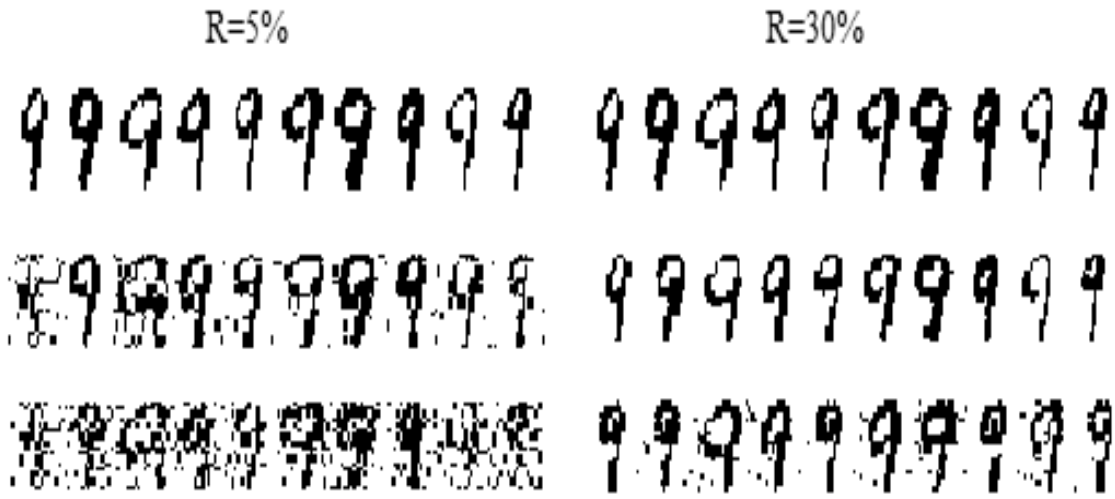


Figure 4.8: Set of denoised digits: first line - **Chol**, second line - **Cholr**, third line -**Nort**.

The denoising was achieved by projecting the data onto the leading $L < R$ eigenvectors found according to the leveling off of the eigenspectrum of the respective kernel matrix (in the range of $5 - 15$). The orthogonal approaches (**Chol** and **Cholr**) have better performance than the non-orthogonal approach (**Nort**). Figure 4.8 illustrates the methods performance for the two subsets and it can be verified that the differences between the orthogonal approaches might not be visually detected.
In table 4.2, the SNR mean values of the denoised images to all digits of the dataset are presented, and it can also be verified that all methods perform better if the subset $\mathbf{\Phi}_R$ is larger. However, the differences in performance for the two subsets are less accentuated with **Chol**, it does not exceed the $0.8dB$ to all the digits. It has to be noticed that **Cholr** presents a similar level of performance for the larger subset, the difference with **Chol** is less than $0.4dB$.
After numerous experiments to test the performance of the algorithms, the strategy adopted during this work is based on incomplete Cholesky (Appendix A.3) decomposition of the kernel matrix, where the criteria to stop the algorithm is the trace of the matrix.

Sinusoidal Signal

The kernel matrix of the noisy 2D signal, appendix A.1.1 has a dimension of $N = 498$ but the rank is 141 and 327 for $SNR = 20dB$ and $SNR = 0dB$, respectively. In feature space, the subspace dimension to recover the embedded sinusoid was $L = 2$. The three strategies to compute the basis vector $\mathbf{U}$ in feature space were implemented varying the size of subset $\mathbf{\Phi}_R$ between 10 and the rank of the kernel matrix. Table 4.3 shows the mean square errors

			SNR		
Digit	Image	R	**Chol**	**Cholr**	**Nort**
1	$\bar{x} = 0.162$	5 %	2.879	2.298	1.580
	$\sigma^2 = 2.177$	30 %	3.471	3.084	2.016
2	$\bar{x} = 2.729$	5 %	4.196	2.547	2.346
	$\sigma^2 = 2.834$	30 %	4.927	4.897	4.06
3	$\bar{x} = 2.890$	5 %	4.843	3.031	2.8928
	$\sigma^2 = 2.077$	30 %	5.235	5.108	4.372
4	$\bar{x} = 1.532$	5 %	3.788	1.865	1.678
	$\sigma^2 = 2.780$	30 %	4.085	3.985	3.450
5	$\bar{x} = 2.967$	5 %	4.498	3.086	3.018
	$\sigma^2 = 2.202$	30 %	5.269	5.118	4.859
6	$\bar{x} = 2.317$	5 %	4.343	3.016	2.897
	$\sigma^2 = 2.35$	30 %	5.030	5.149	4.247
7	$\bar{x} = 1.436$	5 %	4.126	2.081	1.999
	$\sigma^2 = 2.774$	30 %	4.671	4.453	3.836
8	$\bar{x} = 2.771$	5 %	3.891	2.255	2.615
	$\sigma^2 = 2.235$	30 %	4.698	4.613	4.431
9	$\bar{x} = 1.753$	5 %	4.425	3.012	2.767
	$\sigma^2 = 2.591$	30 %	4.877	4.722	4.215

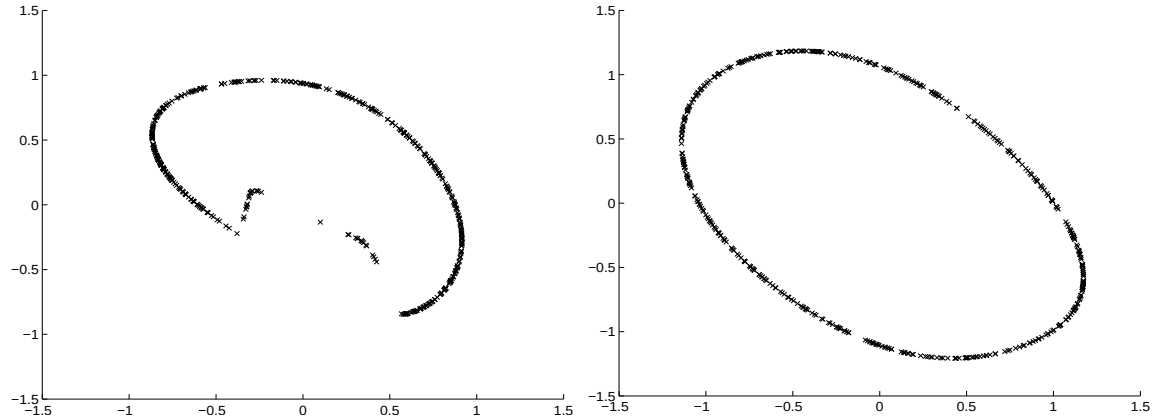
Table 4.2: SNR of the original and denoised images.

between the original sinusoid and denoised versions in two of the total set of experiments. Figure 4.9 illustrates in 2D the results in input space when subset $\mathbf{\Phi}_R$ has $R = 10$ elements. The figure also shows that the ellipse trajectory of the embedded sinusoid is recovered with a mean-square error of $MSE \simeq 0.16$. By the table it is visible that the orthogonal approaches are always better than the corresponding non-orthogonal approach. The difference is lower when the size of subset $\mathbf{\Phi}_R$ increases.

SNR		R=10	R=50
0dB	**Chol**	0.152	0.141
	Cholr	0.368	0.168
	Nort	0.671	0.386
20dB	**Chol**	0.004	0.004
	Cholr	0.162	0.004
	Nort	0.415	0.006

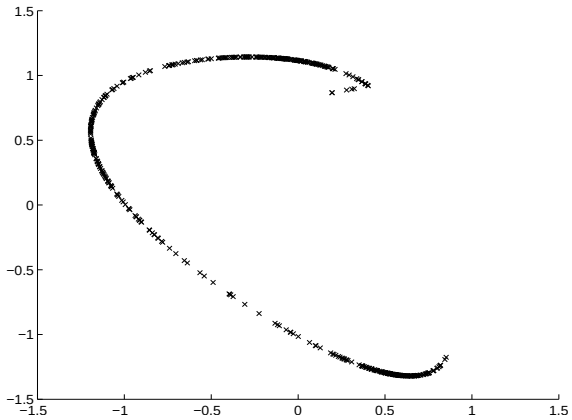
Table 4.3: Mean square error (MSE) between original and denoised versions (sinusoidal signals). Note that the entries of **Cholr** and **Nort** are mean of the result of 1000 random subset selections.

This example shows however, that if the SNR decreases the subset size (in the random strategies like **Cholr** and **Nort**), **Nort** should increase to assure that the subset covers the distribution of the input dataset. In fact, the Cholesky pivoting scheme assures the coverage of the input data distribution in a systematic way. The results obtained with the EEG signal confirm

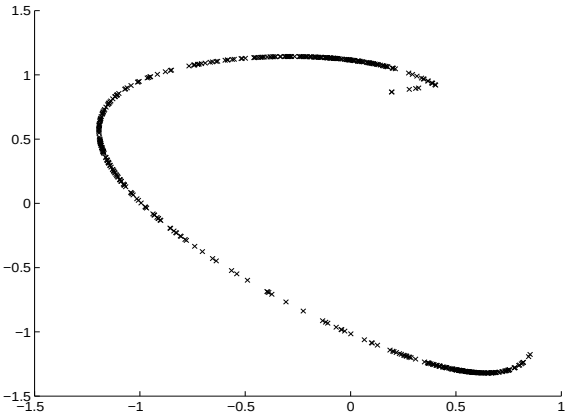


(a) Orthogonal - 0dB

(b) Orthogonal - 20dB



(c) Non-orthogonal - 0dB



(d) Non-orthogonal - 20dB

Figure 4.9: Denoising the embedded sinusoid considering different levels of noise, R=10 with **Cholr** and **Nort**.

the results obtained with other data sets, [13]. Comparing the corrected EEGs obtained with KPCA and this greedy approach, no visual difference can be found and the correlation coefficient is around 0.94 [13]. These simulations show that Greedy KPCA performs better with orthogonal approaches for both rank or non-rank deficient kernel matrices. The best results (in what concerns the size of subset R) were always achieved by the incomplete Cholesky with symmetric pivoting **Chol**. But Cholesky decomposition **Cholr**, after a random choice, can achieve very similar results at the expense of an increasing size of the subset R.

4.4 RBF Parameter

The RBF parameter can be chosen according to any suitable data spread criterion. In the literature, this parameter is selected by different ways: experimental work [22, 49], estimated by leave-one-out cross-validation [50] or based on the variance of the dataset [4, 22].
In this work two approaches have been used to compute the σ value:

1. In denoising applications:

$$\sigma = max_i \left\| \mathbf{x}_i - \mathbf{x}_{mean} \right\|, \quad i = 1, ..., N \tag{4.51}$$

2. In feature extraction:

$$\sigma = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathbf{x}_{mean} \right\| \tag{4.52}$$

   where $\mathbf{x}_{mean}$ is the mean of the dataset.

The value of σ determines the sparsity of the kernel matrix because the entries are in the range $]0, 1]$. By equation 4.9 it is possible to find some heuristics about the sparsity of the kernel matrix.

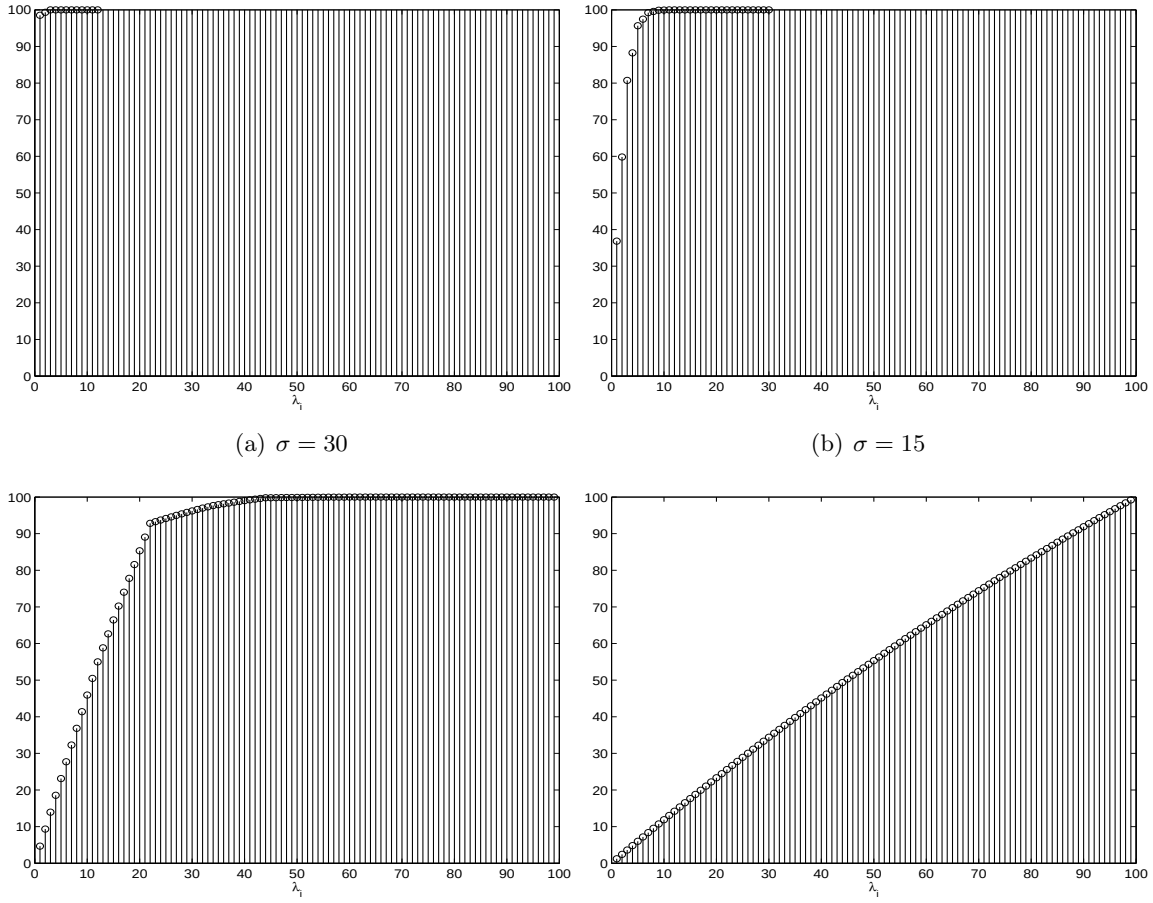
1. The entries of kernel matrix are equal or near 1 (non sparse matrix) when:

   - $\sigma >> \sqrt{\frac{1}{2} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|} \quad \forall i, j$

2. The entries of kernel matrix are near 0 (sparse matrix) when

   - $\sigma << \sqrt{\frac{1}{2} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|} \quad \forall i, j$

Using an RBF function to compute the kernel matrix, the trace (the sum of eigenvalues) is obtained as $tr(\mathbf{K}) = N$ where N denotes the size of the dataset whatever is the value of σ. Furthermore, it can be verified that the first eigenvalue can dominate the eigendecomposition for large σ or having a more smooth decay for smaller values. So, the parameter σ controls the decay of the eigenvalues of kernel matrix. Figure 4.10 represents the cumulative eigenspectrum of the kernel matrices in different sigma parameters. The kernel matrix is obtained by a sinusoid added with 0dB and embedded in 3D, section A.1.1. As a result, the number of directions in input space to project the data are only three. By default the number of directions available in feature space are $N = 100$, however the number of non-zero eigenvalues can vary if the σ parameter is altered. By inspection of figure 4.10, it is possible to obtain a diferent number of non-zero eigenvalues changing the σ parameter.
In this experiment four values for σ parameter were chosen: $\sigma = 5$ associated to eqn. 4.51; $\sigma = 15$ associated to eqn. 4.52; and two extreme values $\sigma = 0.5$ and $\sigma = 30$. If σ has a high value ($\sigma >> \sqrt{\frac{1}{2} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|} = 30$), the number of non-zero eigenvalues in feature space becomes similar to input space (only three). Decreasing the sigma value ($\sigma << \sqrt{\frac{1}{2} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|} = 0.5$), the number of non-zero eigenvalues available increases. In this case, the kernel matrix is almost a identity matrix and all eigenvalues are near 1 .
The results show that the number of non-zero eigenvalues available by the σ of eqn. 4.52 is higher than the σ of eqn. 4.51. For this reason, eqn. 4.51 to compute the σ value is used in denoising applications and eqn. 4.52 is used in feature extraction.



(a) $\sigma = 30$ (b) $\sigma = 15$

(c) $\sigma = 5$ (d) $\sigma = 0.5$

Figure 4.10: Cumulative sum of the kernel matrix eigenvalues for different sigma parameters using the 3D sinusoidal signal (section A.1.1).

4.5 Centering the Data in Feature Space

All deductions done in the last sections were conducted assuming that the data is centered. In input space this can be considered a pre-processing step that must be accomplished before computing the covariance or kernel matrix and before projecting any new data vector. However, in kernel methods this step must be integrated in the projection step. The issue to be discussed is the influence of centering the data onto the models. The proposal is to perform the centering and simultaneously maintain the new representation of the training dataset uncorrelated.

4.5.1 KPCA and a Complete Training Set

In feature space centering the mapped data is a more elaborate procedure that must be performed mostly during the computation of the projections. To facilitate the exposition, let's consider a vector $\mathbf{m}$ with N elements all of which equal $1/N$, and a matrix $\mathbf{M}$ filled with N column vectors $\mathbf{m}$. Therefore, to project a new data point $\phi(\mathbf{x}_j)$ and to take into

account the centered training dataset, the following operations need to be integrated into the dot product

$$(\mathbf{\Phi} - \mathbf{\Phi M})^T(\phi(\mathbf{x}_j) - \mathbf{\Phi m}) \tag{4.53}$$

The first term removes the mean to the training dataset, the second subtracts the mean of the training set from the new data. Then the manipulation of the previous expression results into four terms:

- $\mathbf{k}_1 = \mathbf{\Phi}^T\phi(\mathbf{x}_j)$ is a vector of dot products between the point and the training set
- $\mathbf{k}_2 = \mathbf{M\Phi}^T\phi(\mathbf{x}_j)$ is a vector with identical values in all entries. It is the mean of the dots products between the new point and the training set.
- $\mathbf{k}_3 = \mathbf{\Phi}^T\mathbf{\Phi m}$ is a vector with the mean values of the N rows of the kernel matrix $\mathbf{K}$.
- $\mathbf{k}_4 = \mathbf{M\Phi}^T\mathbf{\Phi m}$ is a vector with all the entries equal to $(1/N^2)k(i,j)$, where $k(i,j)$ is the entry (i,j) of kernel matrix.

Using eqn. 4.7, the L projections in feature space of the input data point x_j read

$$\begin{aligned} \mathbf{y}_{x_j} &= \mathbf{D}^{-1/2}\mathbf{V}^T(\mathbf{k}_1 - \mathbf{k}_2 - \mathbf{k}_3 + \mathbf{k}_4) \\ &= \mathbf{D}^{-1/2}\mathbf{V}^T(\mathbf{I} - \mathbf{M}^T)\mathbf{k}_1 - \mathbf{D}^{-1/2}\mathbf{V}^T(\mathbf{I} - \mathbf{M}^T)\mathbf{k}_3 \end{aligned} \tag{4.54}$$

The second summand in this difference only depends on the training set. It is present in every data point projected onto $\mathbf{U}$ and can be stored in advance and constitute a bias term. It can be easily shown that projecting the complete training set $\mathbf{\Phi}$ to obtain $\mathbf{Y}$, the last terms within parenthesis arise from the centered kernel matrix

$$\mathbf{K}_c = (\mathbf{I} - \mathbf{M})\mathbf{\Phi}^T\mathbf{\Phi}(\mathbf{I} - \mathbf{M}) \tag{4.55}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix. Then, to accomplish non-correlated projections for the training dataset the matrices $\mathbf{V}$ and $\mathbf{D}$ should be obtained from the eigendecomposition of $\mathbf{K}_c$. It should also be noticed that with an RBF kernel, the dot products in feature space are always less than the unit, eqn. 4.9, and in particular the contribution of the last two terms depends on the parameter σ of the kernel function.

