

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

SUPPORT VECTOR METHODS FOR UNSUPERVISED LEARNING

Promotor :
Prof. dr. ir. J. Suykens

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Carlos ALZATE

May 2009



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

SUPPORT VECTOR METHODS FOR UNSUPERVISED LEARNING

Jury:
Prof. dr. ir. Y. Willems, voorzitter
Prof. dr. ir. J. Suykens, promotor
Prof. dr. ir. J. Vandewalle
Prof. dr. ir. B. De Moor
Prof. dr. M. Hubert
Prof. dr. ir. M. Van Barel
Prof. dr. ir. M. Verleysen (UCL, Louvain-la-Neuve)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Carlos ALZATE

U.D.C. 681.3*I5 May 2009

©Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

ISBN 978-94-6018-054-5

U.D.C. 681.3*I5

D/2009/7515/37

Foreword

The research work presented in this thesis corresponds to the result of a quite smooth and enriching journey during my doctoral studies at the SISTA research division. A journey full of scientific challenges, successful experiments, interesting insights and satisfactory experiences.

I am very grateful to my promotor Johan Suykens for this opportunity to start my doctoral studies in this research group. Johan, your continuous support, constructive criticism, encouraging enthusiasm, daily feedback and key ideas have been essential for this research work and the rewarding experiences that came along during my studies. Many thanks also to Bart De Moor, Joos Vandewalle, Mia Hubert, Marc Van Barel and Michel Verleysen for being part of the jury of this thesis. I would like to acknowledge my colleagues, who made this journey an enjoyable experience and with whom I collaborated on several occasions. I extend my gratitude to the administrative staff, namely Ida Tassens and Ilse Pardon for the help I received during these years.

My gratitude also to many friends, especially Lody Iza, León Gómez, Alvaro Caicedo, Juan Quinchía, the changos, who have always been very supportive and with whom I have shared memorable moments. My most special thanks go to my girlfriend An for the daily support and care. An, you have always been there patiently with kind words and optimistic attitude. This gratitude extends also to An's family, for their interest in my Ph.D and for the nice family meetings.

Finalmente, quiero agradecer a mi familia, en especial a mi tía Betty, mi tío Heri y a mi madre por todo el apoyo y confianza que siempre me han dado.

Carlos Alzate

Leuven, May 2009.

Abstract

In the context of unsupervised learning, elements such as extensions to out-of-sample data, regularized formulations, model selection criteria and applicability to large-scale data become key aspects. Starting from least squares support vector machine (LS-SVM) core formulations, we provide extensions of unsupervised learning methods towards the incorporation of advanced elements. The primal-dual optimization framework typical of LS-SVM formulations allows extensions of the core models by adding additional constraints to the primal problem or by changing the loss function. This family of kernel methods is formulated in terms of high dimensional feature spaces at the primal level. The model at the dual level is represented in terms of Mercer kernels. In this thesis, we formulate extensions for classical unsupervised learning applications such as principal component analysis (PCA), canonical correlation analysis (CCA), independent component analysis (ICA) and spectral clustering. In the case of kernel PCA, a formulation with general underlying loss function is proposed. This formulation allows the incorporation of robustness and sparseness into kernel PCA by using a robust epsilon-insensitive loss function. For kernel CCA, a multivariate formulation is described where regularization appears naturally in the primal problem. This method can be also used a contrast function for ICA when universal kernels are used. In the context of clustering, we propose a multiway spectral clustering methodology with out-of-sample extensions based on weighted kernel PCA together with a model selection criterion exploiting the structure of the solutions. This clustering method is further extended towards developing sparse models and incorporating prior knowledge. Real-life applications considered in this thesis include image segmentation, image denoising, time series clustering, image demixing and scientific journal clustering.

Notation

Variables and Symbols

x^T	Transpose of the vector x
Ω^T	Transpose of the matrix Ω
Ω_{ij}	ij-th entry of the matrix Ω
I_N	$N \times N$ identity matrix
1_N	N-dimensional vector of ones
$\{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$	Training sample of N data points
$\mathcal{F}$	Feature space of dimension n_h
$\varphi(\cdot)$	Feature map
$K(x_i, x_j)$	Mercer kernel evaluated on data points x_i, x_j
$\{\mathcal{A}_p\}_{p=1}^k$	Grouping composed of k clusters
$\alpha_i^{(l)} \in \mathbb{R}$	i-th entry of the l-th eigenvector $\alpha^{(l)} \in \mathbb{R}^N$
D	$N \times N$ graph degree matrix
$\min_x f(x)$	Minimization over x, minimal function value returned
$\arg\min_x f(x)$	Minimization over x, optimal value of x returned

Acronyms

ARI	Adjusted Rand Index
BLF	Balanced Line Fit
cBLF	Constrained Balanced Line Fit
CCA	Canonical Correlation Analysis
COCO	Constrained Covariance
ECOC	Error Correcting Output Codes
ESI	Essential Science Indicators
FDA	Fisher Discriminant Analysis
ICA	Independent Component Analysis
KCC	Kernel Canonical Correlation
KCA	Kernel Component Analysis
KGV	Kernel Generalized Variance
KKT	Karush-Kuhn-Tucker
KMI	Kernel Mutual Information
KRC	Kernel Regularized Correlation
LLE	Local Linear Embedding
LPP	Locality Preserving Projections
LS-SVM	Least Squares Support Vector Machine
MDS	Multidimensional Scaling
MSV	Mean Silhouette Value
NMI	Normalized Mutual Information
OLS	Ordinary Least Squares
PAR	Periodic Autoregression
PCA	Principal Component Analysis
POD	Proper Orthogonal Decomposition
QP	Quadratic Programming
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
RKHS	Reproducing Kernel Hilbert Space
SQP	Sequential Quadratic Programming
SVM	Support Vector Machine
TDP	Typical Daily Profile
TF-IDF	Term Frequency - Inverse Document Frequency

Contents

Foreword **i**

Abstract **ii**

Notation **v**

Contents **vii**

1 Introduction **1**
1.1 General Background 1
1.2 Challenges and Problems 2
1.3 Objectives 3
1.4 Chapter by Chapter Overview 4
1.5 Contributions of this work 6

2 Kernel Methods for Unsupervised Learning **11**
2.1 Kernel Methods 12
2.1.1 Mercer kernels 12
2.1.2 Support Vector Machines for Classification 13
2.1.3 Kernel functions 14
2.2 Kernel PCA 16
2.2.1 Linear PCA 16
2.2.2 PCA in Feature Spaces 17
2.3 Kernel CCA 19
2.3.1 Linear CCA 20
2.3.2 CCA in kernel-induced feature spaces 21
2.4 ICA via the kernel trick 22
2.4.1 Preliminaries 22
2.4.2 Kernel ICA 23

2.5 Spectral Clustering 24
2.5.1 Preliminaries 26
2.5.2 Markov Random Walks 27
2.5.3 The k-way *NCut* Relaxation 28
2.5.4 Perturbation Analysis and Piecewise Constant Eigenvectors 29
2.5.5 From Eigenvectors to Clusters 30

3 Least Squares Support Vector Machines 33
3.1 LS-SVM models for classification 34
3.2 LS-SVM models for regression 36
3.3 LS-SVM formulation to kernel PCA 38
3.3.1 Unconstrained Formulation 38
3.3.2 Constrained Formulation 39
3.4 LS-SVM formulation to kernel CCA 42
3.5 Conclusions 44