4.5.2 KPCA and a Reduced Training Set

As described in section 4.3.2, to avoid the direct computation of the kernel matrix, the incomplete Cholesky decomposition is performed having as input the complete training set. The outcome is a $R \times N$ matrix $\mathbf{C}$, then to turn the projections related to the centered data, the low rank approximation of the kernel matrix can be centered

$$\tilde{\mathbf{K}_c} = (\mathbf{I} - \mathbf{M})\mathbf{C}^T\mathbf{C}(\mathbf{I} - \mathbf{M}) \tag{4.56}$$

where the mean $\mathbf{Cm}$ is subtracted from every column of $\mathbf{C}$. In that case the eigenvectors $\mathbf{V}_q$ must be computed with $\mathbf{Q}$ after centering the matrix $\mathbf{C}$. Then, the term $\mathbf{b} = \mathbf{V}_q^T\mathbf{Cm}$ should

also be subtracted from every data projected onto the model,

$$\mathbf{y}_{x_j} = \mathbf{U}^T \phi(\mathbf{x}_j) - \mathbf{b} \tag{4.57}$$

4.6 Conclusions

Kernel Principal Component Analysis is widely used in classification, feature extraction and denoising applications. This method created by a non-linear transformation of the original dataset represents a projective subspace technique applied in feature space.
In denoising or classification problems, this method can simultaneously denoise or classify with better performance than the linear subspace methods. This happens because the projections are accomplished in the higher dimensional feature space, while still retaining the non linear structure of the data.
In this chapter the two bottlenecks of KPCA were studied, namely the pre-image problem and the dimension of the kernel matrix .
The pre-image step influences the outcome of the algorithm. In denoising applications it is unavoidable to deal with the pre-image problem which constitutes the most complex step in the whole processing chain. One of the methods to tackle this problem is an iterative solution based on a fixed-point algorithm. An alternative strategy considers an algebraic approach that relies on the solution of an undetermined system of equations. In this work, a method that uses this algebraic approach to estimate a good starting point to the fixed-point iteration is present. The two methods of pre-image estimation in literature were discussed and simple modifications were suggested, which have proved to be very robust and effective in the applications that were carried out. The comparison of them on a denoising task revealed that the solution obtained by the distance method strongly depends on the number of nearest neighbors chosen and sometimes does not yield reliable results. Further, if the number of nearest neighbors is smaller than the dimension of the data space, the distance method can often be closely approximated by simply choosing the mean of the nearest neighbors which speeds up computation considerably. In addition, a new proposal to estimate the pre-image was done - mean of neighbors.
Considering the fixed-point algorithm with a random initialization, as suggested in literature, a very slow convergence is often the result. Initializing the algorithm with the mean of the nearest neighbors considerably speeds up convergence and yields a very robust algorithm.
The eigendecomposition of a kernel matrix can present a computational burden in many kernel methods if the datasets are large. Nevertheless, only the largest eigenvalues and corresponding eigenvectors need to be computed. The exploitation of methods like Nyström to achieve the low rank eigendecomposition, is a strategy that has been considered. Furthermore, those techniques can also achieve a solution without the manipulation of the full matrix. In a common algebraic framework, the Nyström approaches to compute the basis were discussed. These approaches differ on the way the eigenvectors are computed: one achieves orthogonal eigenvectors, the other does not. The main differences are the complexity of the different approaches and the properties of the computed projections. Among all possible variants found

in the literature, numerical simulations show that there is no additional profit in adding a step to select the proper subset. The centering problem was considered and the model description to remove the mean of the data was adapted.

Bibliography

[1] B. Boser, I. Guyon, and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," 1992 1992.

[2] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to Kernel-Based Algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–202, 2001.

[3] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[4] R. Rosipal, M. Girolami, L. Trejo, and A. Cichocki, "Kernel PCA for Feature Extraction and De-Noising in Non-linear Regression," *Neural ComputingApplications*, vol. 10, pp. 231–243, 2001.

[5] V. Franc and V. Hlaváč, "Greedy Algorithm for a Training Set Reduction in the Kernel Methods," in *10th International Conference on Computer Analysis of Images and Patterns*, (Groningen, Holland), pp. 426–433, Springer, 2003.

[6] F. R. Bach and M. I. Jordan, "Kernel Independent Component Analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[7] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller, "Fisher Discriminant Analysis with Kernels," in *Proceedings of the 1999 IEEE Signal Processing Society Workshop*, (Madison, WI, USA), pp. 41–48, 1999.

[8] A. Jade, B. Srikanth, V. Jayaraman, B. Kulkarni, J. Jog, and L. Priya, "Feature Extraction and Denoising using Kernel PCA," *Chemical Engineering Science*, vol. 5, no. 19, pp. 4441–4448(8), 2003.

[9] K. Kim, M. O. Franz, and B. Schölkopf, "Iterative Kernel Principal Component Analysis for Image Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1351 – 1366, 2005.

[10] J. Wei, L. Jiake, W. Xuan, and S. Rongrong, "A Novel Hybrid Approach of KPCA and SVM for Crop Quality Classification," *Computer Science and Software Engineering, International Conference on*, vol. 4, pp. 739–742, 2008.

[11] G. Blanchard, P. Massart, R. Vert, and L. Zwald, "Kernel Projection Machine: a New Tool for Pattern Recognition," 2004.

[12] A. Teixeira, A.M.Tomé, K. Stadlthanner, and E. W. Lang, "KPCA denoising and the pre-image problem revisited," *Digital Signal Processing*, vol. accepted, 2007.

[13] A. R. Teixeira, A.M.Tomé, and E.W.Lang, "Exploiting Low-Rank Approximations of Kernel Matrices in Denoising Applications," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2007)*, (Thessaloniki, Greece), 2007.

[14] A. R. Teixeira, N. Alves, A.M.Tomé, M.Böhm, E.W.Lang, and C.G.Puntonet, "Single-channel electroencephalogram analysis using non-linear subspace techniques," in *IEEE International Symposium on Intelligent Signal Processing (WISP 2007)*, (Madrid, Spain), 2007.

[15] A. R. Teixeira, A. M. Tomé, and E. W. Lang, "Greedy KPCA in biomedical signal processing," in *LNCS 4669- International Conference on Artificial Neural Networks-ICANN 07*, (Porto, Portugal), pp. 486–495, 2007.

[16] A. Teixeira, A.M.Tomé, K. Stadlthanner, and E. W. Lang, "KPCA denoising and the pre-image problem revisited," *Digital Signal Processing*, vol. accepted, 2007.

[17] A. R. Teixeira, A.M.Tomé, E.W.Lang, R. Schachtner, and K.Stadlthanner, "On the use of KPCA to extract artifacts in one-dimensional biomedical signals," in *Machine Learning for Signal Porcessing, MLSP 2006* (S. McLoone, J. Larsen, M. V. Hulle, A. Rogers, and S. C. Douglas, eds.), (Dublin), pp. 385–390, IEEE, 2006.

[18] A. R. Teixeira, A. M. Tomé, K. Stadlthanner, and E. W. Lang, "KPCA denoising and the pre-image problem revisited," *Digital Signal Processing*, vol. 18, pp. 568–590, 2008.

[19] A. R. Teixeira, A. M. Tomé, and E. W. Lang, "Feature Extraction using Low-Rank Approximations of the Kernel matrix," in *LNCS 5112- ICIAR 2008* (A. Campilho and M. Kamel, eds.), (Porto), pp. 404–412, 2008.

[20] J. T. Kwok and I. W. Tsang, "The Pre-Image Problem in Kernel Methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, 2004.

[21] B. Schölkopf, S. Mika, A. Smola, G. Ratsh, and K.R.M"uller, "Kernel PCA Pattern Reconstruction via Approximate Pre-Images," in *Proceedings of the 8th International Conference on Artificial Neural Networks*, (Berlin), pp. 147–152, Springer Verlag, 1998.

[22] B. Schölkopf, S. Mika, C. J. Barges, P. Knirsch, K.-R. Müller, G. Ratsch, and A. J.Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1016, 1999.

[23] G. Bakir, J. Weston, B. , and Schölkopf, "Learning to Find Pre-Images," *Advances in Neural Information Processing Systems*, vol. 16, pp. 449–456, 2004.

[24] C. J. C. Burges, "Simplified support vector decision rules," in *Proceedings of the 13th International Conference on Machine Learning* (I. L. Saitta, ed.), (San Mateo, CA), pp. 71–77, 1996.

[25] J. C. Gower, "Adding a point to vector diagram in multivariate analysis," *Biometrika*, vol. 55, pp. 582–585, 1968.

[26] J.T.Kwork and I.W.Tsang, "The Pre-Image Problem in Kernel Methods," in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, (Washington DC), pp. 408–415, 2003.

[27] T. Takahashi and T. Kurita, "Robust de-noising by kernel PCA," in *ICANN2002* (J. Dorronsoro, ed.), LNCS 2415, (Madrid, Spain), pp. 739–744, Springer-Verlag, 2002.

[28] A. Smola and B.Schölkopf, "Sparse Greedy matrix Approximating for Machine Learning," in *Proceedings on 17th International Conference on Machine Learning*, pp. 911–918, 2000.

[29] F. R. Bach and M. I. Jordan, "Predictive low-rank decomposition for kernel methods," in *International Conference on Machine Learning*, (Bonn, Germany), 2005.

[30] M. Ouimet and Y. Bengio, "Greedy spectral embedding," in *10th Workshop on Artificial Intelligence and Stastistics*, 2005.

[31] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping using the Nystr'om Method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–224, 2004.

[32] M. Bonis and C. Laurita, "Nyström method for Cauchy singular integral equations with negative index," *Journal of Computational and Applied Mathematics*, vol. 232, no. 2, pp. 523–538, 2009.

[33] C. K. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, pp. 682–688, MIT Press, 2001.

[34] K. Zhang, I. Tsang, and J. Kwok, "Improved nyström low-rank approximation and error analysis," in *Proceedings of the 25 th International Conference on Machine Learning*, (Helsinki, Finland), 2008.

[35] S. Marukatat, "Sparse Kernel PCA by Kernel K-means and preimage reconstruction algorithms," in *PRICAI* (Springer, ed.), pp. 451–463, 2006.

[36] P.Drineas and M.Mahoney, "On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.

[37] J. Zhang, H. Chung, and W. Lo, "Chaotic time series prediction using a neuro-fuzzy system with time-delay coordinates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, p. 956964, 2008.

[38] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, pp. 682–688, MIT Press, 2000.

[39] C. Fowlkes, S. Belongie, and J. Malik, "Efficient spatiotemporal grouping using the Nyström method," in *Conference in Computer Vision and Pattern Recognition*, IEEE, 2001.

[40] V. Franc and V. Hlaváč, "Stastical pattern recognition toolbox for matlab," 2004.

[41] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[42] R. Liu, V. Jain, and H. Zhang, "Sub-sampling for Efficient Spectral Mesh Processing," in *Computer Graphics International*, (China), 2006.

[43] H. Wang, Z. Hu, and Y. Zhao, "An efficient algorithm for generalized discriminant analysis using incomplete cholesky decomposition," *Pattern Recognition Letters*, vol. 28, pp. 254–259, 2007.

[44] F. R. Bach, "Kernel Independent Component Analysis," 2003.

[45] G. C. Cawley and N. L. C. Talbot, "Efficient Formation of a Basis in a Kernel Induced Feature Space," in *European Symposium on Artificial Neural Networks* (M. Verleysen, ed.), (Bruges, Belgium), pp. 1–6, d-side, 2002.

[46] G. Baudat and F. Anouar, "Feature Vector Selection and Projection using Kernels," *Neurocomputing*, vol. 55, pp. 21–38, 2003.

[47] G. Baudat and F. Anouar, "Kernel-based methods and function approximation," in *International Joint Conference on Neural Networks*, vol. 2, (Washington, USA), pp. 1244–1249, IEEE, 2001.

[48] A. M. Tomé, A.R.Teixeira, E.W.Lang, and A. M. d. Silva, "Greedy KPCA applied to single- channel EEG recordings," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2007)*, (Lyon), 2007.

[49] S.Mika, B.Schölkopf, and all, "Kernel PCA and Denoising in Features Spaces," in *Advances in Neural Information Processing Systems 11*, (San Mateo), pp. 536–542, Cambridge, MA: MIT Press, 1998.

[50] H. Hoffmann, "Kernel PCA for Novelty Detection," *Pattern Recognition*, vol. 40, pp. 863–874, 2006.

[51] A. Delorme and S. Makeig, "EEGLAB: An Open Source Toolbox for Analysis of Single-Trial EEG Dynamics," *Journal of Neuroscience Methods*, vol. 134, pp. 9–21, 2004.

Chapter 5

Time Series - Denoising

"The brain is a world consisting of a number of unexplored continents and great stretches of unknown territory"
- Santiago Ramon y Cajal -

Contents

Artifacts are a major noise source in electroencephalogram recordings. Eye movement, blinks and patient movement produce electrical interference that distort the electric field over the scalp. These artifacts often complicate the interpretation of the EEG. To correct and remove

these artifacts from the EEG many techniques have been developed in the literature.
In this work it will be shown that projective subspace techniques can be used favorably to get rid of most of the noise contributions in multidimensional signals. The goal of subspace methods is to project the noisy signal onto two subspaces: the signal subspace and the noise subspace.
Hence an estimate of the clean signal can be made by removing or nulling the components of the signal in the noise subspace, retaining only the components in the signal subspace. In this chapter, two strategies to denoise the signal will be compared: techniques based on singular spectrum analysis and techniques based on kernel methods. The goal was to study the performance of these algorithms when applied to EEG signals, artificially mixed with EOG artifacts. The study was to compare the algorithms performance, regarding the main frequency bands of interest (beta, alpha, theta and delta) of the EEG.

This chapter is organized as follows: in section 5.1 a background about artifact removal is presented. An explanation about the EEG signals, the recorded artifact and the artificial mixtures used will also be exposed in sections 5.2 and 5.3. The correlation coefficient and the coherence function are computed to measure the performance of the algorithms in frequency and in time, section 5.4. After this, the results obtained by SSA, Local SSA, Greedy KPCA and ICA algorithms are discussed, section 5.5. A preliminary real application of the Local SSA and Greedy KPCA to remove artifacts is done and the results obtained are compared among them, section 5.6. Finally, some conclusions are made, section 5.7. Note that all the results present in this chapter are published already in [1, 2, 3, 4, 5, 6].

5.1 Artifacts Removal

The availability of digital EEG recordings allows the investigation of procedures trying to remove artifact components from the recorded brain signals. The primary goal will be to remove such superimposed artifacts without distorting the underlying brain signals. Several automatic procedures have been proposed in the literature to correct or remove ocular artifacts from EEG recordings.
One common strategy is artifact rejection. This procedure is laborious, slow and often results in considerable loss of the EEG data. Artifact rejection is often used to eliminate artifacts in evoked potential studies. In [7] a detailed review of reduction strategies in evoked potential studies is discussed. Traditionally, EOG artifacts have been corrected using regression methods in time [8] and frequency domains [9]. To achieve that, the ocular channel must be recorded placing an electrode near the eye. The corrected EEG is obtained by regressing out the recorded EOG from the EEG signal. It consists in the subtraction of the scaled EOG channel (or horizontal and vertical EOG recording channels) from the EEG signal. Recently, in [10] a new regression method was presented to extract EOG artifacts and the same has been implemented for online analysis and is available through BioSig [11]. Adaptive digital filters have also been used for ocular artifact removal [12, 13, 14, 15]. The main limitation

of the two methods referred above, consists in the obligation to record the reference EOG to subtract to the signal or to adapt to the filter model.
Many methods, [16, 17, 10], need a reference channel. This reference channel must be recorded simultaneously with the EEG. However, these reference signals never provide a pure reference to the artifact. For example, the reference EOG signal always contains EEG contaminations. In [18] a comparison between time domain regression and adaptive filtering methods using simulated data was made. The authors conclude that when there is a shape difference or a misalignment between the reference EOG and the EOG artifact, the adaptive filtering method can be more accurate. Spatial filters are also a method described in the literature to remove artifacts from the EEG signals [19]. The quality of this algorithm depends crucially on how well the topographies separate artifacts and brain activity. If the artifact and brain activities are not modulated adequately, some distortion of spatially correlated brain activity occurs.
In [20] a method based on the detection of the EOG activation periods from a reference EOG channel is presented. The Generalized Eigenvalue Decomposition (GEVD) method was used to detect the artifacts. Another class of methods is based on linear decomposition of the EEG and EOG, thus resulting source components, so that one can identify artifact components and finally, reconstruct the EEG without artifactual components. These methods allow the isolation of the artifact and cerebral activity.
PCA and SVD are the classical approaches, proposed in [21] and in [22], respectively. Recently, BSS [23, 24] techniques and specially ICA [16, 17], have been used in the extraction of EEG artifacts. ICA has been described as an algorithm capable of extracting EOG artifacts as well as artifacts generated by another source [25]. The most recent works use independent component analysis: in [26] use the INFOMAX algorithm was used, in [27, 28] the joint approximate diagonalization of eigen-matrices algorithm (JADE) was applied, in [24] an approximate joint diagonalization of time-delayed correlation matrices (SOBI) was used, while in [29] the fast fixed point algorithm (FASTICA) was applied. Another work, concluded that JADE and ICA were more effective than EOG subtraction (regression methods in time) and PCA to remove EOG artifacts. [30].
In the majority of works, the EOG channel is used on the processed dataset [16, 17], however there are works showing that ICA is possible without the EOG signal, [29, 31]. Although in the literature these methods have been presented as automatic and capable of eliminating any kind of artifacts, in real situations this does not happen.
There are several works that compare the artifacts extraction methods, but it is impossible to conclude which one is the best. In [32] a variety of procedures has been proposed to correct ocular artifacts in real and simulated EEG, (PCA, ICA and regression methods). The majority of the methods presented are focused on the extraction of a single type of artifact. The algorithm application is not always possible on signals with different artifacts.
The proposed methods to remove artifacts from EEG recordings are based on singular spectrum analysis (Local SSA) and kernel methods (Greedy KPCA), described in chapters 3 and 4. These methods are one-dimensional algorithms, and in the case of the EEG, this is even better because it only needs as input a single channel recording. This is a definite advantage as artifacts appear differently in different channels in certain segments, some may not even

contain artifacts at all, therefore, artifacts can be processed more specifically in each channel if needed. Another advantage concerns the identification of the artifact related components in projection methods, which generally can become very tedious in methods like ICA. With the studied methods, there is a natural assignment of high amplitude artifacts to signal components associated to the largest eigenvalues of the decomposition. Moreover, Local SSA, KPCA and Greedy KPCA do not need a proper reference signal, like separately recorded EOG signal, but regression and adaptive filtering methods do.

5.2 EEG: Overview

Electroencephalography (EEG) is the recording of electrical activity along the scalp produced by the firing of neurons within the brain [33]. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a short period of time, usually [20 40] minutes, as recorded from multiple electrodes placed on the scalp. In neurology, the main diagnostic application of EEG is in epilepsy. The epileptic activity creates abnormalities on a standard EEG recording. such as high amplitude depolarization waves of membrane potential. A secondary clinical use of EEG is in the diagnosis of coma and encephalopathies. EEG used to be a first-line method for the diagnosis of tumors, stroke and other focal brain disorders.
Derivatives of the EEG technique include evoked potentials (EP), which involves the averaging of EEG activity time-locked to the presentation of a stimulus of some sort (visual, somatosensory, or auditory). Event-related potentials refer to averaged EEG responses that are time-locked to more complex processing of stimuli; this technique is used in cognitive science, cognitive psychology, and psychophysiological research.

5.2.1 EEG Bands

The EEG is typically described in terms of rhythmic activity [33]. The rhythmic activity is divided into bands by frequency. Most of the cerebral signal observed in the scalp EEG falls in the range of $125Hz$ (activity below or above this range is likely to be artifactual, under standard clinical recording techniques). The principal bands are:

1. delta: [0.5 , 3.5] Hz
2. theta: [3.5 7.5] Hz
3. alpha: [7.5 , 13] Hz
4. beta: [13 , 25] Hz

5.2.2 Type of Artifacts

EEG records are often contaminated with extra cerebral signals that are originated from non-cerebral source called artifacts. Artifacts are noises introduced to an EEG signal by biological sources or by sources of electric field outside the patient's body. The amplitude of artifacts

can be quite large relative to the size of amplitude of the cortical signals of interest. This is one of the reasons why it takes considerable experience to correctly interpret clinically the EEG. Some of the most common types of biological artifacts include: EOG (Eye-induced artifacts that include eye blinks and eye movements), EKG (cardiac artifacts), EMG (muscle activation), induced artifacts and Glossokinetic artifacts. In addition to artifacts generated by the body, many artifacts originate from outside the body. Movement by the patient, or even the settling of the electrodes, may cause electrode pops (spikes originating from a momentary change in the impedance of a given electrode).

5.3 Data Collection

The data for this work was recorded at Hospital Geral Santo António and belongs to a group of patients with several pathologies. The signals were selected by three specialists after visual inspection. The signals were selected with a clear predominance in one of the characteristic bands (beta, alpha, theta and delta) and clean of artifacts (EOG, EMG or patient movements), figure 5.1.

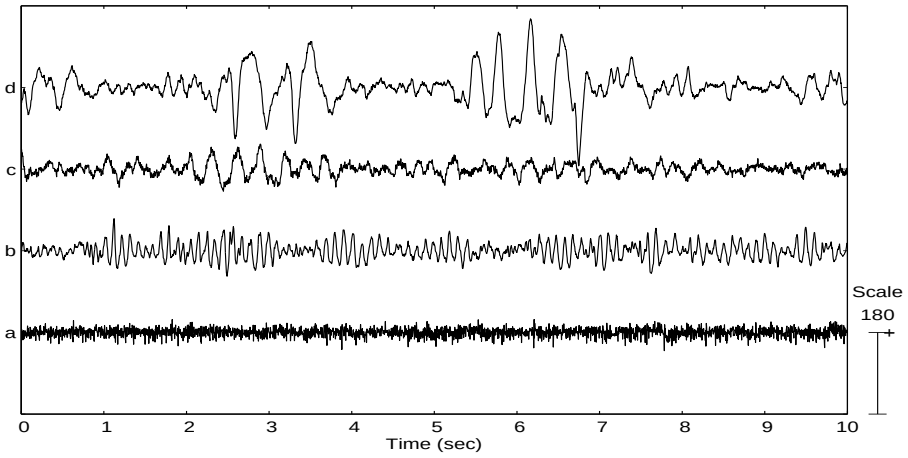


Figure 5.1: Illustration of EEG signals with different activities: a - Beta activity, b - Alpha activity, c - Theta activity and d - Delta activity.