4 Kernel Component Analysis 45
4.1 Extension to general loss functions 46
4.1.1 Set of Nonlinear Equations 46
4.1.2 Weighting scheme 48
4.2 A robust and sparse loss function 51
4.2.1 Robustness in PCA and kernel PCA 51
4.2.2 Sparseness 53
4.3 Algorithms 54
4.3.1 SQP Algorithm 54
4.3.2 Weighted Algorithm 57
4.4 Model Selection 58
4.5 Experimental Results 60
4.6 Conclusions 62

5 ICA via Kernel Regularized Correlation 69
5.1 The Kernel Regularized Correlation (KRC) 70
5.1.1 Multivariate Kernel CCA 70
5.1.2 The KRC Contrast Function 72
5.2 Approximations and Algorithm 73
5.2.1 The Incomplete Cholesky Decomposition 73
5.2.2 The Stiefel Manifold 74
5.3 Model Selection 75
5.4 Experimental Results 77

5.4.1 Toy Examples - Simple signals . . . 77
5.4.2 Toy Examples - Source distributions . . . 78
5.4.3 Speech signals . . . 79
5.4.4 Image demixing . . . 81
5.5 Conclusions . . . 83

6 Spectral Clustering with Out-of-Sample Extensions 85
6.1 Preliminaries . . . 86
6.2 Spectral Clustering and Weighted Kernel PCA . . . 87
6.2.1 Multiway Formulation . . . 87
6.2.2 Encoding / Decoding Scheme . . . 89
6.3 Out-of-Sample Extension . . . 90
6.4 Model Selection . . . 92
6.5 Examples . . . 94
6.6 Application: Image Segmentation . . . 99
6.7 Application: Clustering Load Time Series . . . 103
6.7.1 Data Description . . . 106
6.7.2 Existing Methodology . . . 109
6.7.3 Proposed Clustering Scheme . . . 110
6.8 Application: Scientific Journal Clustering . . . 114
6.8.1 Journal By Term/Concept dataset . . . 114
6.8.2 Journal cross-citation dataset . . . 115
6.8.3 Data Integration . . . 115
6.8.4 Experimental Results . . . 116
6.9 Conclusions . . . 117

7 Sparse Spectral Clustering 125
7.1 Sparse model via the incomplete Cholesky decomposition . . . 126
7.1.1 Approximated Eigenvectors . . . 126
7.1.2 Reduced Set Method . . . 127
7.2 Algorithm . . . 128
7.3 Empirical Results . . . 129
7.3.1 Three Gaussian clouds in 3D . . . 130
7.3.2 Large-scale intertwined spirals problem . . . 130
7.3.3 Image Segmentation . . . 131
7.4 Conclusions . . . 133

8 Spectral Clustering with Pairwise Constraints 137
8.1 Hard Pairwise Constraint . . . 138
8.2 Soft Pairwise Constraint via Regularization . . . 141

8.3 The Constrained BLF 143
8.4 Experimental Results 143
8.4.1 Toy Problems 145
8.4.2 Image Segmentation 147
8.5 Conclusions 150

9 General Conclusions **155**
9.1 Concluding Remarks 155
9.2 Future Research 156

A The Nyström Method **159**

Bibliography **161**

Curriculum Vitae **171**

Publications by the author **173**

Chapter 1

Introduction

1.1 General Background

The topic of this thesis is related to developing new unsupervised learning schemes based on kernel methods particularly on least squares support vector machines (LS-SVM). Unsupervised learning corresponds to a broad family of machine learning methods with the aim of extracting useful information from data when no target or desired patterns are known or given. Typical unsupervised learning problems include feature extraction, dimensionality reduction, data visualization, density estimation and clustering. This type of problems arises commonly in fields such as statistics, engineering, cognitive science, data mining and artificial intelligence at the intersecting point of several disciplines of science and mathematics. Unsupervised learning studies how systems can learn to summarize and express key aspects from a particular set of data in such a way that it is coherent with the underlying structure of the data. In contrast with supervised or reinforcement learning, there are no explicit target outputs or associated rewards. Instead, the system learns to make representations of the input patterns and to distinguish them from unstructured noise. The first work in unsupervised learning can be traced back to Pearson in 1901 [86] with the introduction of principal component analysis (PCA) also called Hotelling transform, Karhunen-Loève transformation [71] or proper orthogonal decomposition (POD) [68] depending on the application. PCA has been extensively used particularly for dimensionality reduction and linear feature extraction. After PCA, other unsupervised learning

techniques were proposed such as canonical correlation analysis (CCA) [47, 57], independent component analysis (ICA) [27, 61], multidimensional scaling (MDS) [108] and k-means [73].

Kernel-based methodologies [29, 96] have been popularized in recent years in the fields of machine learning and data mining. The statistical learning framework proposed by Vapnik [110] led to the introduction of the support vector machine (SVM) [28, 110] which has been successfully applied for nonlinear classification and regression in learning problems with high-dimensional data. The success of the SVM led to further developments in the area of kernel-based learning such as kernel PCA [95, 97], kernel CCA [2, 70], kernel Fisher discriminant analysis (FDA) [80], kernel ridge regression [93] and ICA using kernels [14, 49]. One particular class of kernel machines are the least squares support vector machines (LS-SVM) [102, 104] aimed at simplifying classical SVM formulations and developing a wider range of learning algorithms applicable beyond classification and regression. LS-SVMs are formulated using the L_2 loss function in a constrained optimization framework with primal and dual insights resulting in a systematic and extendable methodology for kernel-based modeling particularly for unsupervised learning.

1.2 Challenges and Problems

This thesis is focused on developing new methods of unsupervised learning using LS-SVMs as core models. The current challenges and problems are described as follows:

- **Out-of-sample extensions:** Unsupervised learning methods are usually solved in terms of matrix decompositions. Although in general, these type of problems can be solved efficiently, the underlying learning model is not known. This drawback leads to unclear extensions to new points outside the training pool. Approximation techniques and ad-hoc methods are often used in order to extend the results to out-of-sample data. Schemes with clear extensions to new data are required in order to solve unsupervised problems in a learning framework.
- **Model selection:** The use of kernels in unsupervised learning incorporates a powerful degree of complexity to the methods. Well-known linear techniques can be extended to elegantly include nonlinearity into

the analysis by the use of nonlinear kernels. However in practice, the chosen model might be too powerful for the application at hand and overfitting can easily occur. For this reason, model and parameter selection are key issues in unsupervised learning problems mainly due to the lack of clear underlying learning model. Selecting the parameters is often done in an ad-hoc and heuristic manner.

- **Large-scale problems:** The amount of data available from fields such as bioinformatics, text mining, computer vision, information retrieval, process industry is increasing at an explosive rate. Large-scale problems become a challenge for unsupervised learning techniques in order to extract useful information from oceans of data. Approximation techniques, reduced set methods, subsampling schemes and sparse models are needed to deal with the large-scale data using standard hardware.

- **Incorporating prior information:** In real-life applications, additional side information is typically available together with the data. This prior information can come from expert knowledge, perceived or observed insights and even common sense. Classical unsupervised learning methods ignore this type of information resulting in too general models. It becomes important then to have a flexible methodology in order to incorporate prior information in an easy way.

1.3 Objectives

The objectives of this thesis in the context of unsupervised learning using support vector machines are summarized as follows.

1. The first objective consists of extending the basic LS-SVM methodologies for unsupervised learning towards the inclusion of more advanced elements in a constrained optimization framework. The primal-dual formulation typical of LS-SVMs allow the incorporation of key aspects relevant for unsupervised learning such as out-of-sample extensions, sparseness, robustness and prior information.

2. The second objective is related to understanding the role of regularization in unsupervised learning methods based on kernels. Usually in these methods, the solutions arise from matrix decomposition

problems such as eigenvalue equations with no clear information on how to control the complexity of the corresponding learning kernel models.

3. The third objective deals with the applicability of the proposed unsupervised learning methods to a wide variety of problems including nonlinear feature extraction, clustering and denoising. Emphasis is given to application areas such as image segmentation, time series analysis and text mining in the context of large-scale datasets.

1.4 Chapter by Chapter Overview

This thesis is organized in seven main chapters as can be seen in Figure 1.1. With the exception of chapters 2 and 3 which introduce the main concepts of kernel-based unsupervised learning and LS-SVM methodologies, all chapters in the sequel present different contributions.

The content of the chapters is organized as follows.

- Chapter 2 gives a general introduction to kernel functions, the Mercer's theorem and unsupervised learning using kernel-based methods. Classical algorithms such as kernel PCA, kernel CCA, ICA using correlation measures in the feature space and spectral clustering methods are summarized in this chapter.

- Chapter 3 presents LS-SVM formulations for kernel-based learning. LS-SVMs are a class of kernel machines aiming at extending methodologies typical of support vector machines for solving machine learning problems beyond classification, regression and unsupervised learning. The primal-dual structure of LS-SVM allow extensions of the models by e.g. adding additional constraints to the primal problem. The chapter summarizes LS-SVM formulations towards classification, regression, kernel PCA and kernel CCA. The main contributions of this thesis are built considering LS-SVMs as core models.

- Chapter 4 extends the LS-SVM formulation for kernel PCA to general underlying loss functions. The loss function underlying kernel PCA was shown to be L_2 [103]. Using the extended formulation, it is possible to plug in a different loss function in order to obtain desirable properties. In this particular case, a robust epsilon-insensitive loss

function is used to incorporate robustness and sparseness into kernel PCA at the same time. This chapter also provides two algorithms: the first solves a set of nonlinear equations using a sequential quadratic programming (SQP) approach while the second corresponds to a quick scheme leading to solving a sequence of generalized eigenvalue problems. Experimental results with toy data and handwritten digit denoising are presented.

- Chapter 5 extends kernel CCA based on LS-SVMs to the multivariate case when more than 2 datasets are given. The proposed formulation leads to a kernel regularized correlation (KRC) measure in high dimensional feature spaces induced by kernels. The regularization is incorporated naturally in the primal problem leading to a dual generalized eigenvalue problem avoiding overfitting problems known in the context of kernel CCA. If the kernels used are universal, then the KRC also characterize pairwise independence in the input space. In this way, the KRC can be used as a contrast function for ICA. This chapter also discusses computational issues due to the large size of the matrices involved which are tackled used the incomplete Cholesky decomposition [48] by computing low-rank approximations of the kernel matrices. Practical examples are given.

- Chapter 6 presents a multiway spectral clustering method based on a weighted LS-SVM formulation to kernel PCA. In the first part of this chapter, a summary of classical methods for spectral clustering is given. It continues towards formulating a scheme for binary spectral clustering using weighted kernel PCA. This formulation is then extended to the multiway case by adding additional variables and bias terms in a setting related to multiclass theory. One of the main advantages of the proposed methodology is the possibility to extend the clustering model to out-of-sample data which becomes important for generalization and predictive purposes. A model selection criterion called the balanced line fit (BLF) is also presented in this chapter. This criterion exploits the structure of the eigenvectors and the corresponding projections when the clusters are well-formed. The spectral clustering problem can then be solved in a learning framework with model using the proposed methodology. Toy examples are given together with real-life applications of image segmentation, scientific journal clustering and power load time series clustering.

- Chapter 7 extends the spectral clustering method introduced in the

previous chapter to a sparse kernel model. For large-scale problems, the cost of storing and computing an eigendecomposition of the matrices involved can become prohibitive. Moreover, the cluster indicators are obtained from the projections onto the eigenvectors which are typically expressed in terms of dense kernel expansions where every training data point contributes. The proposed sparse kernel model is based on the incomplete Cholesky decomposition used previously in chapter 5 for computing low-rank approximations. Examples with large-scale toy data are given.

- Chapter 8 presents another extension of the clustering method described in chapter 6. It extends the multiway spectral clustering scheme towards including prior knowledge into the analysis. Using this kind of side information together with unsupervised clustering can be interpreted as a type of semi-supervised learning. Prior knowledge is typically available in the form of pairwise constraints on the cluster assignment such as must-link and cannot-link constraints. The proposed formulation leads to a modification of the metric causing regularized rank-1 downdates of the kernel matrix. Experimental examples with toy data and image segmentation problems are given.

1.5 Contributions of this work

The main contributions of this thesis are summarized as follows.

- **Learning framework for kernel-based unsupervised learning** We have shown and described a common learning framework for kernel-based unsupervised learning. The modularity and extendability of the proposed methodologies allow solving unsupervised learning problems in a constrained optimization framework with out-of-sample extensions and model selection schemes which correspond to key aspect in terms of predictive purposes and generalization.
- **Kernel Component Analysis (KCA)** We propose a formulation for generalized kernel PCA where the loss function is explicit and can easily be changed. By using a robust epsilon-insensitive loss function, we have incorporated robustness and sparseness at the same time into the formulation. These two properties are missing in classical kernel PCA due to the underlying L_2 loss function. Chapter 4.

[4] C. Alzate and J. A. K. Suykens. Extending kernel principal component analysis to general underlying loss functions. *In Proceedings of the International Joint Conference on Neural Networks (IJCNN05)*, pages 214-219, 2005.

[8] C. Alzate and J. A. K. Suykens. Kernel component analysis using an epsilon insensitive robust loss function. *IEEE Transactions on Neural Networks*, 19(9):1583-1598, 2008.

- **Kernel Regularized Correlation (KRC)** Typically, kernel CCA approaches need ad-hoc regularization in order to obtain useful estimates of the canonical correlation in high dimensional feature spaces. The KRC corresponds to the dual solution of a primal optimization problem where regularization is explicit into the formulation. This insight leads to algorithms which are better conditioned. In the case of universal kernels, the KRC can also be used to characterize independence, thus serving as a contrast function for ICA. Chapter 5.

  [6] C. Alzate and J. A. K. Suykens. ICA through an LS-SVM based kernel CCA measure for independence. *In Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN07)*, pages 2920–2925, 2007.

  [9] C. Alzate and J. A. K. Suykens. A regularized kernel CCA contrast function for ICA. *Neural Networks*, 21(2-3):170–181, 2008.