In this study, a strategy proposed in [34] was followed where the multichannel data set is obtained by a mixing model. More details about the artifacts and the mixtures can be found in section A.2. Figure 5.2 illustrates an artificial mixture where the artifact is added to each of the original signals represented in figure 5.1.

5.4 Performance Measures

To quantify the effectiveness and to quantitatively compare the accuracy of each algorithm in the extraction of the artifacts, time and frequency domain measures were used. The comparison was made between the corrected EEG ($y[k]$) and the original EEG ($x[k]$).

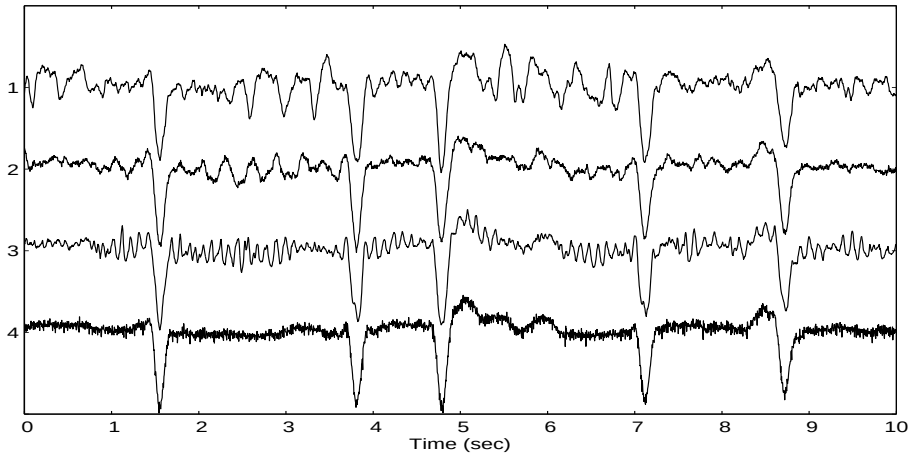


Figure 5.2: Illustration of an artificial mixture in different activities represented in figure 5.1.

5.4.1 Time Domain

In time domain, the correlation coefficient was calculated providing a measure of the similarity in shape for the recovered EEG.
This measure is independent of scaling or mean (m) differences. The absolute value ranges from [0 , 1] and is expressed by

$$ct_{xy} = \frac{\sum_{k=0}^{K-1} (x[k] - m_x)\,(y[k] - m_y)}{\sigma_x \sigma_y} \tag{5.1}$$

where σ_x and σ_y represent the standard deviations of the amplitude of the segments x and y respectively.

5.4.2 Frequency Domain

In frequency domain, a coherence function between the spectra of the two segments was measured to evaluate the similarity in spectrum energy. This function has values in the range [0 , 1] and is computed using the periodograms of subsegments. Partitioning the signals into L segments and given the Discrete Fourier Transform (DFT) of the ith subsegment of each signal, X_i and Y_i, the coherence of the mth bin in frequency is defined as

$$cf_{xy}(m) = \frac{\left|\sum_{i=1}^{L} X_i^*(m) Y_i(m)\right|^2}{\sum_{i=1}^{L} |X_i(m)|^2 \sum_{i=1}^{L} |Y_i(m)|^2} \tag{5.2}$$

where X^* represents the complex conjugate of X.
In the experimental results to be discussed in the next sections, the subsegments have an overlap of 50% and the DFT was computed with a resolution of 1 Hz. Furthermore, the

coherence values are presented for each of the four characteristic EEG bands by averaging the bins within the frequency range of the band.

5.5 Subspace Methods Application

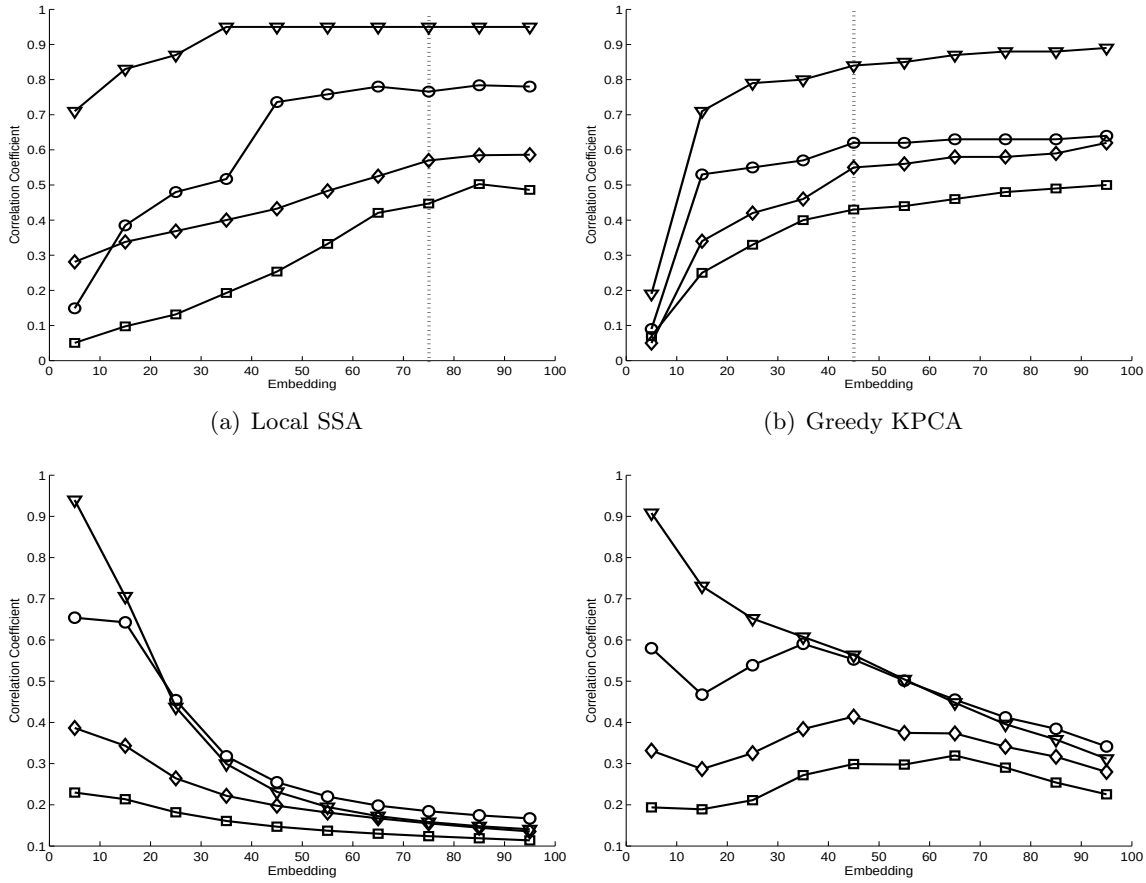
The two projective subspace techniques are evaluated using the real data but artificially mixed. In this thesis, only the results related to the EOG artifact will be exposed [5, 4]. The artificial data is used to quantify the performance of the algorithms, particularly the influence of the embedding dimension M.
The algorithms were applied to every signal of the artificial dataset and the performance measures were taken between the corrected and original EEGs. The complexity of the approaches will also be discussed in this study. The dataset $\mathbf{X}$ formed either by using a multichannel analysis or a unichannel analysis can be used to remove artifacts in EEG recording systems. It has to be noticed that using a unichannel analysis, the artifact is removed from a single channel while with a multichannel analysis the artifact is removed from a set of channels.

5.5.1 Embedding Dimension

The subspace techniques discussed so far are applied to multidimensional signals resulting in an embedding of the recorded time series $x[k]$ into their delayed coordinates. Then, the embedding dimension M is a choice to be made before the application of the two subspace techniques studied. The value of M is changed between [6 96] in steps of 5. The goal was to understand the influence of the M parameter in EEG signals with different activities. After embedding, each segment of the dataset is represented by a multidimensional dataset $\mathbf{x}_n$, $n = 1, ..., N$ and will be the input of the algorithms. The output is the extracted artifact $y[k]$ which will be subtracted from the mixed signal to obtain the corrected EEG ($x[k]$) as described in section 2.5.1. In this experiment different algorithms were considered:

1. SSA with 1 direction ($L = 1$)
2. SSA using the MDL algorithm
3. Local SSA using the MDL algorithm, to select the directions in each cluster
4. Greedy KPCA using 95% of explained variance and a threshold to stop the algorithm $\epsilon \leq 0.01N$

To study the dependence of the performance on the subspace dimension M, the correlation coefficient cc_{oy} between the original and corrected EEG was considered, figure 5.3. The level of performance depends on the dominant frequency range of the original EEG: from 0.9 (Type D) to 0.4 (Type A), being more reliable for segments with dominant frequencies ranging far from the frequency contents of the artifacts. By the results present in figure 5.3, it can be verified that to achieve a stable behavior, the embedding dimension M for Greedy KPCA can be smaller ($M = 46$) than with Local SSA ($M = 76$). However, the level of performance is worse than the one achieved with Local SSA despite having a similar tendency. In the



(a) Local SSA

(b) Greedy KPCA

(c) SSA (L=1)

(d) SSA MDL

Figure 5.3: Mean Correlation coefficient (ct_{oy}) versus embedding dimension (M):*left*- Local SSA and *right*- Greedy KPCA. Segment Type: □ type A; ⋄ type B; ∘ type C and ∇ type D.

SSA variations, (SSA with 1 direction and SSA using the MDL algorithm), the embedding dimension M was not fixed; it was instead kept variable and the output was chosen accordingly to the correlation between the corrected and original signals. This way, the implementation is not useful in any practical application as the artifact-free original signal is not available.

5.5.2 Performance of the Algorithms

The study was made in EEG signal artifact elimination considering distinct EEG activities described in detail in section A.2. The algorithms used were: ICA, SSA, Local SSA and Greedy KPCA. The considerations used on each algorithm are as follows:

- ICA

  The purpose of the experiment was to compare the performance of RunICA algorithm in artifact elimination with or without the artifact channel. This algorithm is implemented in the EEGLAB [35] platform which has a built in facility to remove the artifacts based on the INFORMAX algorithm [36].

  In these experiments the component that was visually assigned as the EOG artifact will

be removed and it has to be noticed that in all runs the algorithm found the number of independent components equal to the number of provided mixtures (16 - ICA16 or 17 - ICA17). After the identification of the component related to the EOG, the multichannel recording is constructed using the pseudo-inverse of the separation matrix.

- SSA
  In the case of the SSA algorithm, the M parameter is chosen accordingly to the frequency contents of the EEG and only one direction $L = 1$ is selected.

- Local SSA
  In Local SSA the M parameter is chosen according to the heuristic $M = 76$ as shown in the last section. The number of clusters automatically assigned for the dataset varied between $2 - 9$ and did not depend on the EEG segment used to generate the artificial mixture, it was, instead, related to the artifact. If the segment has only one or two blinks and no baseline drifts, the number of clusters is 2. For an increasing number of blinks and baseline drifts or ocular movements, the number of clusters also increases.

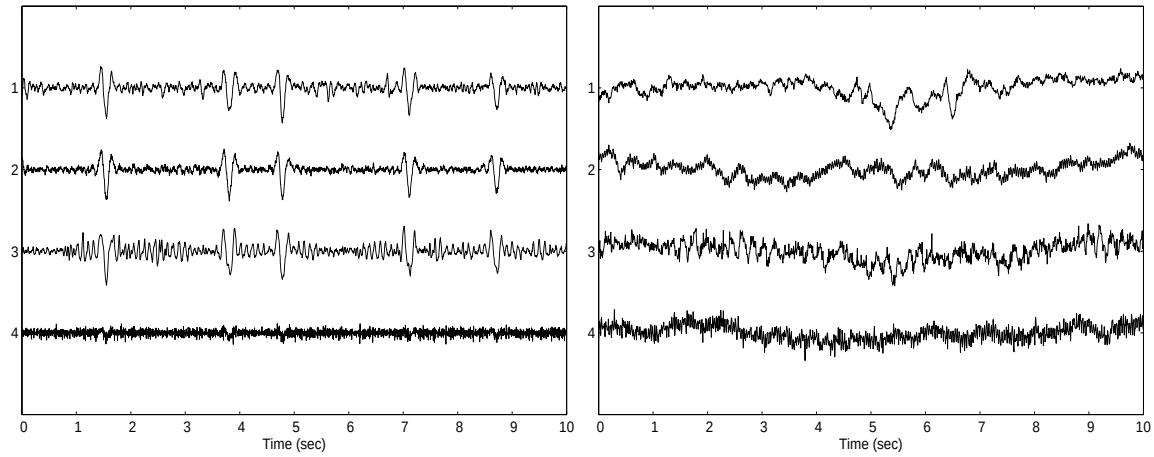
- Greedy KPCA
  The Greedy KPCA was applied using the incomplete Cholesky decomposition, where the threshold to stop the algorithm was $\epsilon \leq 0.01N$. After the eigendecomposition of matrix $\mathbf{Q}$, the number of directions L was chosen to maintain 95% of the variance of the data in feature space, eqn. 3.12. The number of pivots needed to fulfill the error criterion change for the different segments. Table 5.1 shows the range of values for each

	Pivots (R)			Directions (L)		
Type	Min	Med	Max	Min	Med	Max
A	33	79	207	14	24	46
B	24	74	177	10	20	39
C	34	88	390	13	25	77
D	33	90	645	9	18	48

Table 5.1: Greedy KPCA (Min-Minimum; Med-Median; Max - Maximum).

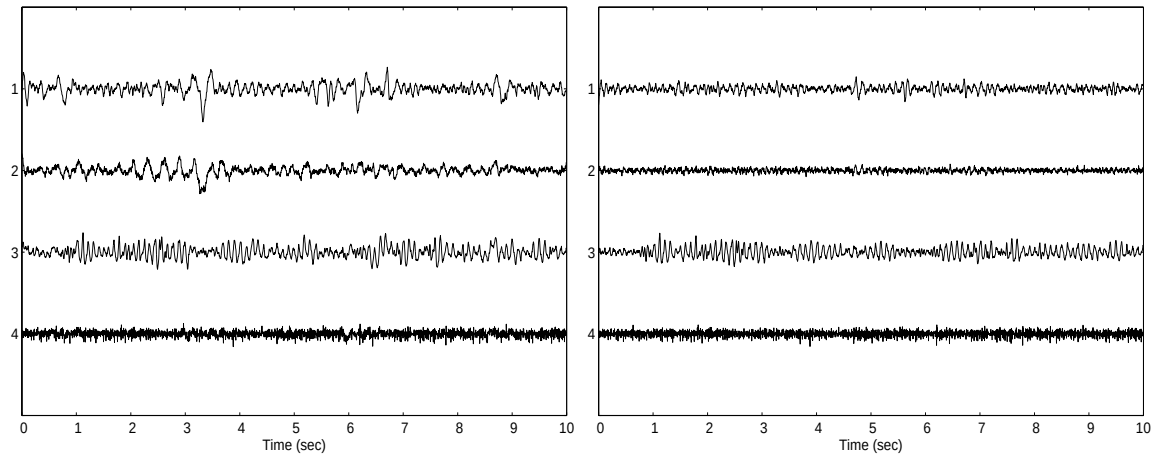
  type of segment when $M = 46$ (value considered by figure 5.3 (b)).
  The number of pivots is related to the type of artifact, not with the EEG segment used in the mixture. Note that the size of the training dataset for each segment is $N = 2456$ and the R median value is less than 100 in all cases. The maximum values only occur for a segment that has simultaneously ocular artifacts and baseline drifts. Furthermore notice that the maximal values of L reveal that less than 1/4 of the computed eigenvectors of $\mathbf{Q}$ are used.

The performance of these algorithms in artifact removal from the EEG was quantified and compared. Figure 5.4 illustrates an example of the output of algorithms for the segments of the corrupted signal represented in figure 5.2. It can be verified the difference on the performance of the algorithms, namely for Type A and Type B where some slow wave is only



(a) SSA

(b) ICA



(c) LOCAL SSA

(d) Greedy KPCA

Figure 5.4: Corrected EEG segments using different linear subspace techniques: *Top to Bottom*- Segment type: (1st) Type A- Delta, (2nd) Type B- Theta, (3rd) Type C- Alpha and (4th) Type D-Beta.

visible in the Local SSA outcome. Note that with SSA with one projection, the artifact is still visible except for the segments where beta dominates. Table 5.2 presents the correlation coefficients for the segments represented in figure 5.4 and the values confirm that when the correlation coefficient is low, the distortions are visible. In the frequency domain the results also confirm the time domain ones.

Figure 5.5 shows a comparison of the correlation coefficients between the output of the algorithms and the original for all the segments in the dataset. The level of distortion is related to the segment type. The correlation coefficient decreases as the EEG frequency contents are closer to the frequency of the artifact. But in all cases, Local SSA performs better than Greedy KPCA and SSA. Note that ICA17 has a correlation coefficient higher than ICA16, because the EOG artifact is used in the decomposition thus increasing the algorithm performance. The analysis in the frequency range confirms these results, figure 5.6. Whatever is the segment, the beta band is always the least distorted, i.e. the coherence function cf_{oy} has

	Correlation Coefficient			
Type	Local SSA	Greedy KPCA	SSA	ICA17
A	0.54	0.30	0.38	0.38
B	0.63	0.42	0.37	0.11
C	0.87	0.65	0.67	0.05
D	0.95	0.91	0.91	0.34

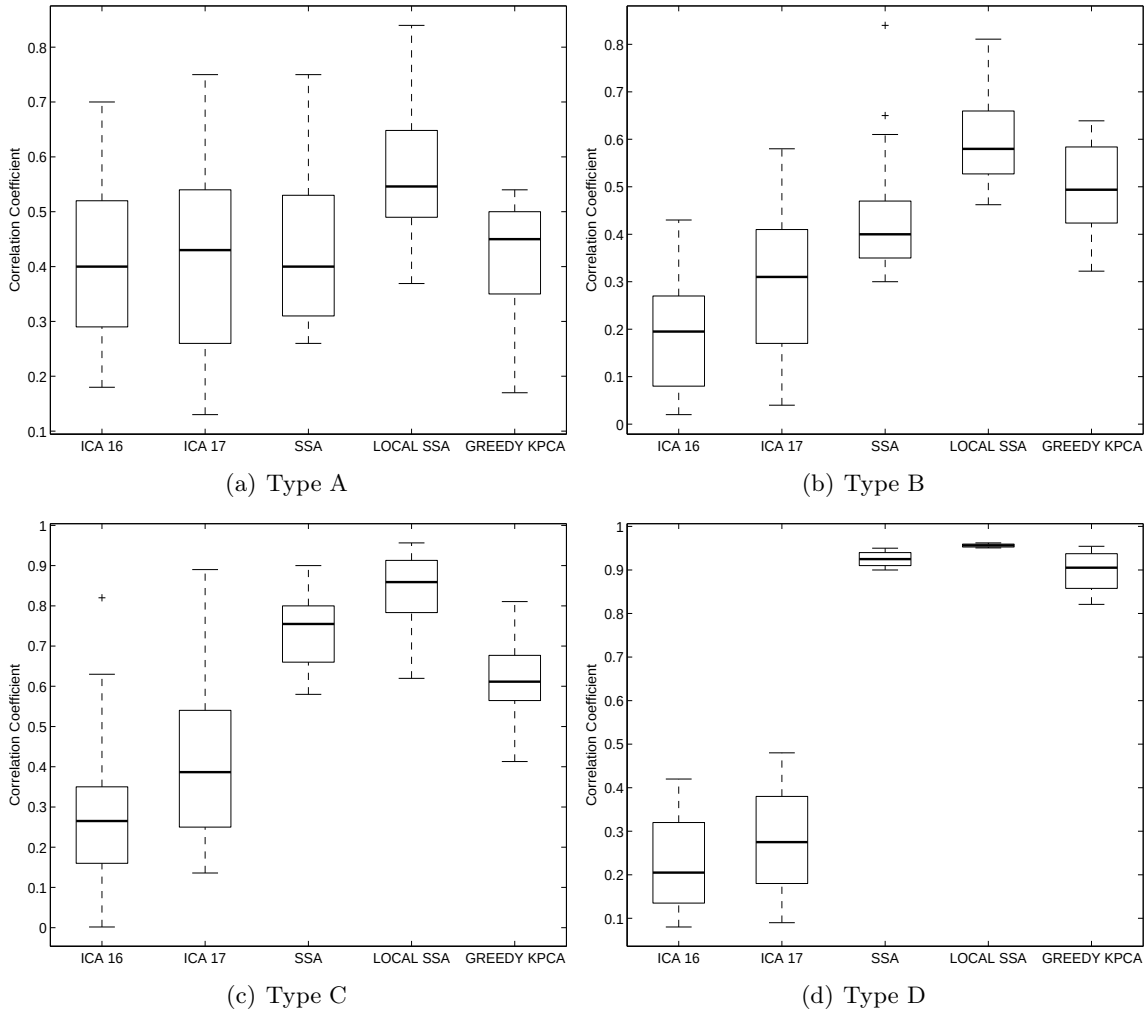
Table 5.2: Correlation coefficient between original EEG and corrected EEG. The values correspond to segments of signals represented in figure 5.4.