- **Multiway spectral clustering with out-of-sample extensions** Spectral clustering methods are expressed as relaxations of graph partitioning problems and usually lead to eigenvalue decompositions of Laplacian matrices. While eigenvalue problems can be solved efficiently, the model underlying the clustering is not known in general. We provide with this contribution a multiway spectral clustering method based on weighted kernel PCA which can be used to extend the clustering model to out-of-sample data. The out-of-sample extension also allows model selection in order to avoid kernel models that are too general for the problem at hand. Chapter 6.

  [5] C. Alzate and J. A. K. Suykens. A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. *In Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN06)*, pages 138–144, 2006.

[7] C. Alzate and J. A. K. Suykens. Image segmentation using a weighted kernel PCA approach to spectral clustering. *In Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)*, pages 208–213, 2007.

[11] C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. In Press, 2009.

- **Spectral clustering extensions: towards sparseness and semi-supervised learning** We have implemented extensions for spectral clustering towards issues such as sparseness and the incorporation of prior information. The primal-dual interpretation of the proposed spectral clustering approach allows easy extensions by adding additional constraints at the primal level. Sparseness is important for large-scale data and is tackled in this contribution using the incomplete Cholesky decomposition and reduced set methods. Incorporating pairwise constraints in the primal leads to an equivalent regularized kernel function in the dual. Chapters 7 and 8.

  [10] C. Alzate and J. A. K. Suykens. Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition. *In Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN 2008)*, pages 3555–3562, 2008.

  [3] C. Alzate and J. A. K. Suykens. A regularized formulation for spectral clustering with pairwise constraints. Internal Report 09-07, ESAT-SISTA, K.U.Leuven (Leuven, Belgium). *Accepted for publication in IJCNN 2009*.

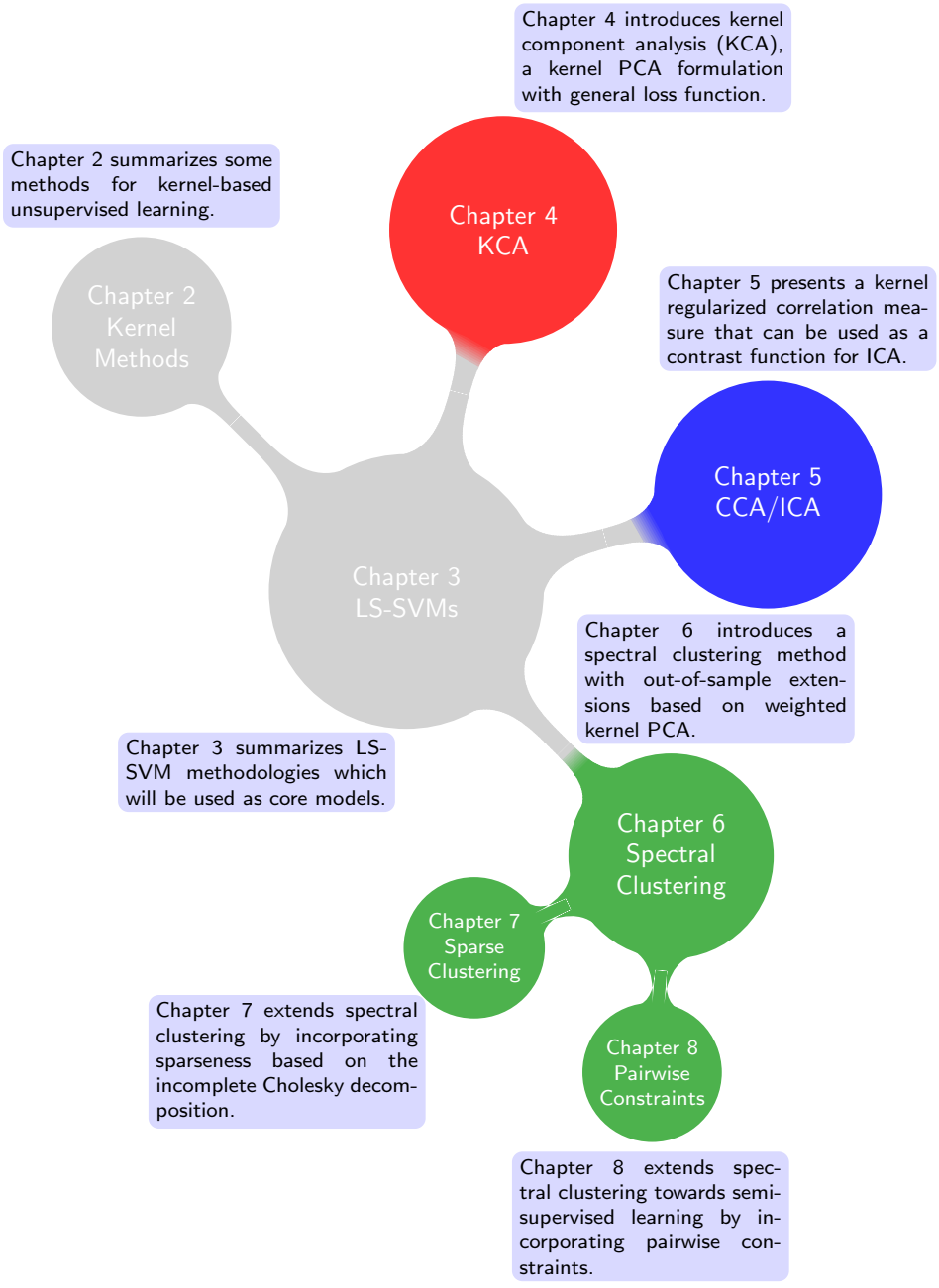


Figure 1.1: Structure of the thesis. Chapters $4, 5, 6, 7$ and 8 constitute the main contributions of this thesis and are depicted in dark colors.

Chapter 2

Kernel Methods for Unsupervised Learning

This chapter aims at introducing the main concepts in kernel-based methods particularly to unsupervised learning problems. Kernel functions have been studied in different areas such as integral equation analysis, estimation in reproducing kernel Hilbert space (RKHS) and Gaussian processes. Fundamental mathematical contributions have been made by Mercer [79], Aronszajn [13] and Schoenberg [94]. The first use of kernel functions in pattern recognition problems was described in Aizerman et al. [1]. The main idea of kernel methods in machine learning is to perform linear estimation in a high dimensional (even infinite dimensional) feature space, which is nonlinearly related to the original input space. The kernel function induces a high dimensional feature space, which is never computed explicitly via the application of Mercer's theorem. The successful application of kernel methods to high-dimensional input data has led to an increasing interest of these learning formulations [96]. Classical examples of kernel methods especially in the area of unsupervised learning include kernel principal component analysis (PCA) [97], kernel CCA [70], kernel ICA [14, 49] and kernel k-means [97]. This chapter is organized as follows. Section 2.1 introduces the notion of positive definite kernels and the Mercer's theorem. Section 2.2 shows the derivation of kernel PCA. In Section

2.3, a classical kernelized version of CCA is discussed. Section 2.4 contains a summary of commonly used correlation and covariance functions in RKHS to perform ICA and in Section 2.5, classical spectral clustering is presented.

2.1 Kernel Methods

2.1.1 Mercer kernels

A positive definite kernel (also called Mercer kernel) on a space $\mathcal{X}$ is defined as a symmetric continuous function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for any set of points $\{x_1, \ldots, x_N\} \in \mathcal{X}$, the $N \times N$ kernel matrix Ω defined by $\Omega_{ij} = K(x_i, x_j), i, j = 1, \ldots, N$ is positive definite. A Mercer kernel induces a high dimensional feature space $\mathcal{F}$ and a mapping $\varphi(\cdot)$ from $\mathcal{X}$ to $\mathcal{F}$ such that $K(x, z)$ equals the dot product between $\varphi(x)$ and $\varphi(z)$. The latter property was introduced in 1909 in Mercer's theorem:

Theorem 2.1. ***Mercer [79]*** *Suppose $\mathcal{X} \subseteq \mathbb{R}^d$ and K is a symmetric continuous function such that the integral operator*

$$T_K : L_2(\mathcal{X}) \to L_2(\mathcal{X})$$
$$(T_K f)(\cdot) = \int_{\mathcal{X}} K(\cdot, x) f(x) dx$$

is positive, which means that for all $f \in L_2(\mathcal{X})$ we have

$$\int_{\mathcal{X} \times \mathcal{X}} K(x, z) f(x) f(z) dx dz \geq 0.$$

Let $\phi_j \in L_2(\mathcal{X})$ be normalized orthogonal eigenfunctions of T_K with corresponding eigenvalues $\lambda_j > 0$. Then $K(x, z)$ can be expanded as

$$K(x, z) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(z).$$

For n_h finite, the last equation can be written as

$$K(x, z) = \sum_{j=1}^{n_h} \sqrt{\lambda_j} \phi_j(x) \sqrt{\lambda_j} \phi_j(z) = \varphi(x)^T \varphi(z) \tag{2.1}$$

where $\varphi(x) = \left[\sqrt{\lambda_1}\phi_1(x), \sqrt{\lambda_2}\phi_2(x), \ldots, \sqrt{\lambda_{n_h}}\phi_{n_h}(x)\right]$. Thus, a positive definite kernel induces a nonlinear mapping $\varphi(\cdot)$ to a high dimensional feature space $\mathcal{F} \in \mathbb{R}^{n_h}$ (which can be infinite dimensional) without explicitly computing this space. The kernel evaluation provides the dot product of the mapped points in $\mathcal{F}$.

Consider as a simple example, computing the Euclidean distance of the points x and z in a kernel-induced feature space: $||\varphi(x)-\varphi(z)||_2$. Expanding the norm, we get $\big(\varphi(x)^T\varphi(x) - 2\varphi(x)^T\varphi(z) + \varphi(z)^T\varphi(z)\big)^{1/2}$ and applying (2.1) leads to $\big(K(x,x) - 2K(x,z) + K(z,z)\big)^{1/2}$. The use of (2.1) for interpreting kernels as dot products in a feature space was coined as *the kernel trick* and introduced into pattern recognition in 1964 by [1].

After the SVM, classical linear algorithms such as PCA, Fisher discriminant analysis (FDA), k-means, and ridge regression have been *kernelized*.

2.1.2 Support Vector Machines for Classification

The kernel trick has been extensively used for creating non-linear versions of well-known linear algorithms. A typical example is the support vector machine (SVM) [28, 110] which is currently a state-of-the-art method for classification and regression problems. The SVM is a maximum-margin linear classifier in a high-dimensional kernel-induced feature space and corresponds to a non-linear decision boundary in the original input space. The goal in classification consists of estimating a model of the form:

$$y(x) = \text{sign}(w^T\varphi(x) + b),$$

given a set of training data $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ are the input data and $y_i \in \{-1, 1\}$ are the class labels. The convex primal problem of the SVM as introduced by Vapnik [110] is formulated as:

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}w^Tw + c_0\sum_{i=1}^N \xi_i \\ \text{such that} \quad & y_i\big(w^T\varphi(x_i) + b\big) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, N. \end{aligned} \tag{2.2}$$

where the objective function (2.2) is a trade-off between maximizing the margin $\frac{2}{||w||_2}$ which is related to generalization and minimizing the

misclassifications represented by the slack variables ξ_i. The trade-off is controlled through the regularization constant c_0. The dual problem to this constrained optimization problem can be obtained by computing the Karush-Kuhn-Tucker (KKT) conditions for optimality of the Lagrangian and applying the kernel trick $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j), i, j = 1, \ldots, N$. The dual is then given by the following quadratic programming (QP) problem to be solved in terms of the Lagrange multipliers α_i:

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{such that} \quad & \begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c_0, i = 1, \ldots, N \end{cases} \end{aligned} \tag{2.3}$$

The classification model in the dual can now be expressed in terms of the Lagrange multipliers:

$$y(x) = \text{sign}\left(\sum_{s=1}^{N_{\text{SV}}} \alpha_s y_s K(x, x_s) + b \right)$$

where $N_{\text{SV}} \leq N$ is the number of *support vectors* which are training points with corresponding nonzero α_i. This sparseness property on the dual model follows from solving the QP problem (2.3).

2.1.3 Kernel functions

Typical choices of positive definite kernel functions when $\mathcal{X} \subseteq \mathbb{R}^d$ are:

- Linear kernel: $K(x, z) = x^T z$.
- Polynomial kernel: $K(x, z) = (x^T z + c_p)^{d_p}$, where $d_p \in \mathbb{N}$ is the degree and $c_p \geq 0$. The feature space induced by the polynomial kernel has dimensionality $\binom{d+d_p}{d_p}$ [110].
- Radial Basis Function (RBF) kernel: $K(x, z) = \exp(-||x - z||_2^2 / 2\sigma^2)$ where $\sigma \in \mathbb{R}$ is the kernel width. The feature space induced by the RBF kernel is infinite-dimensional for every value of σ [110].

Remark 2.1 (Parameter tuning)**.** *The kernel parameters (e.g. c_p, d_p, σ) should be chosen carefully in order to ensure good generalization. This is*

particularly important for very flexible kernels such as the RBF kernel where too small values of σ will result in ill-conditioning and overfitting. On the other hand, very large values of σ will make the RBF kernel function behave like a constant function and hence providing no information.

Remark 2.2 (Centering in the feature space)**.** *The kernel trick can also be used to move the center of mass of the mapped points to the origin. Consider a set of N data points $\{x_i\}_{i=1}^N$ and the corresponding mean vector in the feature space $\hat{\mu}_\varphi = (1/N)\sum_{i=1}^N \varphi(x_i)$. The centered feature vector of the i-th data point becomes $\varphi(x_i) - \hat{\mu}_\varphi$. Hence, the kernel in this transformed space is*

$$\begin{aligned} K_c(x_i, x_j) &= (\varphi(x_i) - \hat{\mu}_\varphi)^T(\varphi(x_j) - \hat{\mu}_\varphi) \\ &= K(x_i, x_j) - \frac{1}{N}\sum_{r=1}^{N} K(x_i, x_r) - \frac{1}{N}\sum_{r=1}^{N} K(x_j, x_r) \\ &\quad + \frac{1}{N^2}\sum_{r,s=1}^{N} K(x_r, x_s), \end{aligned}$$

and expressing this operation in matrix form leads to the centered kernel matrix Ω_c

$$\Omega_c = M_c \Omega M_c \tag{2.4}$$

where $M_c = I_N - (1/N)1_N 1_N^T$ is the centering matrix, I_N is the $N \times N$ identity matrix and 1_N is an $N-$dimensional vector of ones.