always a value closer to 1 and the standard deviation (across segments) is very small for all segments used (see vertical line in figure 5.6). The alpha band also has values around 0.9 in 3 cases for Local SSA (Type A, B and C) while for the other algorithms the values have a broader range. Greedy KPCA, in particular, has 0.4 for Type C segments, while other algorithms have 0.9. The values of coherence for segments Types A and B of Local SSA in delta and theta bands vary between [0.4 , 0.6] while Greedy KPCA is [0.1 , 0.4]. Table 5.3 shows the mean of the coherence in the frequency range of each characteristic band for signals of figure 5.4. The coherence values of Local SSA are high in the beta and alpha range, whatever the type of segment used in the mixture. For this method, the distortion is in the delta band and is even higher when that reference signal is of delta type. In [32], it was also reported a distortion in the frequency range $5 - 25$Hz when ICA algorithms were used, but [14] reported that the distortion was higher for ICA algorithms based on higher order statistics. In most

		Coherence			
	Type	Delta	Theta	Alpha	Beta
Local SSA	A	0.35	0.56	0.93	0.98
	B	0.56	0.65	0.89	0.95
	C	0.37	0.51	0.88	0.98
	D	0.39	0.28	0.72	0.96
ICA	A	0.21	0.52	0.32	0.10
	B	0.38	0.22	0.08	0.13
	C	0.40	0.05	0.11	0.12
	D	0.37	0.11	0.08	0.10
GKPCA	A	0.19	0.38	0.76	0.95
	B	0.12	0.41	0.78	0.94
	C	0.18	0.23	0.59	0.95
	D	0.12	0.19	0.69	0.96

Table 5.3: Coherence in frequency related to the results in figure 5.4.

cases, the SSA algorithm shows a performance similar to Local SSA. When comparing the performance of the studied algorithms, it is possible to conclude that Local SSA presents the better results. It is an autonomous, automatic and adaptive algorithm to the input signal.
When seeing the results obtained by ICA, one can verify that ICA was effective in the removal of artifacts in EEG signal. This can only happen if the same artifact is present in all channels



(a) Type A

(b) Type B

(c) Type C

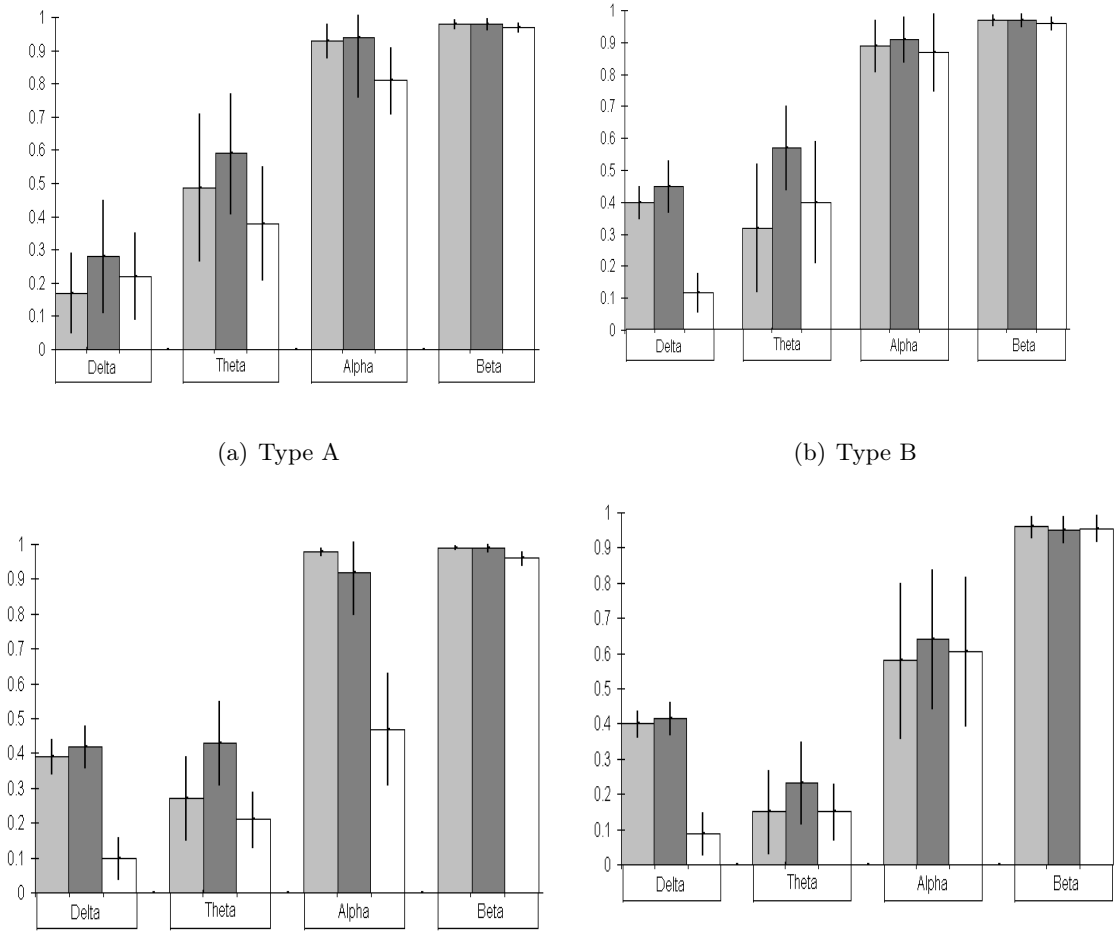
(d) Type D

Figure 5.5: Boxplots of Correlation coefficients (ct_{oy}) for ICA16, ICA17, SSA, Local SSA and Greedy KPCA algorithms.

without base line oscillations. On the other hand, ICA was not able to find the components associated to the EEG artifacts. One can conclude that Local SSA is a fully automated algorithm able to adapt the parameters to the input signal, without the user interaction. For signals with high amplitude artifacts, the denoised signal is always achieved with minimal loss. The results indicated that the performance of the algorithms is directly related to the adjacent base activity. In signals with beta and alpha activities, the artifact elimination by the algorithms, leads to a minor loss.

5.6 Preliminary Real Applications

The signals of the set of channels recorded along the monitoring session suffer from distinct forms of distortion. In particular, the high-amplitude interference arising from ocular movements are more visible in frontal channels, while electrode artifacts show up in different channels spread over the scalp.



(a) Type A

(b) Type B

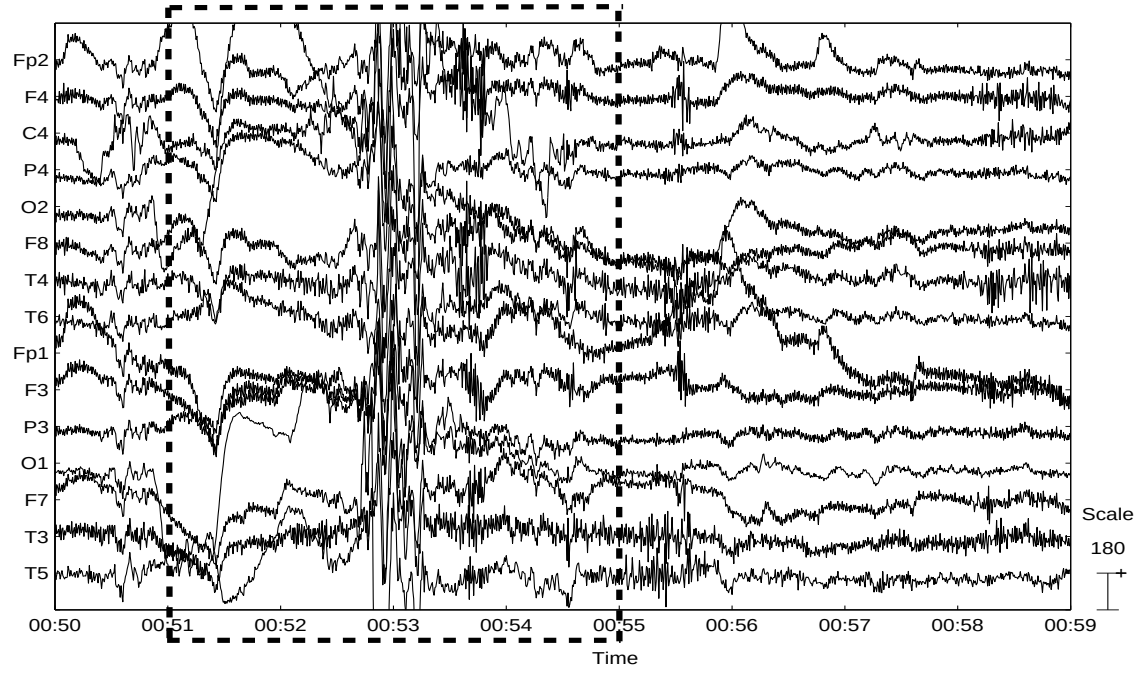
(c) Type C

(d) Type D

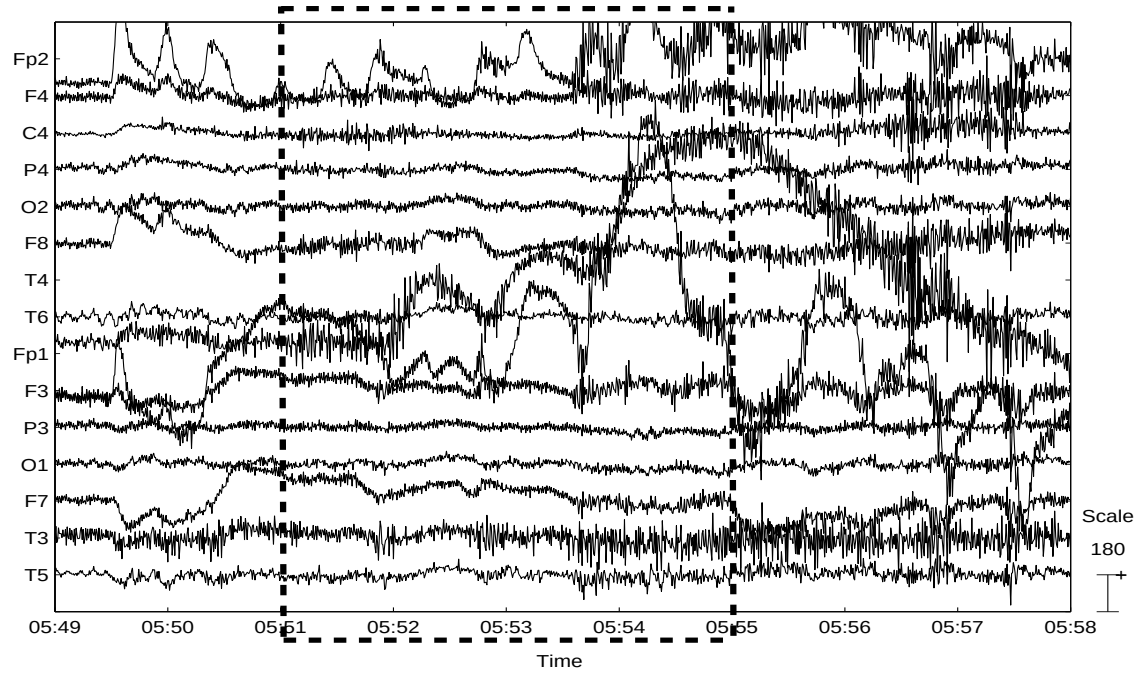
Figure 5.6: Coherence values in the different frequency bands for SSA (*gray bar*), Local SSA (*black bar*) and Greedy KPCA (*white bar*).

In this section, a preliminary real application of the two subspace techniques studied will be discussed. The results were discussed and published in [3, 2].

The EEG signals were chosen from a database of epileptic patients recorded on long-term EEG monitoring sessions. The EEG signals were recorded using 19 electrodes placed according to the $10-20$ system (common ground reference at Fz). The signals are filtered, digitalized (sampling rate-$128Hz$) and stored as European Data Format (EDF) files. Monopolar (common Cz reference) brain signals were visualized using EEGLAB [37]. Two subsegments (with $K = 1280$ samples) will be used in the analysis, figure 5.7, belonging to a patient which suffered from a partial complex seizure from the right temporal focus. The two subsegments correspond to EEG signals preceding the epileptic seizure onset and are corrupted by high-amplitude artifacts: the first segment starts 28 minutes before seizure onset and the second 24 minutes before seizure onset. The analysis is performed in parallel in different single channels



(a) Segment 1



(b) Segment 2

Figure 5.7: Segments of real EEG signal.

using Local SSA and Greedy KPCA. The analysis is performed in parallel in more than one channel using an embedding dimension of $M = 41$ for Local SSA and $M = 11$ for Greedy KPCA. Only two segments were analised with 4 seconds:

- In ***segment 1*** all channels are corrupted with base line drift artifacts, figure 5.7 (a), so

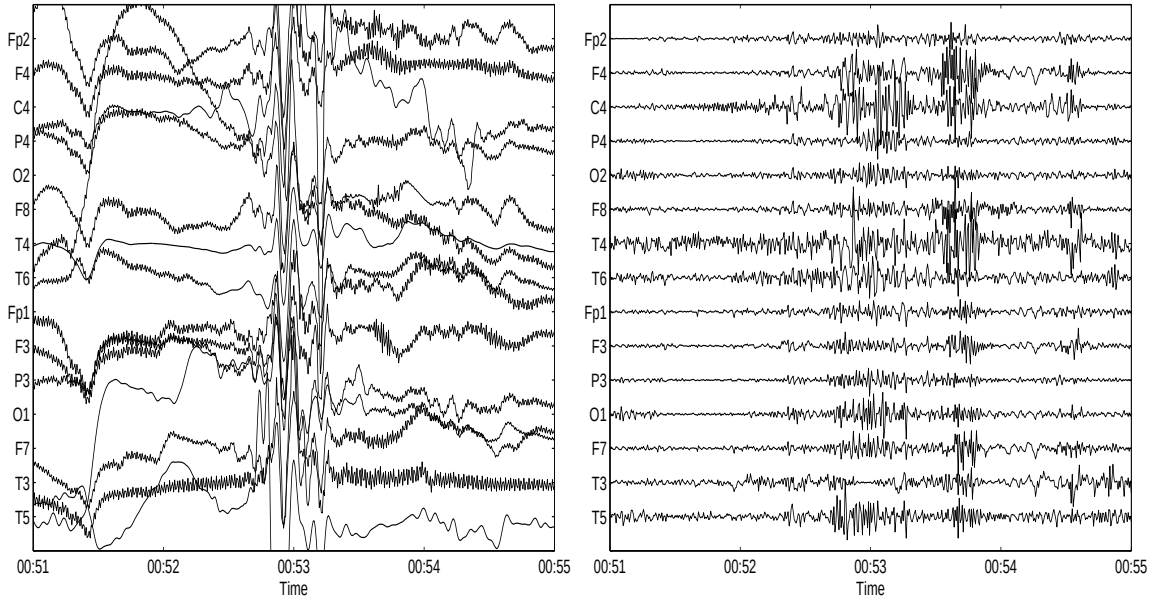
they were all processed one after the other by the algorithms, figure 5.7 (a)

- In ***segment 2*** proeminent eye movement artifacts recorded at the frontal channels are shown, figure 5.7(b). Only these frontal channels and channel $T4$, monitoring temporal cortex, were processed, figure 5.7 (b)

In the next sections the Local SSA and Greedy KPCA results will be exposed and discussed.

5.6.1 Local SSA Results

Segment 1 All channels are processed one after the other by the algorithm, figure 5.8, and it can be seen that the correct EEG, figure 5.8(a) exhibits the high-amplitude components of the original signals. In most of the channels, an instantaneous frequency analysis (spectrogram) shows that the frequency contents is mainly in the low frequency range ($< 10Hz$) and also around $50Hz$, figure 5.8 (a). The corrected EEG, figure 5.8(b), mainly possesses the high frequency ($> 10Hz$) contents of the original signal. However, in $T4$ and $T6$, bursts of theta ($3 - 7Hz$) waves can be seen (around 53s). The bursts of spikes can now also be seen in the frontal channels.



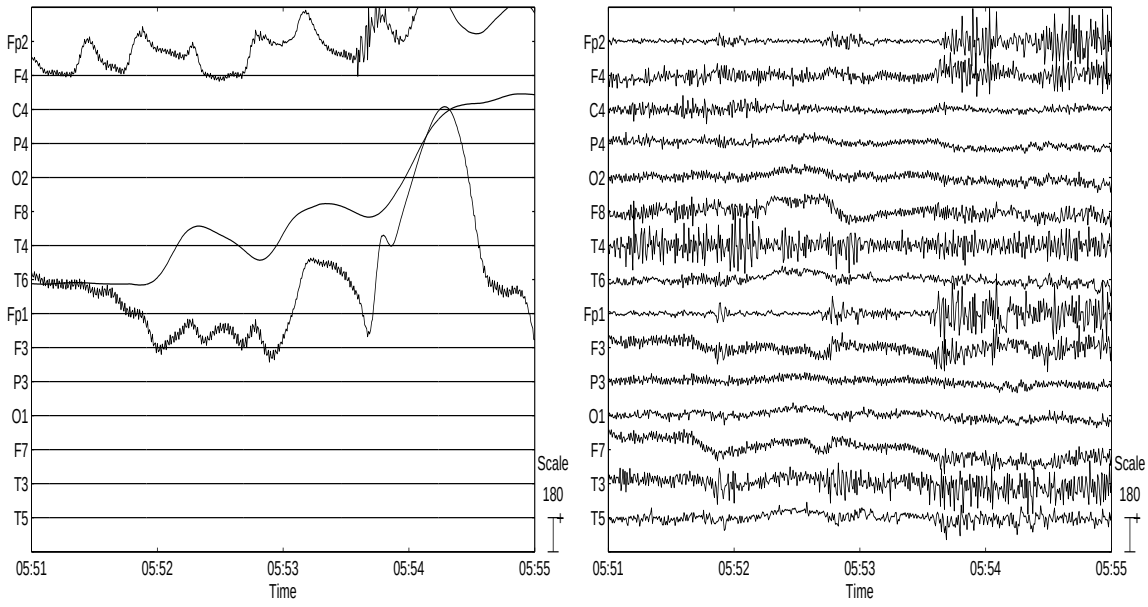
(a) Extracted EEG

(b) Corrected EEG

Figure 5.8: First Segment of EEG signals, figure 5.7 (a), processed by Local SSA.

Segment 2 The extracted signal is only related to the EOG artifact and the $50Hz$ line noise, figure 5.9 (a) and the corrected EEG has the lower amplitude components of the signal, figure 5.9 (a). In $T4$ a burst of spikes (after 5m 51s) can be seen while in other channels ($F4$ and $F8$) single spikes also occur during the same period. Comparing the corrected $T4$ channel, figure 5.9 (b) with the corresponding channel before the seizure onset, it can be verified that both exhibit bursts of spike waves [2]. The paroxysmal activity in $T4$ before the seizure initiation indicates the possible origin of the the epileptogenic focus. $T4$ channel of figure 5.9(b) reveals

an activity which is similar to the corresponding segment that precedes the seizure initiation and in such case indicates the epileptogenic focus.



(a) Extracted Artifacts

(b) Corrected EEG

Figure 5.9: Second Segment of EEG signals, figure 5.7 (b), processed by Local SSA.

5.6.2 Greedy KPCA Results

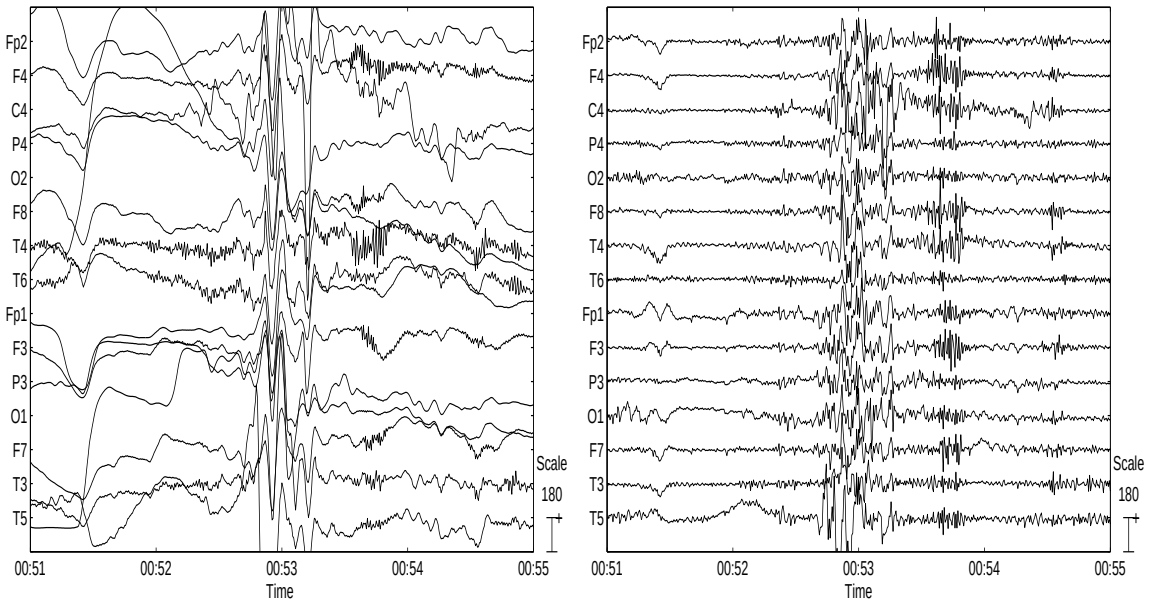
In this subsection the Greedy KPCA results will be presented. The parameters of the model are computed using the greedy approach. In the multidimensional signal, the K vectors of the training set are chosen randomly.