Remark 2.3 (Normalized kernels)**.** *Consider the cosine of the angle of the points x and z in the input space: $cos(\theta) = x^T z/(||x||_2||z||_2)$. Performing this operation in the feature space leads to:*

$$cos(\theta_\varphi) = \frac{\varphi(x)^T\varphi(z)}{||\varphi(x)||_2||\varphi(z)||_2} = \frac{K(x,z)}{\sqrt{K(x,x)K(z,z)}} = K_n(x,z)$$

where $K_n(x, z)$ denotes a normalized kernel. Normalizing the kernel matrix can be achieved via

$$\Omega_n = M_n \Omega M_n$$

where $M_n = diag\big([1/\sqrt{\Omega_{11}}, 1/\sqrt{\Omega_{22}}, \ldots, 1/\sqrt{\Omega_{NN}}]\big)$. Normalized kernels play a key role for robustness in the context of robust statistics [25].

2.2 Kernel PCA

In this Section, a summary of linear and kernel PCA is presented. Kernel PCA has been extensively used as a natural nonlinear extension of linear PCA in applications such as nonlinear feature extraction, pattern denoising, compression and density estimation. One of the main advantages of the kernel PCA formulation lies in the fact that it follows from an eigendecomposition in a similar way as linear PCA. This differs from other methods for non-linear PCA where the solutions are obtained from iterative schemes which suffer from local minima.

2.2.1 Linear PCA

The classical linear PCA considers a given set of zero mean data $\{x_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$. The objective is to find directions w in which the projected variables $w^T x_i$ have maximal variance. The directions are assumed to have unitary norm leading to the following constrained maximization problem:

$$\max_w \frac{1}{N}\sum_{i=1}^{N}(w^T x_i)^2 = w^T C w$$
$$\text{such that } w^T w = 1$$

where $C = (1/N)\sum_{i=1}^N x_i x_i^T$ is the sample covariance matrix. The Lagrangian for this constrained optimization problem is given by:

$$\mathcal{L}(w;\lambda) = \frac{1}{2} w^T C w - \lambda(w^T w - 1)$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. Applying the conditions for optimality $\partial\mathcal{L}/\partial w = 0$, $\partial\mathcal{L}/\partial\lambda = 0$ leads to the symmetric eigenvalue problem:

$$Cw = \lambda w. \tag{2.5}$$

The eigenvector w corresponding to the largest eigenvalue defines the direction in which the projected variables have maximal variance. Subsequent eigenvectors and eigenvalues are usually found by deflating the data matrix or by adding orthogonality constraints with respect to the previously found eigenvectors [65].

2.2.2 PCA in Feature Spaces

The kernel PCA algorithm for non-linear PCA was described in Schölkopf et al. [97]. This method performs linear PCA in a kernel-induced feature space $\mathcal{F}$ nonlinearly related to the original input space. In this case, consider finding directions in which the projected variables in the feature space $w^T\varphi(x_i), i = 1, \ldots, N$ have maximal variance. Assuming zero mean mapped data $\sum_{i=1}^{N} \varphi(x_i) = 0$, the eigendecomposition of the sample covariance matrix in the feature space C_φ becomes

$$C_\varphi w_\varphi = \lambda w_\varphi \tag{2.6}$$

with

$$C_\varphi = \frac{1}{N}\sum_{i=1}^{N} \varphi(x_i)\varphi(x_i)^T = \frac{1}{N}\Phi^T\Phi \tag{2.7}$$

where Φ is the $N \times n_h$ feature matrix:

$$\Phi = \begin{bmatrix} \varphi(x_1)^T \\ \varphi(x_2)^T \\ \vdots \\ \varphi(x_N)^T \end{bmatrix}.$$

The computation of $\varphi(\cdot)$ and C_φ is complicated due to the high dimensionality of $\mathcal{F}$. As noted in [97], the eigenvectors w_φ can be expressed as linear combinations of the mapped data points Φ, hence there exist coefficients $a \in \mathbb{R}^N$ such that $w_\varphi = \Phi^T a$. Eliminating w_φ, premultiplying (2.6) by Φ and applying Mercer's theorem leads to following eigendecomposition:

$$\Omega a = N\lambda a \tag{2.8}$$

where $\Omega = \Phi\Phi^T$ is the kernel matrix with ij-th entry $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$. Typically, the data are preprocessed in the feature space by removing the mean. This step is accomplished by centering the kernel matrix (Remark 2.2). Thus, the kernel PCA eigenvalue problem becomes

$$\Omega_c a = N\lambda a \tag{2.9}$$

where Ω_c is the centered kernel matrix. The resulting eigenvectors are normalized in $\mathcal{F}$ by requiring $||a^{(j)}||_2 = 1/\sqrt{\lambda_j}, j = 1, \ldots, N$. The

projections of an out-of-sample point x onto an eigenvector can be computed as:

$$z(x) = \sum_{i=1}^{N} a_i K(x_i, x).$$

Even though the mapping $\varphi(\cdot)$ from the input space to the feature space is one of the fundamental ideas in kernel methods, the mapping back from the feature space to the input space become useful in certain applications especially denoising with kernel PCA. Consider a point ϕ in $\mathcal{F}$, the pre-image problem can be defined as finding a data point $x \in \mathcal{X}$ such that $\phi = \varphi(x)$. Exact pre-images typically do not exist (when they exist, they might not be unique) [69, 81]. However, approximate pre-images can be found by solving the following nonlinear optimization problem:

$$z^\star = \arg\min_z ||\phi - \varphi(z)||_2^2, \tag{2.10}$$

where $z^\star \in \mathcal{X}$ is the approximate pre-image of ϕ. Several schemes to solve (2.10) have been discussed in the literature including a fixed-point iteration method for RBF kernels [81] together with a robust extension [105] and a non-iterative algebraic scheme based on multi-dimensional scaling (MDS) [69]. Figure 2.1 shows a simple illustration of the pre-image problem.

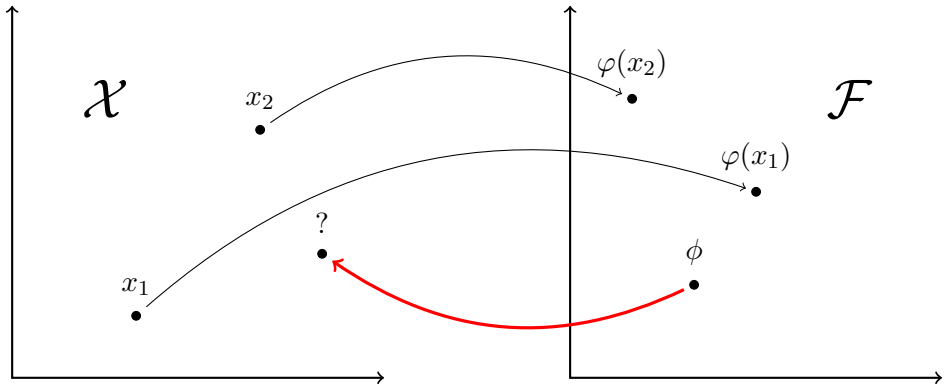


Figure 2.1: The pre-image problem. In certain applications, it is also useful to map back from the feature space to the input space. Not every point ϕ in $\mathcal{F}$ that can be written as linear combinations of mapped points has necessarily an exact pre-image in $\mathcal{X}$. Thus, it is only possible to find approximate pre-images.

Although, the pre-image problem have been studied mainly in denoising applications of kernel PCA, it is also useful in other cases such as visualization of centroids in kernel-based clustering methods. Figure 2.2 shows a toy example of denoising kernel PCA with a nonlinear kernel. The dataset consists of $N = 1,000$ training points depicted as gray circles. Denoised data points are depicted as black crosses. In linear PCA, the maximum number of components that can be extracted is $d = 2$ while in kernel PCA it is possible to extract up to N components. The plot shows the denoised variables using the first principal component for linear PCA and the denoised points using 10 components using kernel PCA with the RBF kernel $\sigma = 0.5$ and the pre-image method [81]. Note that, the nonlinear shape of the dataset is lost in PCA.

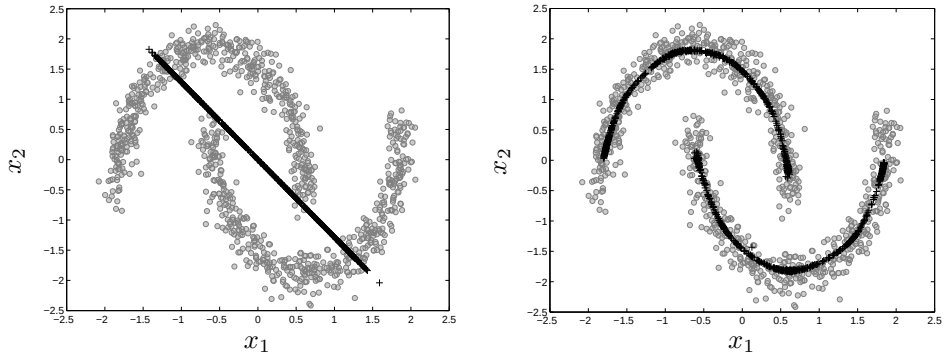


Figure 2.2: Denoising using linear and kernel PCA. The gray circles correspond to the training data points and the black crosses correspond to the denoised variables. **Left:** Linear PCA denoising using the first principal component. **Right:** Kernel PCA with RBF kernel $\sigma = 0.5$ using 10 components and pre-image algorithm [81]. Note that in linear PCA the nonlinear structure of the data is lost.

2.3 Kernel CCA

Canonical correlation analysis is useful for finding relations between two representations of the same underlying concept. This unsupervised learning method has been applied in econometrics, image-text retrieval and subspace identification. A summary of classical CCA and kernel CCA is given as follows.

2.3.1 Linear CCA

The problem of CCA consists of measuring the linear relationship between two sets of variables [47, 57]. Let $\{x_i^{(1)}\}_{i=1}^N$ and $\{x_i^{(2)}\}_{i=1}^N$ denote N observations of $x^{(1)}$ and $x^{(2)}$. The observations are assumed to have zero mean. Then, the objective of CCA is to find basis vectors $w^{(1)}$ and $w^{(2)}$ such that the projected variables ${w^{(1)}}^T x^{(1)}$ and ${w^{(2)}}^T x^{(2)}$ are maximally correlated:

$$\max_{w^{(1)},w^{(2)}} \rho = \frac{{w^{(1)}}^T C_{12} w^{(2)}}{\sqrt{{w^{(1)}}^T C_{11} w^{(1)}}\sqrt{{w^{(2)}}^T C_{22} w^{(2)}}} \tag{2.11}$$

where $C_{11} = (1/N)\sum_{i=1}^N x_i^{(1)} {x_i^{(1)}}^T, C_{22} = (1/N)\sum_{i=1}^N x_i^{(2)} {x_i^{(2)}}^T, C_{12} = (1/N)\sum_{i=1}^N x_i^{(1)} {x_i^{(2)}}^T$. This is typically formulated as the optimization problem:

$$\max_{w^{(1)},w^{(2)}} \quad {w^{(1)}}^T C_{12} w^{(2)} \tag{2.12}$$

$$\text{such that} \quad \begin{cases} {w^{(1)}}^T C_{11} w^{(1)} = 1 \\ {w^{(2)}}^T C_{22} w^{(2)} = 1 \end{cases} \tag{2.13}$$

which leads to the generalized eigenvalue problem:

$$\left[\begin{array}{c|c} 0 & C_{12} \\ \hline C_{21} & 0 \end{array}\right] \left[\begin{array}{c} w^{(1)} \\ \hline w^{(2)} \end{array}\right] = \rho \left[\begin{array}{c|c} C_{11} & 0 \\ \hline 0 & C_{22} \end{array}\right] \left[\begin{array}{c} w^{(1)} \\ \hline w^{(2)} \end{array}\right]$$

where ρ is the correlation coefficient. This problem has $d_1 + d_2$ eigenvalues $\{\rho_1, -\rho_1, \ldots, \rho_p, -\rho_p, 0, \ldots, 0\}$, where $p = \min(d_1, d_2)$.

An alternative scheme is to find the smallest eigenvalue of the following generalized eigenvalue problem

$$\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right] \left[\begin{array}{c} w^{(1)} \\ \hline w^{(2)} \end{array}\right] = (1+\rho) \left[\begin{array}{c|c} C_{11} & 0 \\ \hline 0 & C_{22} \end{array}\right] \left[\begin{array}{c} w^{(1)} \\ \hline w^{(2)} \end{array}\right].$$

The advantage of this alternative formulation lies in the fact that the correlation measure now ranges from zero to one. In general, CCA is defined only for two sets of variables. A multivariate extension of CCA towards m sets of variables was introduced in [67] and coined *generalized* CCA.

2.3.2 CCA in kernel-induced feature spaces

A nonlinear extension of CCA using kernels was introduced in [14, 70]. The data is first embedded into a high dimensional feature space induced by a kernel and then linear CCA is applied. Consider two sets of data $\{x_i^{(1)}\}_{i=1}^N, x_i^{(1)} \in \mathbb{R}^{d_1}$ and $\{x_i^{(2)}\}_{i=1}^N, x_i^{(2)} \in \mathbb{R}^{d_2}$ with feature maps $\varphi^{(1)}(\cdot) : \mathbb{R}^{d_1} \to \mathbb{R}^{n_1}$, $\varphi^{(2)}(\cdot) : \mathbb{R}^{d_2} \to \mathbb{R}^{n_2}$ to high dimensional feature spaces of dimension n_1 and n_2 respectively. The centered feature matrices $\Phi_c^{(1)} \in \mathbb{R}^{N \times n_1}, \Phi_c^{(2)} \in \mathbb{R}^{N \times n_2}$ become

$$\begin{aligned} \Phi_c^{(1)} &= [\varphi^{(1)}(x_1^{(1)})^T - \hat{\mu}_{\varphi^{(1)}}^T; \ldots; \varphi^{(1)}(x_N^{(1)})^T - \hat{\mu}_{\varphi^{(1)}}^T] \\ \Phi_c^{(2)} &= [\varphi^{(2)}(x_1^{(2)})^T - \hat{\mu}_{\varphi^{(2)}}^T; \ldots; \varphi^{(2)}(x_N^{(2)})^T - \hat{\mu}_{\varphi^{(2)}}^T], \end{aligned}$$

where $\hat{\mu}_{\varphi^{(l)}} = (1/N)\sum_{i=1}^N \varphi^{(l)}(x_i^{(l)}), l = 1, 2$. The canonical correlation in the feature space can then be written as:

$$\begin{aligned} \max_{w^{(1)}, w^{(2)}} \quad & w^{(1)^T} \Phi_c^{(1)^T} \Phi_c^{(2)} w^{(2)} \\ \text{such that} \quad & \begin{cases} w^{(1)^T} \Phi_c^{(1)^T} \Phi_c^{(1)} w^{(1)} = 1 \\ w^{(2)^T} \Phi_c^{(2)^T} \Phi_c^{(2)} w^{(2)} = 1, \end{cases} \end{aligned} \tag{2.14}$$

the projection vectors $w^{(1)}$ and $w^{(2)}$ are assumed to lie in the span of the mapped data:

$$\begin{aligned} w^{(1)} &= \Phi_c^{(1)^T} \beta^{(1)} \\ w^{(2)} &= \Phi_c^{(2)^T} \beta^{(2)}. \end{aligned}$$