The number of patterns in the training subset R are chosen by the incomplete Cholesky algorithm. The number L of eigenvectors ($\mathbf{U}$) to project the data for reconstruction in the feature space is determined by the eigenspectrum of the matrix $\mathbf{Q}$ which should approximate the eigenvalues of the kernel matrix of the training set. Table 5.4 shows the range of those parameters for both data segments.

	K	R	L
Segment 1	384	[9 , 18]	[3 , 6]
Segment 2	384	[13 , 15]	4

Table 5.4: Parameters of the algorithm Greedy KPCA.

Segment 1 The corrected EEG, figure 5.10, mainly possesses the high frequency ($> 10Hz$) contents of the original signal. However, in $T4$ and $T6$, bursts of theta ($3 - 7Hz$) waves and slow sharp waves can be seen as seen in the Local SSA results, figure 5.8. Now, the bursts of spikes are clearly visible in the frontal channels.



(a) Extracted Artifacts

(b) Corrected EEG

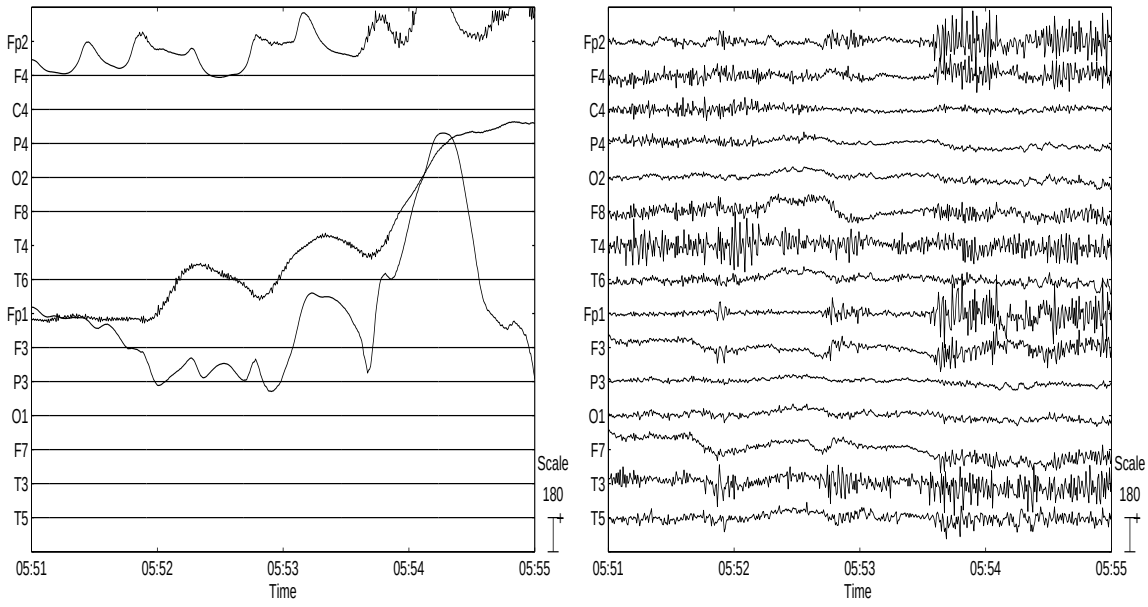
Figure 5.10: First Segment of EEG signals, figure 5.7 (a), processed by Greedy KPCA.

Segment 2 In channel $T4$, a burst of spikes can be seen, while in other channels ($F4$ and $F8$) single spikes also occur during the same period. Comparing the corrected recordings of channel $T4$ with the corresponding recording before the seizure, the pronounced burst of spike waves is more clearly seen in the corrected recording. This paroxysmal activity in $T4$ before the seizure onset indicates the possible origin of the the epileptogenic focus. The same has been seen with the Local SSA results.

5.6.3 Results Comparison

The results obtained by the Local SSA and Greedy KPCA algorithms described in the last section can now be compared. The correlation coefficient between the denoised channels is used to evaluate the algorithms performance. In both cases, the correlation coefficient of the corrected EEG ranges between [0.88 , 0.99] and the correlation coefficient of the extracted EEG ranges between [0.79 , 0.94]. The latter interval results from the fact that the Local SSA also extracts the 50Hz line interference while the Greedy KPCA does not.

Figure 5.12 compares the power spectral density of the channel Fp2 in the second segment from both methods. The results show that the low frequency content of the corrected EEG is differently affected by the two methods. The Greedy KPCA seems to preserve more spectral information in the very low frequency regime ($f \leq 3Hz$) but yields similar results in higher frequencies, with the exception of $50Hz$.



(a) Extracted Artifacts

(b) Corrected EEG

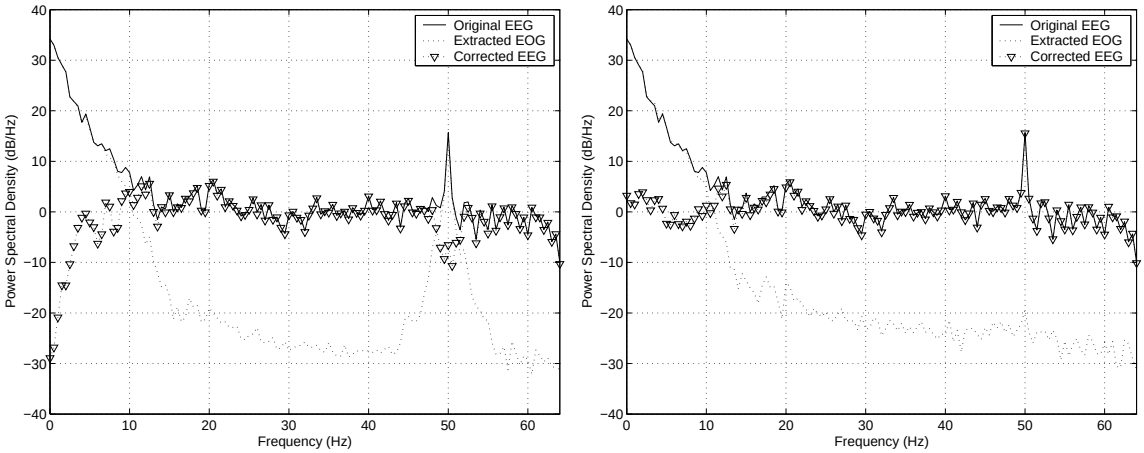
Figure 5.11: Second Segment of EEG signals, figure 5.7 (b), processed by Greedy KPCA

5.7 Conclusion

Artifact reduction in EEG recordings [38] is a very important problem that needs to be addressed in a systematic way. Such artifacts can often be found but only in certain channel recordings, like EOG artifacts in frontal channel recordings. Hence, it is of practical interest to be able to efficiently remove such artifacts from single channel recordings without the need to do a full multichannel analysis. A study was done to prove the feasibility of projective subspace techniques in addressing the problem using a single-channel approach. Projections either into input space or into a high-dimensional feature space were considered, after a non-linear mapping of the data from input space to feature space.

Results with artificial mixtures The numerical simulations using artificially mixed data revealed that both techniques were feasible to remove the high-amplitude ocular movements and blinks. Local SSA showed a better performance as the corrected EEG exhibited less distortion in all frequency bands. However, it was shown that both approaches had similar performance in what concerned frequency distortions in the frequency range of beta and alpha bands. The frontal EEG signal analysis confirmed these results, [3, 39]. The algorithms were also applied to the corrupted EEG's segments due to patients movements artifact. The results obtained corroborated the results exposed.

Preliminary real Results The methods studied need the information contained within a single channel only therefore, they can be applied to each channel separately. Thus, only channels which contain such artifacts need to be processed. The results confirmed that Local



(a) Local SSA (b) Greedy KPCA

Figure 5.12: Power spectral densities (psd) resulting from Local SSA and Greedy KPCA, using the channel Fp2 of the second segment analyzed.

SSA and Greedy KPCA showed good performance in removing artifacts like eye or head movements, baseline drifts and line noise. In summary, with the methods proposed it is possible to separate EEG signal recordings into two components: artifacts and undistorted EEG. It should be pointed out that the algorithms do not require any user intervention to select the components of the reconstruction, as conventional ICA methods do for example. Furthermore, the user can choose to process a subset of channels keeping others unprocessed which will also allow a comparison of the outcomes of the algorithm with non-processed channels. Both algorithms (Local SSA and Greedy KPCA) were incorporated in the EEGLAB [37] environment. This open-software tool based on MATLAB offers visualization facilities that will allow the accomplishment of clinical evaluation tasks.

Bibliography

[1] A. R. Teixeira, A. M. Tomé, E.W.Lang, P. Gruber, and A. M. d. Silva, "Extraction and separation of high-amplitude artifacts in electroencephalograms from epileptic patients," in *Fourth IASTED International Conference on Biomedical Engineering- BIOMED2006* (C. Ruggiero, ed.), (Innsbruck, Austria), pp. 270–275, IASTED, 2006.

[2] A. R. Teixeira, A. M. Tomé, E. Lang, P. Gruber, and A. M. Silva, "Automatic removal of high-amplitude artifacts from single-channnel electroencephalograms," *Computer Methods and Programs in Biomedicine*, vol. 83, no. 2, pp. 125–138, 2006.

[3] A. M. Tomé, A.R.Teixeira, E.W.Lang, and A. M. d. Silva, "Greedy KPCA applied to single- channel EEG recordings," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2007)*, (Lyon), 2007.

[4] A. R. Teixeira, A. M. Tomé, M. Böhm, C. G. Puntonet, and E. W. Lang, "How to apply non-linear subspace techniques to univariate biomedical time series," *IEEE Transactions on Instrumentation & Measurement*, vol. 58, no. 8, pp. 2433 – 2443, 2009.

[5] A. R. Teixeira, A. M. Tomé, E. W. Lang, and A. M. d. Silva, "Subspace techniques to remove artifacts from EEG: a quantitative analysis," in *30th Annual International IEEE EMBS Conference* (IEEE, ed.), (Vancouveur), pp. 4395–4398, 2008.

[6] A. R. Teixeira, A. M. Tomé, K. Stadlthanner, and E. W. Lang, "KPCA denoising and the pre-image problem revisited," *Digital Signal Processing*, vol. 18, pp. 568–590, 2008.

[7] G. Gratton, "Dealing with artifacts: The EOG contamination of the event-related brain potential," *Behavior Research Methods, Instruments,& Computers*, vol. 30, no. 1, pp. 44–53, 1998.

[8] G. Gratton, M. Coles, and E. Donchin, "A new method for offline removal of ocular artifacts," *Electroencephalogr Clin Neurophysioly*, vol. 55, pp. 468–484, 1983.

[9] J. Woestenburg, M. Verbaten, and J. Slanger, "The removal of the eye-movement artifact from the EEG by regression analysis in the frequency domain," *Biol Psychol*, vol. 16, pp. 127–147, 1983.

[10] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, and G. Pfurtscheller, "A fully automated correction method of EOG artifacts in EEG recordings.," *Clinical Neurophysiology*, vol. 118, no. 1, pp. 98–104, 2007.

[11] A. Schlögl and B. C., "BioSig: A Free and Open Source Software Library for BCI research," *Computer*, vol. 41, pp. 44–50, 2008.

[12] P. He, G. Wilson, and C. Russell, "Removal of ocular artifacts from electro-encephalogram by adaptive filtering," *Medical and Biological Engineering and Computing*, vol. 42, no. 3, pp. 407–412, 2004.

[13] P. Kumar, R. Arumuganathan, K. Sivakumar, and C. Vimal, "An Adaptive method to remove ocular artifacts from EEG signals using Wavelet Transform," *Journal of Applied Sciences Research*, vol. 5, no. 7, pp. 741–745, 2009.

[14] S. Romero, M. Mananas, and M. Barvanoj, "Ocular reduction in EEG signals based on Adaptive Filtering, Regression and Blind Source Separation," *Annals of Biomedical Engineering*, vol. 37, no. 1, pp. 176–191, 2009.

[15] A. Garcés Correa, E. Laciar, H. D. Patiño, and M. E. Valentinuzzi, "Artifact removal from EEG signals using adaptive filters in cascade," *Journal of Physics: Conference Series*, vol. 90, no. 1, 2007.

[16] T. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. Sejnowski, "Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects," *Clinical Neurophysiology*, vol. 111, no. 10, pp. 1745–1758, 2000.

[17] J. Iriarte, E. Urrestarazu, M. Valencia, M. Alegre, A. Malanda, C. Viteri, and J. Artieda, "Independent component analysis as a tool to eliminate artifacts in EEG: a quantitative study," *Journal of clinical neurophysiology*, vol. 20, no. 4, pp. 249–257, 2003.

[18] P. He, G. Wilson, C. Russell, and M. Gerschutz, "Removal of ocular artifacts from the EEG: a comparison between time-domain regression method and adaptive filtering method using simulated data," *Med Biol Eng Comput*, vol. 45, no. 5, pp. 495–503, 2007.

[19] N. Ille, P. Berg, and M. Scherg, "Artifact correction of the ongoing EEG using spatial filters based on artifact and brain signal topographies.," *Journal of Clinical Neurophysiology*, vol. 19, no. 2, pp. 113–124, 2002.

[20] C. Gouy-Pailler, R. Sameni, M. Congedo, and C. Jutten, *Iterative Subspace Decomposition for Ocular Artifact Removal from EEG Recordings*, vol. 5441. Springer Berlin / Heidelberg, 2009.

[21] S. Casarottoa, A. M. Bianchia, S. Ceruttia, and G. A. Chiarenzab, "Principal Component Analysis for reduction of ocular artefacts in event-related potentials of normal and dyslexic children," *Clinical Neurophysiology*, vol. 115, pp. 609–619, 2004.

[22] P. Sadasivan and D. N. Dutt, "SVD based technique for noise reduction in electroencephalographic signals," *Signal Processing*, vol. 55, pp. 179–189, 1996.

[23] G. Gomez-Herrero, W. Clercq, H. Anwar, O. Kara, K. Egiazarian, V. Huffel, and W. V. Paesschen, "Automatic removal of ocular artifacts in the EEG without an EOG reference channel," in *7th IEEE Nordic Signal Processing Symposium*, (Reykjavik, Iceland), IEEE, 2006.

[24] C. Joyce, I. Gorodnitsky, and M. Kutas, "Automatic removal of eye movement and blink artifacts from EEG data using blind component separation," *Psychophysiology*, vol. 41, no. 2, pp. 313–325, 2004.

[25] A. Delorme, T. Sejnowsky, and S. Makeig, "Enhanced Detection of Artifacts in EEG data using Higher-Order Statistics and Independent Component Analysis," *NeuroImage*, vol. 34, pp. 1443–1449, 2007.

[26] T. Jung, S. Makeig, C. Humphries, T. Lee, M. Mckeown, V. Iragui, and T. Sejnowski, "Removing electroencephalografic artifacts by blind source separation," *Psychophysiology*, vol. 37, pp. 163–178, 2000.

[27] E. Urrestarazu, J. Iriarte, M. Alegre, M. Valencia, C. Viteri, and J. Artieda, "Independent Component Analysis Removing Artifacts in Ictal Recordings," *Epilepsia*, vol. 45, no. 9, pp. 1071–1078, 2004.

[28] W. Zhou and J. Gotman, "Removing Eye-movement artifacts from the EEG during the Intracarotid Amobarbital Procedure," *Epilesia*, vol. 46, no. 3, pp. 409–414, 2005.

[29] R. Vigário, "Extraction of ocular artefacts from EEG using Independent Component Analysis," *Electroencephalogram Clinical Neurophysioly*, vol. 103, no. 3, pp. 395–404, 1997.

[30] L. Vigon, M. Saatchi, J. Mayhew, and R. Fernandes, "Quantitative evaluation of techniques for ocular artefact filtering of EEG waveforms," *IEE Proc. Science Measurement and Technology*, vol. 147, no. 5, pp. 219–228, 2000.

[31] F. Viola, J. Thorne, B. Edmonds, T. Schneider, T. Eichele, and S. Debener, "Semi-Automatic Identification of Independent Components Representing EEG Artifact," *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, vol. 120, no. 5, pp. 868–877, 2009.

[32] G. L. Wallstrom, R. E. Kass, A. Miller, J. C. Cohn, and N. A. Fox, "Automatic correction of ocular artifacts in the EEG: a comparison of regression-based and component-based methods," *International Journal of Psychophysiology*, vol. 53, pp. 105–119, 2004.

[33] E. Niedermeyer and F. Lopes Da Silva, *Electroencephalography: Basic Principles, Clinical Applications and Related Fields*. Wiley-Blackwell, 2004.

[34] C. W. Anderson, J. N. Knight, T. O'Connor, M. J. Kirby, and A. Sokolov, "Geometric subspace methods and time-delay embedding for EEG artifact removal and classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 142–146, 2006.

[35] S. Makeig, S. Debener, J. Onton, and A. Delorme, "EEGLAB," 2007.

[36] A. Delorma and S. Makeig, "EEGLAB: An Open Source Toolbox for Analysis of Single-Trial EEG Dynamics Including Independent Component Analysis," *Journal of Neuroscience Methods*, vol. 134, pp. 9 – 21, 2004.

[37] A. Delorme and S. Makeig, "EEGLAB: An Open Source Toolbox for Analysis of Single-Trial EEG Dynamics," *Journal of Neuroscience Methods*, vol. 134, pp. 9–21, 2004.

[38] R. J. Croft and R. J. Barry, "Removal of ocular artifact from the EEG: a review," *Neurophysiol. Clin.*, vol. 30, pp. 5–19, 2000.

[39] A. R. Teixeira, A. M. Tomé, E. Lang, R. Schachtner, and K. Stadlthanner, "On the use of KPCA to extract artifacts in one-dimensional biomedical signals," in *Machine Learning for Signal Porcessing, MLSP 2006* (S. McLoone, J. Larsen, M. V. Hulle, A. Rogers, and S. C. Douglas, eds.), (Dublin), pp. 385–390, IEEE, 2006.

Chapter 6

Feature Extraction

"Equations are more important to me, because politics is for the present, but an equation is something for eternity."
- Albert Einstein -

Contents

Unsupervised feature extraction methods try to generate representative features from the raw data, thus helping the classifier to learn a more robust solution and to achieve a better generalization performance. Often not all original features are appropriate, and even the number of features might be too large to conduct an efficient training. Subspace techniques can be applied as unsupervised feature generators, simultaneously providing dimension reduction and more suitable representations.

Principal Component Analysis is a subspace technique widely used in many fields of research like face recognition [1] and related computer vision tasks [2]. In this application, a new representation of a given dataset is formed by a linear combination of the original features whereby the data is projected onto orthogonal basis vectors. The goal of this transformation is to retain, in the new representation, most of the energy of the raw data. The new

representation is uncorrelated and can even be of smaller dimension. This model also implies that the original features are linear combinations of these projections. This assumption is a limitation, if the purpose of it all is to model highly complex data. Kernel PCA methods are well suited in such cases to find the nonlinear principal components. In a classification task, the new representation provided by the non-linear kernel method, belongs to a new space where the data will, most probably, become linearly separable [3]. The majority of the linear subspace methods have been adapted to include a non-linear transformation of the data and several computer vision applications incorporate these techniques [4, 5, 6].
In this chapter, the linear and non-linear projective techniques described in chapters 3 and 4 are discussed to perform the feature extraction.

This chapter is organized as follows: section 6.1 resumes the feature extraction and the classification methods used in this work. The numerical simulations compare the performance of classifiers using kernel features, principal component features and a direct classification of the raw data using two classifiers: the nearest neighbor (NN) and linear discriminant function (RL). Furthermore, to evaluate the impact of the projective techniques, a comparative study with the best results published in [7] is presented and discussed, section 6.2. The greedy KPCA is used with a large dataset (USPS dataset of handwritten digits), which is often used as a benchmark test dataset, section 6.3. The conclusions are presented in the last section.
Note that all results present in this chapter are published already in [8, 9].

6.1 Feature Extraction and Classification

Classification algorithms assume that the objects to be labeled (classified) are points in a multidimensional space. However, before executing a training algorithm thus adapting the parameters of the classifier, an additional transformation may be applied to the original vectorial data (the objects). In the literature, both the initial raw data vectors and the transformed vectors are dealt with under the name of feature extraction.
The subspace models were considered and in particular the kernel version of principal component analysis as performing feature extraction on the raw data vectors. The classification scheme that uses subspace methods as a pre-processing step is represented in figure 6.1.

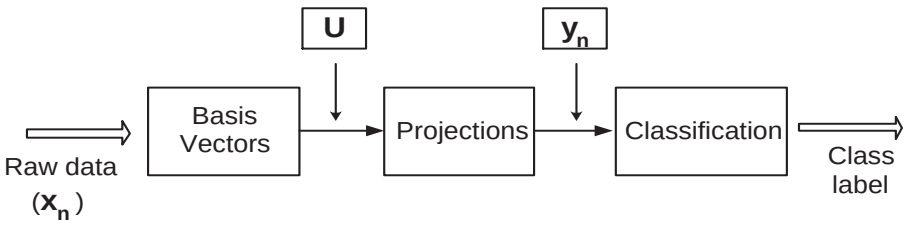


Figure 6.1: Computing features ($\mathbf{y}_n$) using subspace techniques.

The system is composed of two parts:

1. The subspace model
   The subspace model is described by matrix $\mathbf{U}$ whose L columns are the basis vectors of the new representation chosen to project the input data ($\mathbf{x}_n$) onto. Hence, the projections ($\mathbf{y}_n$) constitute the new representation of the data. These projections result either as a simple linear combination of the input data, like with PCA projections, or they represent non-linear components of the data, like with KPCA projections.
   The following table 6.1 summarizes the projections considering the linear and non-linear subspace techniques reported in chapter 3 and 4, respectively.

Algorithm	Projections - $\mathbf{Y}$ (Training Set)
PCA	$\mathbf{U}^T\mathbf{X}$, eqn. 2.7
KPCA	$\mathbf{U}^T\mathbf{\Phi}$, eqn. 4.6
Greedy KPCA	$\mathbf{V}_q^T\mathbf{L}^{-1}\mathbf{\Phi}_R^T\mathbf{\Phi}$, eqn. 4.41

Table 6.1: Resume of all the projections in input and feature space.

   Using the information of table 6.1, it is easily verified that all the projections in input and feature space of the training set can be written as

$$\mathbf{Y} = \mathbf{D}^{1/2}\mathbf{V}^T$$

   The new representations of the training data set ($\mathbf{Y}$) are non-correlated, i.e. $\mathbf{Y}\mathbf{Y}^T$ is a diagonal matrix.

2. The classification
   In a classification task, the projections ($\mathbf{Y}$) then form the input to the classifier. During the training phase, the basis vectors ($\mathbf{U}$) are computed and the projections and corresponding labels are used to adapt the parameters of the classifiers. After learning the subspace model, the test dataset, $\mathbf{\Phi}_{test}$ is also projected onto the basis vectors $\mathbf{U}$,

$$\mathbf{Y}_{test} = \mathbf{U}^T\mathbf{\Phi}_{test}$$

   thus forming the new representation of the test data, $\mathbf{Y}_{test}$. These projections build the input to the classifier and the corresponding outputs, i.e. the classified test data is used to evaluate the performance of both the feature extraction and classification methods.

In this work two classifiers were considered: one-nearest neighbor (NN) and the linear discriminant function (RL). With an NN classifier, each element of the test set is classified according to the nearest neighbor in the training set. In the case of linear discriminant functions, the

weight vectors are computed within the training dataset by using the mean-square-error criterion [10]. Each element in the test set is assigned to the class whose discriminant function has the largest value. The effectiveness of the subspace feature extraction methods discussed above is evaluated by comparing the performance of classifiers. The generalization errors are used to illustrate the influence of the data projection into the different models. For that, experiments were carried out on artificial and real world datasets available on public repositories. Two groups of data were considered: one constituted by thirteen benchmarks, section A.1.3 and the USPS dataset, section A.1.2.

6.2 Dataset Analysis

The benchmarks description was done in appendix A.1.3. Several state-of-the-art algorithms have already been applied to those datasets, among which: SVM, KPCA, Adboost, KFD [7, 11, 12].
The main issue of this study is to compare the algorithms performance with the results present in the literature. The averages and standard deviation of the error rates are presented in the next tables for input and feature space analysis. Therefore, using the training and test sets, the algorithms can be compared. The difference between error percentages achieved by any algorithm can be evaluated statistically. The t-test was performed with 95% significance, considering

$$H0 : \mu_1 = \mu_2$$
$$H1 : \mu_1 \neq \mu_2$$

where μ_1 and μ_2 represent the test error average achieved by two methods. So a statistical test has been carried out in an attempt to reject $\ominus$ or accept $\oplus$ the null hypothesis ($H0$).

6.2.1 Input Space

The PCA algorithm was applied to each dataset. The projections were computed in all possible directions D, on each dataset, and the optimal number L was selected to yield the lowest error rate. Table 6.2 summarizes the average test error results in input classifications obtained by RL and NN classifiers. The four first columns show the classification results on the raw dataset (RD) and on PCA projections. In the two remaining columns, the t-test results between the results of [7]/Projections and Raw data/Projections are present. The results of the classification allow the organization of the datasets into two groups, table 6.2.
In group 1, comprising the first 8 datasets, at least one of the classifiers achieves an error rate comparable to the ones published in [7]. In group 2, encompassing the remaining five datasets, the performance was far from the results presented in [7]. The linear discriminant function classifier achieves the best performance in group 1 with the exception of the *Thyroid* dataset. In the second group, the best result is splitted between the two classifiers. A t-test is used to compare the best result (either from NN or RL classifier) with the ones published. The $H0$ is rejected in the second group and accepted in the first group, except in two datasets

	Raw Data		Projections		t-test	
Classifiers	**NN**	**RL**	**NN**	**RL**	I1	I2
B. Cancer	32.5 ± 4.8	26.9 ± 2.7	32.5 ± 4.8	26.2 ± 2.3	$\oplus$	$\oplus$
Diabetis	30.1 ± 2.0	23.4 ± 1.7	30.1 ± 2.0	$\mathbf{23.4 \pm 1.7}$	$\oplus$	$\oplus$
German	29.4 ± 2.4	24.3 ± 2.9	29.4 ± 2.4	23.9 ± 2.1	$\oplus$	$\oplus$
Heart	23.2 ± 3.7	$\mathbf{15.8 \pm 3.1}$	22.0 ± 3.1	15.9 ± 3.1	$\oplus$	$\oplus$
F. Solar	39.0 ± 4.9	33.5 ± 1.5	39.0 ± 4.9	32.9 ± 1.8	$\oplus$	$\ominus$
Thyroid	4.3 ± 2.2	14.7 ± 3.1	$\mathbf{3.9 \pm 2.1}$	14.7 ± 3.2	$\ominus$	$\oplus$
Titanic	33.0 ± 11.0	22.6 ± 1.0	32.9 ± 11.0	22.6 ± 1.3	$\oplus$	$\oplus$
Twonorm	6.6 ± 0.7	2.6 ± 0.17	3.4 ± 0.5	$\mathbf{2.3 \pm 0.1}$	$\ominus$	$\ominus$
Image	33.0 ± 5.4	16.5 ± 0.9	33.0 ± 0.5	16.5 ± 0.98	$\ominus$	$\oplus$
Ringnorm	35.1 ± 1.3	24.7 ± 0.7	21.3 ± 1.2	24.6 ± 0.7	$\ominus$	$\ominus$
Splice	28.8 ± 1.5	16.2 ± 0.6	22.4 ± 1.4	16.31 ± 0.6	$\ominus$	$\oplus$
Waveform	15.8 ± 0.6	14.8 ± 0.2	11.7 ± 0.7	12.6 ± 0.7	$\ominus$	$\ominus$
Banana	13.6 ± 7.0	46.98 ± 7.0	13.6 ± 7.0	46.9 ± 7.0	$\ominus$	$\oplus$

Table 6.2: Comparison of the error rate classification in input space using three methods on 13 benchmarks. The columns I1 and I2 represent the results of a significant t-test (95%) between Best/Projections and Raw data/Projections, where $\oplus$ accepts $H0$ and $\ominus$ rejects $H0$.

(*Twonorm* and *Thyroid*), table 6.2, column I1. Note, that *Twonorm* dataset, always presents better results in input space with RL classifier than the results present in [7]. In the case of *F. Solar* database, t-test shows that a raw data classification has a significant error when compared to a PCA projection classification, column I2. Input classification for subset 2 is worse than [7], and null hypothesis, $H0$, is always rejected, column I1.

6.2.2 Feature Space

The improvement on the linear classifier with kernel projections is to be expected for the last datasets if, in feature space, the data is linearly separable. In feature space the number of basis vectors that can be computed is equal to the number of training examples N. Both versions of KPCA are used to compute the model: the first depends on the full training set (KPCA) and the second depends on a subset with R elements of the training set, greedy KPCA. In both cases, the number of projections was varied from $L = 1$ up to R. The σ parameter used is according to eqn. 4.52. In table 6.3, the average error rates and standard deviation are shown. Column I1 shows the result of the t-test between the results of [7] and either the NN or the RL classifier. It is possible to verify that in group 1 there is no significant improvement in the performance of the classifiers using the non-linear features of the datasets, except in the *German* dataset. In most cases, the minimum error rate is achieved using a number of projections higher than the dimension (D) of the raw data ($L > D$). One of the exceptions is the *Twonorm*, where $L = 1$, but using a PCA model with this dataset, the performance is similar if the data is projected on the leading eigenvector. For group 2, work in feature space allowed better results than in input space. The $H0$ hypothesis is accepted for all datasets, table 6.3 column I1. Another t-test was performed to see how the number of projections affects the results. The error rates were compared to the error rates achieved when the number of

			KPCA				t-test	
	D	N	L	NN	L	RL	I1	I2
B. Cancer	9	200	7	32.5 ± 4.8	21	25.2 ± 4.5	$\oplus$	$\oplus$
Diabetis	8	468	17	25.3 ± 1.8	10	23.2 ± 1.6	$\oplus$	$\oplus$
German	20	300	12	30.0 ± 2.5	12	23.3 ± 2.1	$\ominus$	$\oplus$
Heart	13	170	8	22.7 ± 3.4	12	15.8 ± 3.0	$\oplus$	$\oplus$
F. Solar	9	400	55	32.2 ± 0.5	25	32.1 ± 0.6	$\oplus$	$\oplus$
Thyroid	5	140	6	4.0 ± 2.2	15	5.8 ± 2.4	$\oplus$	$\oplus$
Titanic	3	150	9	32.3 ± 1.1	10	22.3 ± 1.0	$\oplus$	$\oplus$
Twonorm	20	400	1	3.4 ± 0.4	1	2.3 ± 10.1	$\oplus$	$\oplus$
Image	18	1010	23	2.8 ± 0.6	75	7.9 ± 1.3	$\oplus$	$\ominus$
Ringnorm	20	400	40	3.5 ± 0.4	25	1.6 ± 0.1	$\oplus$	$\ominus$
Splice	60	1000	600	7.5 ± 2.6	720	4.3 ± 2.1	$\oplus$	$\ominus$
Waveform	21	400	29	9.7 ± 0.7	2	12.0 ± 0.8	$\oplus$	$\ominus$
Banana	2	400	5	13.6 ± 0.4	34	10.7 ± 0.4	$\oplus$	$\ominus$

Table 6.3: Test Error rate classification using KPCA on 13 benches. The column I1, I2 represents the results of a significant t-test (95%) between Best/KPCA and KPCA$_D$/KPCA respectively, where $\oplus$ accepts $H0$ and $\ominus$ rejects $H0$.

projections is $L = D$. Column I2 of table 6.3 shows the result and it can be verified that the H0 hypothesis is rejected in group 2 and accepted in group 1. Furthermore, notice that with the linear discriminant function, the best performance is achieved with $L > D$, with the exception of the *Waveform* set. The same analysis was done using the Greedy KPCA algorithm. Table 6.4 shows the performance of both classifiers employing greedy KPCA to extract the features and to compute the projections of the data onto the feature space coordinates. The column R shows the range of values for the size of the subset Φ_R in the training sets. The average error rate obtained is similar to the ones computed with KPCA. Moreover, the null hypothesis is accepted for every dataset, which represents the similarity with the results obtained in [7], table 6.4 column I1. The number of projections used in both methods, however, was not always identical. But considering that greedy KPCA uses an approximation of the kernel matrix, some variation had to be expected.

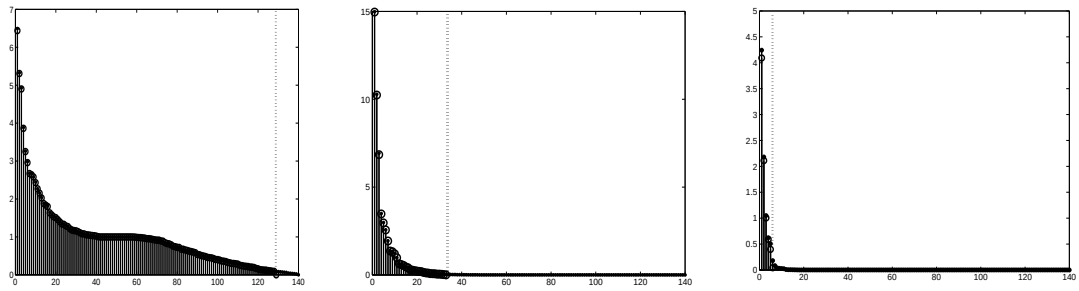
RBF Parameter

Several simulations were conducted to evaluate the suitability of the σ parameter for classification. To select the σ parameter some points must be taken into account. Dimensionality reduction can be used in the classification, however it is important to keep a sufficient number of features representative of the dataset. If the sigma parameter is not properly adjusted to the dataset, it is clearly useless to use a Greedy KPCA to reduce the dimensionality of the problem. In the literature, this parameter is often estimated by attempts using the cross validation of the first five realizations of each dataset [7]. In this experiment three values for σ RBF were used to understand their influence in the decay of the matrix eigenvalues in KPCA and Greedy KPCA algorithms: max, min and mean of the square distance of the centered points in input space.

			Greedy KPCA				t-test
	N	R	L	NN	R	RL	I1
B. Cancer	200	[89, 100]	7	32.5 ± 4.8	22	25.2 ± 4.5	$\oplus$
Diabetis	468	[141, 155]	61	30.2 ± 1.9	10	23.1 ± 1.6	$\oplus$
German	700	[412, 428]	13	29.1 ± 2.4	12	23.4 ± 2.3	$\oplus$
Heart	170	[116, 124]	48	22.8 ± 2.9	11	15.8 ± 3.1	$\oplus$
F. Solar	600	[65, 83]	65	36.8 ± 0.7	48	33.8 ± 0.6	$\oplus$
Thyroid	140	[26, 34]	6	31.2 ± 1.4	25	21.8 ± 1.0	$\oplus$
Titanic	150	[6, 9]	6	7.1 ± 2.4	6	4.2 ± 2.5	$\oplus$
Twonorm	400	[320, 325]	1	3.5 ± 0.6	1	2.3 ± 0.1	$\oplus$
Image	1010	[105, 123]	21	2.9 ± 0.7	80	8.1 ± 1.2	$\oplus$
Ringnorm	400	[264, 275]	45	3.8 ± 0.4	31	1.7 ± 0.1	$\oplus$
Splice	1000	[871, 889]	620	7.7 ± 2.6	764	4.4 ± 2.1	$\oplus$
Waveform	400	[285, 299]	30	9.8 ± 0.3	2	12.0 ± 0.7	$\oplus$
Banana	400	[13, 16]	15	13.6 ± 0.7	5	10.8 ± 1.8	$\oplus$

Table 6.4: Error rate classification using Greedy KPCA on 13 benches. Column I1 represents the results of a significant t-test (95%) between Greedy/KPCA, where $\oplus$ accepts $H0$ and $\ominus$ rejects $H0$.

Figure 6.2 shows an example of the relation between the σ parameter and the eigenspectrum of KPCA and greedy KPCA eigendecomposition for the *thyroid* datase. If the σ value is high



(a) Min (b) Mean (c) Max

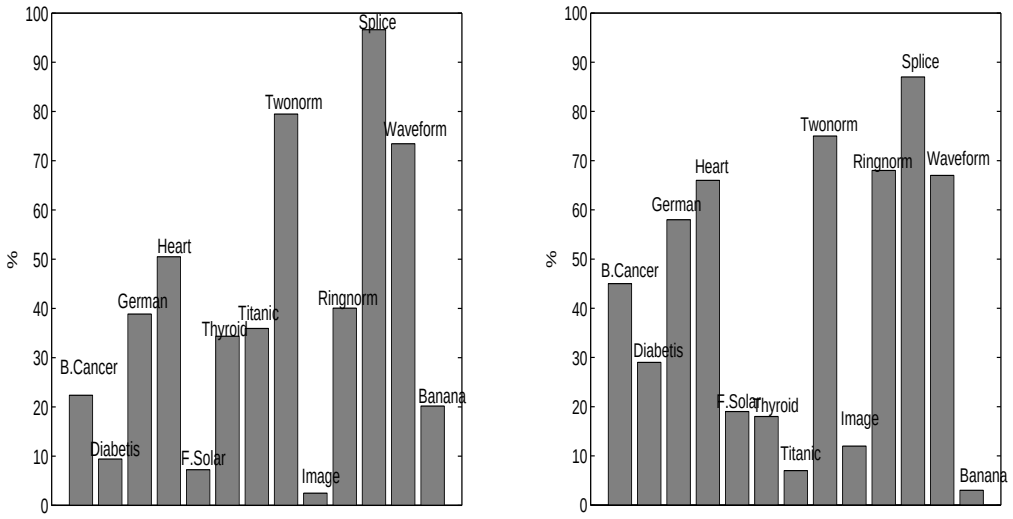
Figure 6.2: Eigenvalues of the kernel matrix of the training set *Thyroid* using the Greedy KPCA (o) and KPCA (.). The σ parameter took the following values: min, mean and max of the square distance of the centered points in the input space.

(max), the decay is abrupt and the number of non null directions is not sufficient to classify the dataset. On the other hand, to a small σ value (min) the decay is too slow and it is necessary to select a higher number of directions. Using the mean as a standard value, eqn. 4.52, it is possible to classify the dataset with a minor error. Moreover, it is always possible to apply the Greedy algorithm, thereby reducing the cost in time and size of the problem.

Greedy KPCA and KPCA - Performance Evaluation

The performance of Greedy KPCA algorithm was evaluated, and it has shown a level of performance comparable to KPCA algorithms, always with reduced classification time, figure

6.3 (a) and with a reduced size of the training set figure 6.3 (b). Figure 6.3 (a) and (b) show the ratio of the processing time and the percentage of the training set size R/N, respectively, between Greedy KPCA and KPCA, in the datasets classification. Greedy KPCA processing time for the majority of the benches is %50 less than KPCA, figure 6.3 (a). The same conclusion can be extended to the training dataset size. The larger reduction occurs mostly



(a) Time

(b) Size

Figure 6.3: Ratio of the processing time (a) and the percentage of the training set size (b) between Greedy KPCA and KPCA, in the datasets classification.

with data with lower dimension like *Banana*, *Titanic* and *Thyroid*. But this is another aspect that naturally influences the decay of eigenvalues of the kernel matrix. The normalized cumulative sum of the decreasing order eigenvalues (eqn. 6.1) can be used to show the difference on the decay of the eigenvalues of the kernel matrix.

$$S(p) = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{6.1}$$

Fig. 6.4 shows normalized cumulative sum $S(p)$ of the datasets of group 2. It can be seen that an abrupt increase on $S(p)$ corresponds to a small value for R, and if $S(p)$ converges slowly to the maximum value, then R has an higher value. The different decay profiles translate into different values for R/N. The Greedy KPCA performance advantage is emphasized because it is often desirable to minimize the complexity of the problem in practical applications. To summarize, one can see the Greedy KPCA as an alternative to KPCA classification.

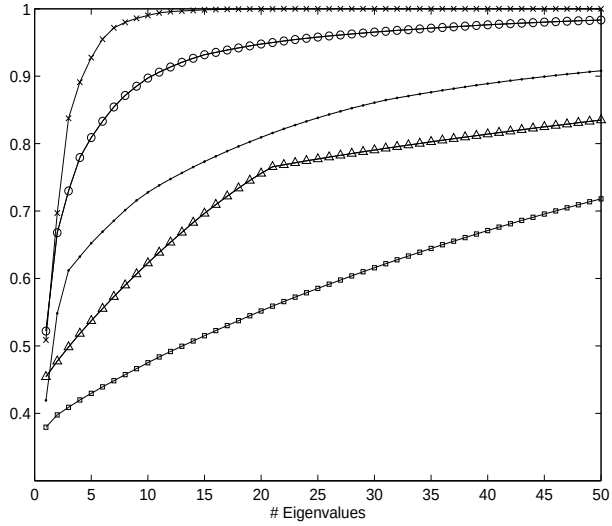


Figure 6.4: Cumulative eigenvalues sum of the kernel matrices. □ - *Splice*; Δ - *Ringnorm*; • - *Waveform* ; ∘ - *Image* and × - *Banana*.

6.3 USPS Dataset - Large Dataset

The USPS dataset described in detail in appendix A.1, is a benchmark of handwritten digits divided into a training dataset with 7921 images and a test dataset with 2007 images. Each image consists of 16×16 pixels. Then the input data vector $\mathbf{x}_n$ has dimension 256 and is formed by row concatenation of the original image. The study done considered the influence of noise in the feature extraction process as well as the performance of the classifier when the size of the training set varied, 10%, 50% and 100% of the available data. A Gaussian noise with variance of $\sigma^2 = 0.25$ will be added to each digit of the training and test sets, noisy data.

Figure 6.5: Digits without and with noise.

Figure 6.5 illustrates examples of digits and their noisy versions. The USPS dataset is used in many works and the best results report an error rate in the range [0.04 , 0.05] [13]. The complete training set was used to classify the test dataset with a nearest neighbor (NN) strategy and an error rate equal to 0.056 was achieved. The same training set was also used to compute linear discriminant functions (RL), the error rate of the test dataset is equal to 0.131.

In the next sections, the same analysis will be done in input and feature spaces using the PCA and Greedy KPCA algorithms respectively.

6.3.1 Input Space

In input space the basis vectors are computed using principal component analysis. For that, the covariance matrix of the training dataset is computed and the basis vectors correspond to its eigenvectors. Ordering the eigenvectors of the covariance matrix according to their related eigenvalues, it can be seen that the data is mostly spread in 50 of the 256 directions of the input space. Fig.6.6 shows the eigenvalues of the covariance matrix of the complete training dataset. The classifiers were trained for the different sizes of input which were determined by

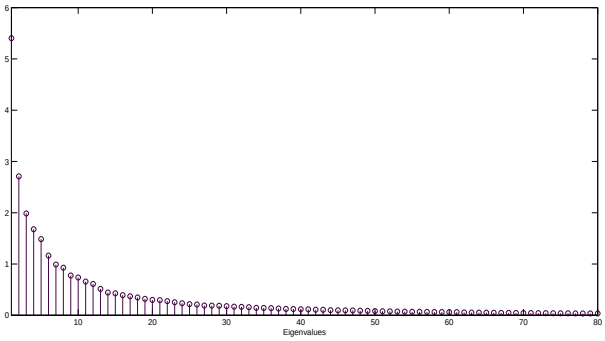


Figure 6.6: Eigenvalues of the covariance matrix of the training set (without noise).

the number of vectors L taken to form the basis vector matrix. In the PCA model L varies in the range $(1 - 100)$. The improvement of the linear classifier with kernel projections is to be expected, since in feature space the data should be linearly separable.

6.3.2 Feature Space

To compute the basis vectors in feature space by the Greedy KPCA algorithm, eqn. 4.5, an eigendecomposition of matrix $\mathbf{Q}$ needs to be performed. The values of R obtained make it possible to achieve feasible eigendecompositions even when the size of the training set is prohibitively large, for example using 50% or 100% of the available data. Besides that, the R size influences other aspects of the application of the method to compute the kernel features such as

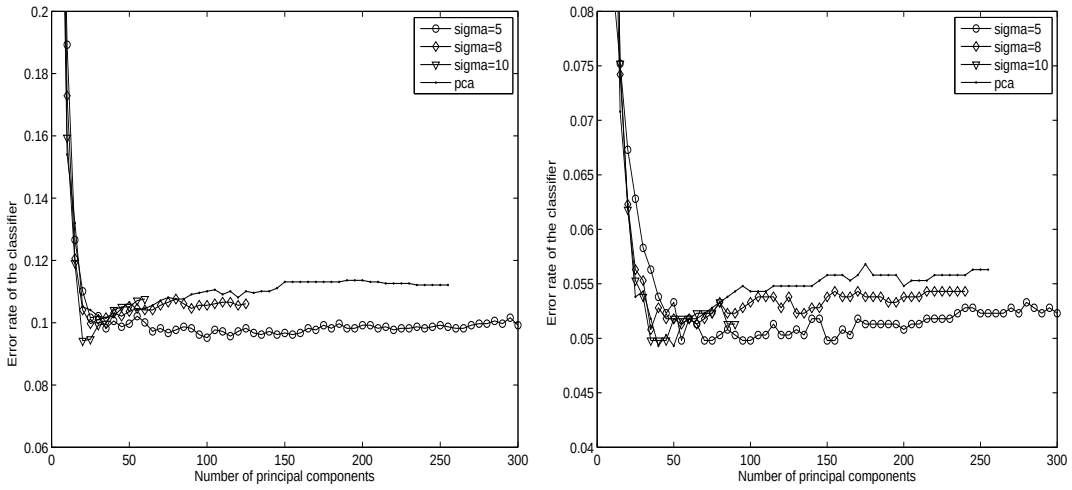
- storage requirements to store the data training set that belongs to $\mathbf{\Phi}_R$
- dimension of the data in feature space is limited to R, so the number of available directions L varies in the range $[1\ ,\ R]$.

Table 6.5 presents the size (R) of subset Φ_R for different sizes of the training set (N) and to different σ values.

Note that $\sigma = 5$ is the value closer to the one computed by eqn. 4.52 for the raw data and $\sigma = 8$ is the value closer to the one computed by eqn. 4.52 for the noisy data.

N	100%			50%			10%		
σ	5	8	10	5	8	10	5	8	10
raw data	1807	241	91	1169	206	80	335	132	63
noisy data	-	1062	306	-	775	282	-	318	190

Table 6.5: Size R of subset $\mathbf{\Phi}_R$ for different values of σ using training sets with different sizes N.



(a) (b)

Figure 6.7: Performance of NN using projections in input space (PCA) and in feature space. Training set with: 729 (*left*) or 7291 (*right*) images.

6.3.3 Results and Discussion

Several simulations were conducted to evaluate the suitability of the projected data for classification. The data is projected onto the L most significant directions that form the basis vectors and the values of the projections are used as input to the classifiers.

Figure 6.7 illustrates the performance of the NN classifier varying the number of projections. The classifiers trained with the complete training dataset have the best performance, achieving an error rate of 0.05 while with the smaller dataset, the error rate is around 0.1. With PCA the best performance is achieved using 50 projections roughly. This result has to be expected as the covariance matrix exhibits approximately 50 significant eigenvalues, the remaining eigenvalues are very close to zero. An error rate near to 0.05 is the best performance of the NN classifier, having PCA or kernel features. However, using $L > 50$ PCA projections the error rate increases slightly (0.007), while with the kernel features computed using $\sigma = 5$ the error rate is maintained, table 6.6.

Figure 6.8 shows the results for the linear classifier (RL), and the performance here is less dependent on the size of the training set. The error rate of the classifier when the inputs are the projections in feature space is around 0.09 while with PCA projections it is 0.14. The results presented in [14] show a similar tendency: the linear SVM classifier performs better using projections computed with KPCA instead of PCA. Calculating 2048 projections

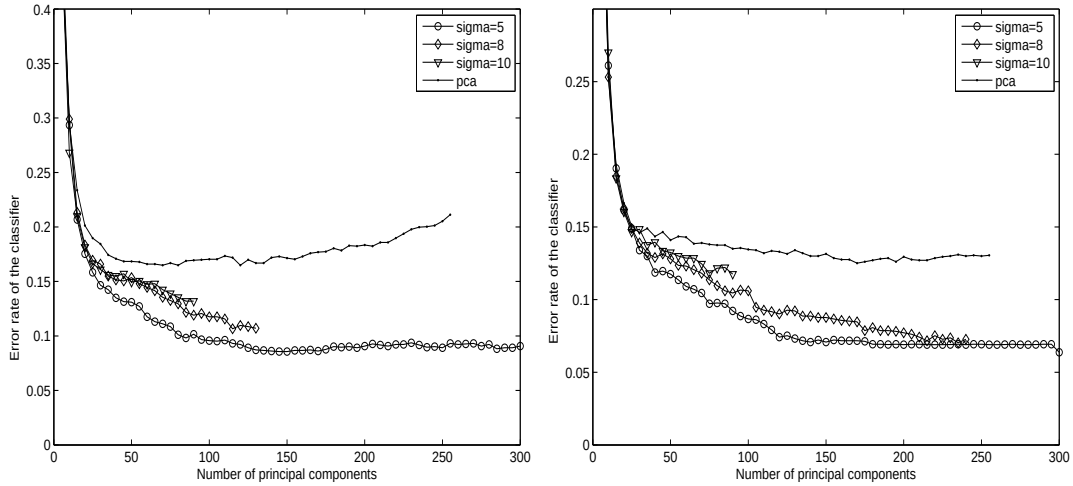


Figure 6.8: Performance of the RL using projections in input space (PCA) and in feature space. Training set with: 729 (*left*) or 7291 (*right*) images.

in KPCA feature space, the improvement in error rate amounts to 0.046 if a polynomial kernel is used. The dataset used for the training consisted of 3000 examples. Note that the NN classifier, having as input the kernel projections and trained with 10% of the training dataset, shows a similar performance. The optimal number of PCA projections is around 50, while with kernel methods more than 100 are needed. In kernel projective techniques, the number (R) of basis vectors has the highest value when $\sigma = 5$, but the performance of RL after $L = 100$, does not change when L is increased. On the other hand, for the other values of σ, the best performance is achieved using projections onto all available basis vectors. All figures demonstrate that the RBF kernel with $\sigma = 5$ shows the best performance. But if noise is added, this parameter needs to be changed and best performance is obtained with $\sigma = 8$.
The KPCA model without centering can be computed for the smaller dataset ($N = 729$) and the resulting eigenvalues λ_i can be compared to the eigenvalues $\tilde{\lambda}_i$ of the Greedy KPCA model. Figure 6.9 shows the relative error defined as

$$\begin{aligned} er(i) &= \frac{\left|\lambda_i - \tilde{\lambda}_i\right|}{\lambda_i} \\ &= \left|1 - \frac{\tilde{\lambda}_i}{\lambda_i}\right| \end{aligned} \tag{6.2}$$

The relative error maintains the same range for all σ values of the RBF kernel and it is larger for the leading eigenvalues.
Figure 6.10 shows the difference between $S(p)$ when the Greedy KPCA is trained with $N = 729$ or $N = 7921$. It is obvious that $S(p)$ exhibits a similar trend in both training sets. Note that the relative value of the first eigenvalue ($S(1)$) is similar for both training sets. However, comparing the $S(p)$ of both training sets, the same absolute level of $S(p)$ is achieved for different values of p. Therefore, this comparison explains the values of R of table 6.5.
Table 6.6 presents the SNR using a variable number of projections L. The table also shows the performance of the system when Gaussian noise is added to both the training and test

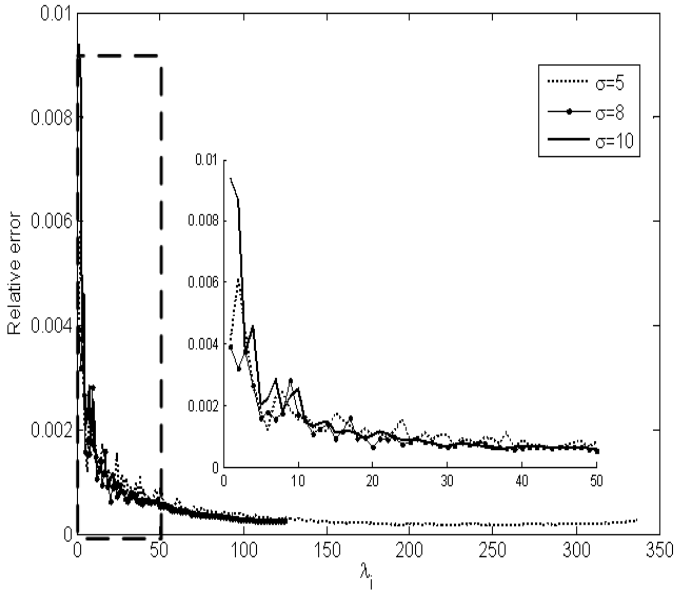


Figure 6.9: Relative error between the eigenvalues of the greedy and kernel matrices, $N = 729$.

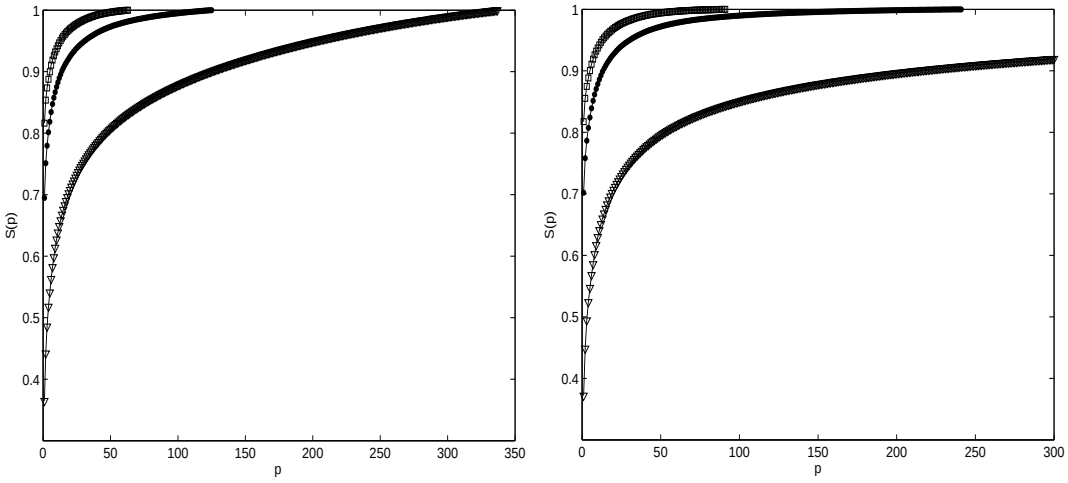
datasets. It is obvious that the performance of all classifiers degrades if noise is added to the data. The degradation level has similar values for both classifiers whatever the number of projections used as input.

	N	100%			50%			10%		
	L	10	50	100	10	50	100	10	50	100
raw data	PCA - NN	0.104	0.049	0.056	0.123	0.061	0.061	0.125	0.102	0.110
	PCA - RL	0.252	0.143	0.135	0.214	0.146	0.143	0.295	0.171	0.171
	RBF5 - NN	0.083	0.054	0.050	0.099	0.061	0.061	0.132	0.101	0.097
	RBF5 - RL	0.193	0.120	0.086	0.184	0.125	0.087	0.235	0.134	0.095
noisy data	PCA - NN	0.192	0.093	0.102	0.208	0.120	0.114	0.240	0.180	0.183
	PCA - RL	0.280	0.168	0.166	0.255	0.193	0.183	0.322	0.224	0.212
	RBF8 -NN	0.212	0.097	0.105	0.195	0.120	0.106	0.191	0.153	0.170
	RBF8 - RL	0.438	0.165	0.162	0.238	0.172	0.169	0.290	0.215	0.204

Table 6.6: Error rate of the classifiers using data (training and test) sets with and without noise.

6.4 Conclusion

A new insight into unsupervised feature extraction techniques based on subspace models was discussed in this chapter. The data projected onto subspace models were new data representations which might be more suitable for classification. In input space the features were calculated using the PCA decomposition. In feature space, KPCA and Greedy KPCA were applied. The numerical simulations compared the performance of classifiers using kernel



(a) $N = 729$ (b) $N = 7291$

Figure 6.10: Normalized cumulative sum of eigenvalues left: $N = 729$ right: $N = 7291$. From top to down: $\sigma = 10$, $\sigma = 8$, $\sigma = 5$.

features, principal component features and a direct classification of the raw data using two classifiers: the nearest neighbor (NN) and linear discriminant function (RL). Furthermore, to evaluate the impact of the projective techniques, a comparative study with the best results published in [7] was presented and discussed. The improvement in performance has the same value as the one presented in [7]. The feature extraction method was also applied to the USPS dataset in order to do the classification. In what concerns classification, using the projections in feature space and a simple linear classifier, the performance was good even when the training set had a reduced size. Experiments showed that these projective subspace techniques casted into Greedy KPCA approach achieved a performance which was similar to a full KPCA analysis. It was demonstrated that the model performs well on a very large dataset like the USPS dataset. Although it is often assumed that extracting non-linear features always results in an increase of performance in classification, the results presented did not give this indication for all datasets. The reason is mostly related to the data characteristics. It is an often used practice to show that the performance is dependent on the parameters of the model, specifically the parameter σ of the RBF kernel functions. Simulations showed that it is possible to estimate an appropriate value of this parameter to achieve a good tradeoff between the eigenvalue decay and the number of non-zero eigenvalues of the kernel matrix and still result in a good classification performance. Using eqn. 4.52 to compute the σ value in the training dataset allowed the computation of a subspace model to project the training and the testing datasets.

Bibliography

[1] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[2] B. Moghaddam, "Principal Manifolds and Probabilistic Subspaces for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 780–788, 2002.

[3] B. Schölkopf, S. Mika, and all, "Input Space vs. Feature Space in Kernel-Based Methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[4] J. Li, X. Li, and D. Tao, "KPCA for semantic object extraction in images," *Pattern Recognition*, vol. 41, pp. 3244 – 3250, 2008.

[5] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping using the Nyström Method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–224, 2004.

[6] J. Yang, X. Gao, Z. D., and J. Y. Yang, "Kernel ICA: An alternative formulation and its application to face recognition," *Pattern Recognition*, vol. 38, pp. 1784–1787, 2005.

[7] G. Rätsch, T. Onoda, and K. R. Müller, "Soft Margins for Adaboost," *Machine learning*, vol. 42, pp. 287–320, 2001.

[8] A. R. Teixeira, A. M. Tomé, and E. W. Lang, "Feature Extraction using Low-Rank Approximations of the Kernel matrix," in *LNCS 5112- ICIAR 2008* (A. Campilho and M. Kamel, eds.), (Porto), pp. 404–412, 2008.

[9] A. R. Teixeira, A. M. Tomé, and E. W. Lang, "Feature extraction using linear and non-linear subspace techniques," in *Artificial Neural Networks-ICANN 2009* (Springer-Verlag, ed.), vol. II, (Cyprus).

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2001.

[11] X. Yong, D. Zhang, F. Song, J. Yang, Z. Jing, and M. Li, "A method for speeding up feature extraction based on KPCA," *Neurocomputing*, vol. 70, pp. 1056–1061, 2007.

[12] S. Mika, B. Schölkopf, A. Smola, K. R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," *In: Advances in Neural Information Processing*, vol. 11, p. 536Ű532, 1999.

[13] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[14] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

Chapter 7

Conclusions and Open Problems

"Fundamental progress has to do with the reinterpretation of basic ideas."
- Alfred North Whitehead -

The cornerstone of this work was linear and non-linear projective subspace techniques. Two ways to compute the subspace models were presented and compared in this work. Each subspace method has a model associated to it, which is described by the basis vectors. The subspace model is formed using the elements from the eigendecomposition of kernel or correlation/covariance matrices computed on multidimensional data sets. The difference that sets the methods apart is the path chosen to calculate them.

Starting by linear subspace projective techniques, the main steps of SSA were presented and explained using linear invariant system approaches. It was shown that SSA can be interpreted as a bank of filters which might be useful to achieve a clear view of the outcomes of the methods. The Local SSA algorithm was also exposed and the choice of the parameters, namely the dimension of the embedding, the number of clusters and the number of directions were discussed. These were automatically adjusted according to the input of the algorithm, i.e. the algorithm is autonomous and independent from the user.

Besides the study of linear subspace techniques, in this work different non-linear subspace techniques were addressed, based on KPCA methology. All the proposed models were consistently described by basis vector matrices using the dual form to exploit the dot product properties of kernels. Two bottlenecks needed to be dealt with using this algorithm, namely the dimension of the kernel matrix and the pre-image problem.

If the dataset is large, the eigendecomposition of the kernel matrix represents a major computational burden and the eigendecomposition is often impractical in real data applications. When applying kernel methods, an eigendecomposition of the related kernel matrix is often required, particularly the most significant eigenvalues and corresponding eigenvectors. This lead to the use and study of low-rank approximations to the kernel matrix based on a Nyström

extension, namely the Greedy Kernel PCA. The big advantage of using low-rank approximations was that they were related to a smaller subset of the data, which constituted the basis of the subspace.

The Greedy KPCA and KPCA were reformulated under a unifying algebraic notation underlying the differences between both approaches. These differences were the complexity inherent to each and the properties of the projections computed by them. Different Nyström approaches (orthogonal and non-orthogonal) to compute the basis vectors in Greedy KPCA algorithm were discussed. Several experiments were done and they showed that Greedy KPCA had the best performance when used with an orthogonal approach.

To obtain the data in input space it is unavoidable to deal with the pre-image problem which constitutes the most complex step in the whole processing chain. The two methods of pre-image estimation discussed in the literature (distance method and fixed-point algorithm), were modified into simple ways which proved very effective in certain applications. The distance method sometimes did not yield reliable results because it bases itself on the number S of nearest neighbors chosen. If S was smaller than the dimension of the data space, the solution can often be closely approximated by simply choosing the mean of nearest neighbors which speeded up the computation considerably. The fixed-point algorithm is addressed in the literature by a random initialization, however a very slow convergence is the result. To speed up the convergence, the mean of the nearest neighbors algorithm was proposed.

The application of projective techniques in multivariate datasets can be done directly, however such can not be done in unidimensional time series. In this work, the feasibility of these techniques in unidimensional signals was also explored. So, a non-linear mapping in input space, known as embedding, M, must be done, which represents a non-linear transformation of the dataset in time delayed coordinates. The diagonal averaging step which is the reverse of embedding is also used.

The subspace techniques studied in this work were used in two distinct applications: denoising and feature extraction.

In denoising applications, a quantitative analysis was done to evaluate the linear and non-linear subspace techniques using artificial mixtures of a set of selected EEG signals. The evaluation was performed in both time and frequency domains by using correlation coefficient and coherence functions, respectively. Applying the linear subspace techniques, three variants were evaluated: SSA, Local SSA and SSA with MDL changing the M (embedding parameter). The three variants had a high correlation coefficient in the segments where beta and alpha were dominant. When theta or delta were dominant, Local SSA was better than any other variant. In conclusion, this algorithm was more stable and it was possible to find a unique M for all types of segments.

The same study was done to non-linear approaches (Greedy KPCA) and it was shown that both approaches (linear and non-linear) had a similar performance in what concerned frequency distortion in the frequency range of beta and alpha bands. However, Local SSA showed a better performance because the corrected EEG exhibited less distortion in all frequency bands. The preliminary real EEG analysis confirmed those results.

The algorithms (Local SSA and Greedy KPCA) were incorporated in the EEGLAB environ-

ment. This open-software tool based on MATLAB offers visualization facilities that will allow the accomplishment of the clinical evaluation task. The goal was to use this facility in the visualization of critical segments of signals from a database of epileptic patients recorded in long-term monitoring sessions and study the impact of the application of the algorithms. In the described scenario, the algorithms can be applied in parallel to the channels that suffer from high-amplitude artifacts. This could be useful to detect the onset of a focal seizure.
The feature extraction based on linear and non-linear subspace techniques is another application explored in this work. A new insight into unsupervised feature extraction techniques based on kernel subspace models was provided. The data projected onto kernel subspace models were new data representations which might be more suitable for classification. As the basis vectors were expressed in terms of the training dataset, the Greedy KPCA method had the advantage of selecting a subset of the training set, reducing in that way the complexity of the problem during testing. The results showed that Greedy KPCA had a similar performance when compared to KPCA. Furthermore, for a large dataset, like USPS, the KPCA model was viable. The results obtained did not allow the assumption that extracting non-linear features had always an increase in performance, mostly due to the characteristics of the data. The parameters of the model, namely the σ parameter of the RBF kernel functions, were used quite often as a performance benchmark. The results proved that it was possible to estimate a value to that parameter retaining a good classification.

7.1 Directions for Further Work

Below, some questions and possible new directions of research raised by this work are listed:

- Applying the linear and non-linear algorithms to detect the onset of a focal seizure in clinical environment. The new facilities (plug-ins) developed in EEGLAB need to be improved to cope with long segments of signal. The goal is to use this facility in the visualization of critical segments of signals from a database of epileptic patients recorded in long-term monitoring sessions and study the impact of the application of the algorithms. In the described scenario, the algorithms can be applied in parallel to the channels that suffer from high-amplitude artifacts.

- Using the algorithms in evoked potential signals to detect the P100 or P300 waves. There are other problems in neurophysiology where the signal needs "enhancement" and not only the elimination of high amplitude artifacts such as evoked potential studies.

- Applying the subspace distance method as a pre-processing method to optimize the algorithms.

- Studying one-class classification algorithms for feature extraction in EEG signals.

Appendix A

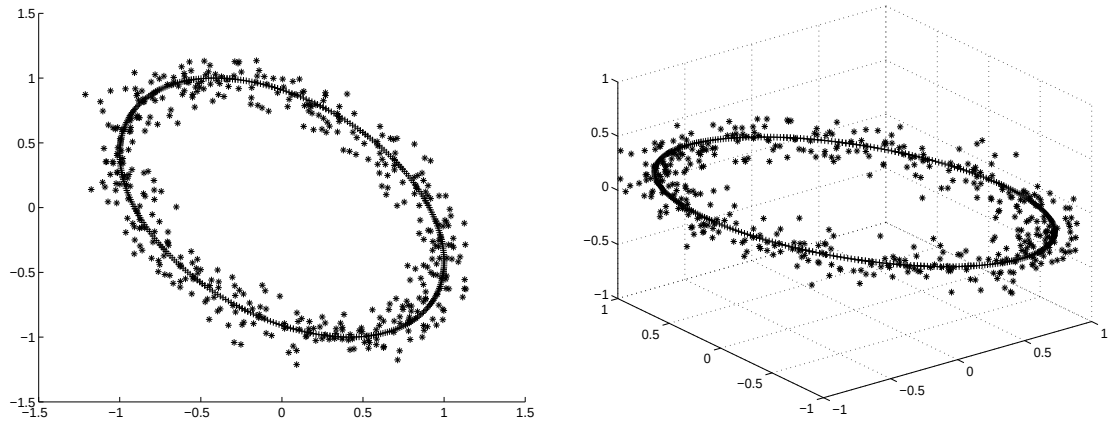
Appendix

A.1 Datasets

In this work three kinds of datasets were used to evaluate the algorithms performance. The following sections describe succinctly the main features of the datasets used.

A.1.1 Sinusoidal Data Set

The toy example to be discussed along this work comprises an artificially generated unidimensional sinusoid. The time series $\tilde{x}[n]$ with $N = 500$ samples was contaminated with Gaussian white noise $x[n] = \tilde{x}[n] + r[n]$ to result in a signal-to-noise ratio of SNR $= 20dB$ and was embedded in $2D$ and in $3D$ space, figure A.1.



(a) 2D (b) 3D

Figure A.1: Embedded signals in 2D space. Sinusoid (+) and sinusoid + gaussian noise(*).

A.1.2 USPS Dataset

The USPS dataset contains 16×16 normalized gray scale handwritten digits. The dataset consists of a training set with 7291 images and a test set with 2007 images.
Then the input data vector of the algorithms used in this work has dimension 256 and is

formed by row concatenation of the original image after adding white Gaussian noise (zero mean and variance of 0.25), figure A.2. Note that the dataset for each digit type has a different

Figure A.2: Set of digits: *Right* - Original, *Left* - with Gaussian noise ($\sigma^2 = 0.25$).

number of elements (in the range $568 - 1005$), table A.1.2.

Digit	0	1	2	3	4	5	6	7	8	9
D	1194	1005	731	658	652	556	664	645	568	644

Table A.1: Datasets Information.

A.1.3 Benchmarks

On Gunnar Rätsch's web site a collection of datasets can be found (accessible at http://ida.first.fraunhofer.de/projects/bench). Several state-of-the-art algorithms have already been applied to those datasets, among which the SVM, KPCA, Adboost, KFD [1, 2, 3]. All results are reported on the web site. Table A.2 resumes the information of the 13 datasets present in the literature. For all collections, 100 partitions into test and training set were

	Best [1]	D	N
B. Cancer (BC)	25.9 ± 4.6	9	200
Diabetis (Di)	23.5 ± 1.7	8	468
German (Gr)	23.6 ± 2.1	20	300
Heart (Hr)	16.0 ± 3.3	13	170
F. Solar (FS)	32.4 ± 1.8	9	600
Thyroid (Ty)	4.4 ± 2.2	5	140
Titanic (Ti)	22.4 ± 1.0	3	150
Twonorm (Tn)	2.7 ± 0.2	20	400
Image (Im)	2.7 ± 0.7	18	1010
Ringnorm (Rg)	1.6 ± 0.1	20	400
Splice (Sp)	9.5 ± 0.7	60	1000
Waveform (Wv)	9.8 ± 0.8	21	400
Banana (Ba)	10.7 ± 0.4	2	400

Table A.2: Overview of the results in literature.

generated with a variable dimension, except *Splice* and *Image* which only have 20 partitions. On each partition datasets, different classification algorithms were used. The results show the average test error over all the partitions and the standard deviation. The first column (Best)

shows the best average classification error achieved in [1], the second column, D, the number of features of raw data and the last column, N, the training data size.
Furthermore, it is possible to download the generalization errors for every data partition from the web page.

A.2 EEG Data Collection

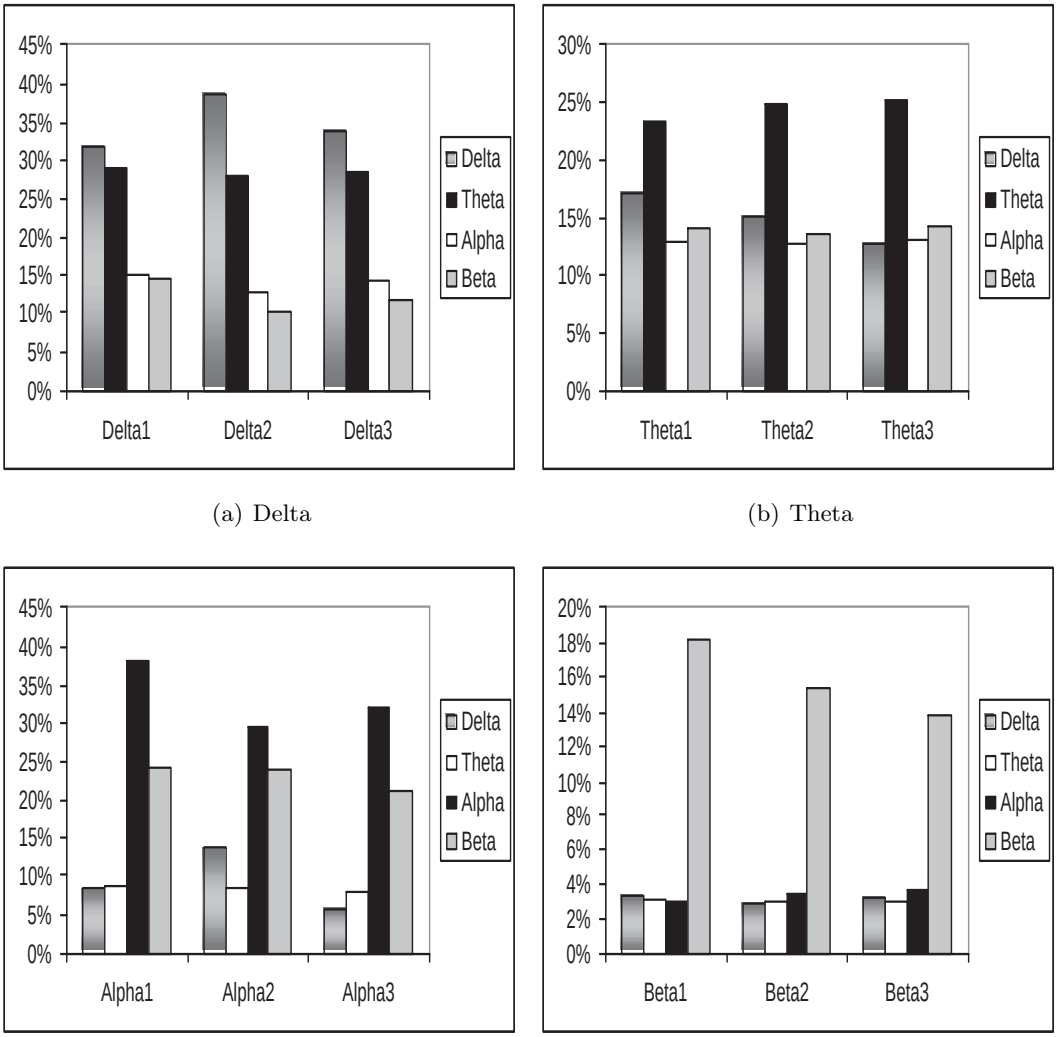
The EEG data collection was recorded at hospital Geral Santo António and belongs to a group of patients with several pathologies. The EEG signals were recorded using 19 electrodes placed according to the $10-20$ system and mounted with a common ground reference to Fz. The signals were filtered and digitalized at a sampling rate of $250Hz$ and stored as European Data Format (EDF), using an EEG XLTEK recording system.
Monopolar brain signals using the Cz electrode as reference, were visualized using EEGLAB [4]. The signals were selected by three specialists after visual inspection. The signals were selected with a clear predominance in one of the characteristic bands (beta [13 , 25] Hz; alpha [7.5 , 13] Hz; theta [3.5 , 7.5] Hz ; delta [0 , 3.5] Hz) and clean of the artifacts (EOG, EMG or patient movements), figure 5.1.
The dataset will then be classified into four types according to the visibility/dominance of one of the activities. Twelve multichannel segments, each of them having 10 seconds (available in `www.ieeta.pt\~ateixeira`), were selected, having the following characteristics:

- Type A: 3 signals with delta activity (F8, T4, T6)
- Type B: 3 signals with theta activity (T4, T6, T5)
- Type C: 3 signals with alpha activity (O2, F8, O1)
- Type D: 3 signals with beta activity (C3, C3, C3)

For a better interpretation of the results, the percentage of energy of each signal in the four principal bands was evaluated, figure A.3. In the frequency domain, the power spectral density computed by the Welch method was considered. The percentage of energy was measured considering the four principal bands.
The graphics of figure A.3 confirm that the selected EEG segments have the main energy concentrated in their respective band and also show that all bands have activity in all segments.

A.2.1 Artifacts

In this study, a set of 10 EOG artifacts was considered, figure A.4. The artifacts are obtained by processing segments of EEG with clear interference of EOG (usually corresponding to a frontal lead) or with a clear interference of patient movements (base line drifts). For that purpose, every artifact was recorded from real EEG signals belonging to a different signal. Each one was processed by the SSA algorithm and the artifact component is extracted accordingly to its relation to the largest eigenvalue (SSA with one direction).



(a) Delta

(b) Theta

(c) Alpha

(d) Beta

Figure A.3: % Energy in each band for different segments of signal.

Note that SSA with one direction always extracts a component that is related to an artifact, however in most of the cases the corrected version of the EEG still shows some remnants of the artifact, as was shown firstly in [5] and then resumed in [6]. By visual inspection, the signals were recognized as clearly defined artifacts not showing any EEG relevant information. The characteristics of each artifact are variable, particularly in terms of amplitude, number of blinks per segment and base line variation. In figure A.5, the % of energy in each band for different EOG signals is represented. In table A.3 all the information about the used artifacts in the experiments related in section 5.3 is presented.

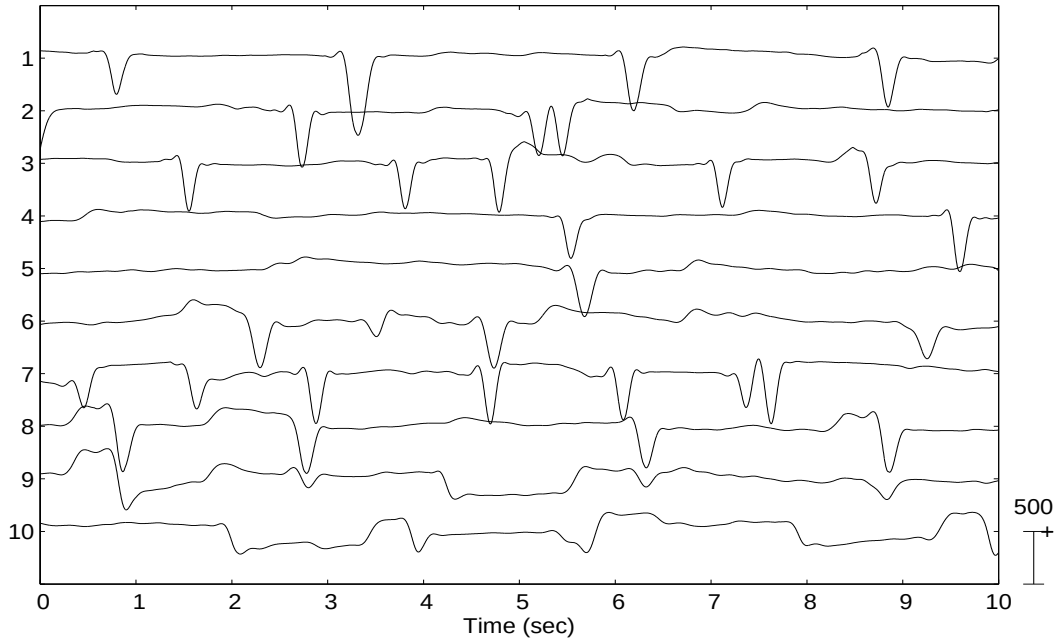


Figure A.4: EOG artifacts used in the experiment.

A.2.2 Artificial Mixtures

The multichannel dataset is obtained by the mixing model, eqn. A.1, that follows a strategy proposed in [7]

$$\mathbf{X} = \mathbf{MS} \tag{A.1}$$

where $\mathbf{S}$ is the source matrix that has the first M rows with M EEG signals and the last row is the artifact, $\mathbf{M}$ is the mixing matrix and $\mathbf{X}$ is the mixing signal matrix. The mixing matrix $\mathbf{M}$ is obtained by

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & m1 \\ 0 & 1 & 0 & 0 & \cdots & 0 & m2 \\ 0 & 0 & 1 & 0 & \cdots & 0 & m3 \\ \vdots & \ddots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & m16 \\ 0.5 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Notice that each row of $\mathbf{X}$ is obtained by linearly adding two signals: the EEG and the artifact. The coefficients, m_i, $i = 1, ..., M$ are chosen so that the amplitude of the artifacts decreases from the frontal channels to the channels placed on the back of the head. The last row of $\mathbf{M}$ constitutes the contamination of the artifact signal with a frontal EEG. For quantitative comparison purposes, between spacial and temporal approaches, only one channel of each multidimensional segments will be used. In order to compare the performance of different methods, the coefficients (m_i) that correspond to the set of reference signals are chosen to have always the same value for all original signals with different activities.

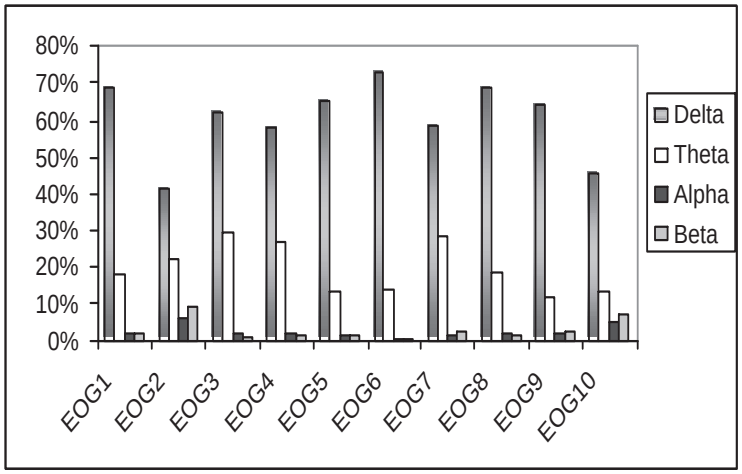


Figure A.5: % Energy of the EOG signals in each band.

	Amp	Blinks	Var $\times 10^4$
EOG1	839	4	1.48
EOG2	650	3	0.75
EOG3	670	5	0.96
EOG4	592	2	0.48
EOG5	567	1	0.44
EOG6	650	4	1.00
EOG7	618	7	1.36
EOG8	660	4	1.21
EOG9	580	5	1.01
EOG10	408	4	1.06

Table A.3: EOG artifacts characteristics: Am - amplitude of the signal; Blinks - number of blinks and Var- variance of the signal.

A.3 Cholesky Decomposition

The Cholesky decomposition is a decomposition of a symmetric, positive-definite matrix $\mathbf{A}$ into the product of a lower triangular matrix and its conjugate transpose. The matrix $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{L}\mathbf{L}^* \tag{A.2}$$

where $\mathbf{L}$ is a lower triangular matrix with strictly positive diagonal entries, and $\mathbf{L}^*$ denotes the conjugate transpose of $\mathbf{L}$.

Bibliography

[1] G. Rätsch, T. Onoda, and K. R. Müller, "Soft Margins for Adaboost," *Machine learning*, vol. 42, pp. 287–320, 2001.

[2] X. Yong, D. Zhang, F. Song, J. Yang, Z. Jing, and M. Li, "A method for speeding up feature extraction based on KPCA," *Neurocomputing*, vol. 70, pp. 1056–1061, 2007.

[3] B. Schölkopf, S. Mika, and all, "Input Space vs. Feature Space in Kernel-Based Methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[4] S. Makeig, S. Debener, J. Onton, and A. Delorme, "EEGLAB," 2007.

[5] A. R. Teixeira, E. W. Tomé, A. M.and Lang, P. Gruber, and A. M. Siva, "Removal of Ocular Artifacts from Electroencephalogram by Singular Spectrum Analysis," in *2nd Int. Conf. Comput. Intelligence Medicine Healthcare (CIMED)*, (Costa da Caparica, Portugal), pp. 24–29, 2005.

[6] A. R. Teixeira, A. M. Tomé, E. Lang, P. Gruber, and A. M. Silva, "Automatic removal of high-amplitude artifacts from single-channnel electroencephalograms," *Computer Methods and Programs in Biomedicine*, vol. 83, no. 2, pp. 125–138, 2006.

[7] C. W. Anderson, J. N. Knight, T. O'Connor, M. J. Kirby, and A. Sokolov, "Geometric subspace methods and time-delay embedding for EEG artifact removal and classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 142–146, 2006.