Applying the kernel trick results in the following optimization problem:

$$\begin{aligned} \max_{\beta^{(1)}, \beta^{(2)}} \quad & \beta^{(1)^T} \Omega_c^{(1)} \Omega_c^{(2)} \beta^{(2)} \\ \text{such that} \quad & \begin{cases} \beta^{(1)^T} \Omega_c^{(1)} \Omega_c^{(1)} \beta^{(1)} = 1 \\ \beta^{(2)^T} \Omega_c^{(2)} \Omega_c^{(2)} \beta^{(2)} = 1 \end{cases} \end{aligned} \tag{2.15}$$

where $\Omega_c^{(l)}$ denotes the l-th centered kernel matrix for $l = 1, 2$. The problem in (2.15) can be written as the generalized eigenvalue problem:

$$\left[\begin{array}{c|c} 0 & \Omega_c^{(1)} \Omega_c^{(2)} \\ \hline \Omega_c^{(2)} \Omega_c^{(1)} & 0 \end{array}\right] \left[\begin{array}{c} \beta^{(1)} \\ \hline \beta^{(2)} \end{array}\right] = \rho \left[\begin{array}{c|c} \Omega_c^{(1)} \Omega_c^{(1)} & 0 \\ \hline 0 & \Omega_c^{(2)} \Omega_c^{(2)} \end{array}\right] \left[\begin{array}{c} \beta^{(1)} \\ \hline \beta^{(2)} \end{array}\right] \tag{2.16}$$

Remark 2.4 (Regularization in Kernel CCA)**.** *The non-zero solutions to* (2.16) *are always* $\rho = \pm 1$*. This is due to the fact that the angle between the subspaces spanned by the columns of the kernel matrices* $\Omega_c^{(1)}$ *and* $\Omega_c^{(2)}$ *is always equal to zero independent of the data. Hence, the canonical correlation estimate is always equal to one [14]. This issue can also be interpreted as ill-conditioning since the kernel matrices can be singular or near-singular. In general, kernel CCA methods need regularization in order to obtain useful estimators of the canonical correlation in high dimensional feature spaces. A particular* ad-hoc *approach proposed in [14] consists of finding the eigenvector corresponding to the smallest eigenvalue of the following problem:*

$$\begin{array}{l} \left[\begin{array}{c|c} (\Omega_c^{(1)} + \frac{Nc_1}{2} I_N)^2 & \Omega_c^{(1)}\Omega_c^{(2)} \\ \hline \Omega_c^{(2)}\Omega_c^{(1)} & (\Omega_c^{(2)} + \frac{Nc_1}{2} I_N)^2 \end{array}\right] \left[\begin{array}{c} \beta^{(1)} \\ \hline \beta^{(2)} \end{array}\right] = \\ \xi \left[\begin{array}{c|c} (\Omega_c^{(1)} + \frac{Nc_1}{2} I_N)^2 & 0 \\ \hline 0 & (\Omega_c^{(2)} + \frac{Nc_1}{2} I_N)^2 \end{array}\right] \left[\begin{array}{c} \beta^{(1)} \\ \hline \beta^{(2)} \end{array}\right] \end{array} \tag{2.17}$$

where c_1 *is small positive regularization constant depending of the sample size.*

2.4 ICA via the kernel trick

Independent component analysis corresponds to a broad concept of finding factors underlying the data. Classical applications include signal processing, noise reduction, biomedical data analysis and blind source separation. ICA is related to PCA in the sense that statistical independence implies uncorrelatedness. However, the contrary does not always hold. PCA is typically used as a preprocessing step in many ICA algorithms. This Section summarizes a different use of the kernel trick in order to perform ICA.

2.4.1 Preliminaries

Consider the following statistical model:

$$Y = MS,$$

where S is a $p \times N$ matrix representing N identically distributed observations of p independent components, Y is the $p \times N$ matrix of mixtures and M

is a $p \times p$ mixing matrix assumed to be invertible. The ICA problem [12, 23, 27, 61, 63] consists of estimating S based solely on Y. If the distributions of the components are specified, maximum likelihood can be used to estimate the ICA parametric model [23]. However, in practice, the distributions of the components are usually not known and the ICA problem is tackled using an estimate of the demixing matrix $W = M^{-1}$ allowing the estimation of independent components via $\hat{S} = \hat{W}Y$ [61]. A typical approach to ICA consists of an approximation of the mutual information between the components of $\hat{S}$ [12, 23]. However, alternative contrast functions based on higher-order moments have also been proposed.

The matrix of mixtures Y is often preprocessed by making its covariance matrix equal to the identity matrix. This preprocessing step is called *whitening* and reduces the number of parameters to be estimated because the demixing matrix $\hat{W}$ is then necessarily orthogonal [61]. Therefore, the ICA problem can be reduced to finding an orthogonal matrix $\hat{W}$ such that the components of $\hat{S}$ are independent.

2.4.2 Kernel ICA

The first work on using criteria to measure independence based on functions in reproducing kernel Hilbert spaces was described in Bach & Jordan [14]. These criteria for independence are based on kernel canonical correlation with the $\mathcal{F}$-correlation as a kernel dependence functional. This functional establishes a link between canonical correlation in feature spaces induced by universal kernels and independence in the original input space. Let $x^{(1)}$ and $x^{(2)}$ be random variables, then the $\mathcal{F}$-correlation can be defined as:

$$\rho_{\mathcal{F}}\big(f^{(1)}(x^{(1)}), f^{(2)}(x^{(2)})\big) = \max_{f^{(1)} \in \mathcal{F}^{(1)}, f^{(2)} \in \mathcal{F}^{(2)}} \operatorname{corr}\big(f^{(1)}(x^{(1)}), f^{(2)}(x^{(2)})\big)$$

where $f^{(l)}(x^{(l)}) = \langle \varphi^{(l)}(x^{(l)}), f^{(l)} \rangle, l = 1, 2$ is the reproducing property of RKHSs.

In the case of universal kernels, this functional is zero if and only if $x^{(1)}$ and $x^{(2)}$ are independent. The first contrast function introduced in [14] is the kernel canonical correlation (KCC) which is a regularized estimate of the spectral norm of the correlation operator between RKHSs. The second contrast function corresponds to the kernel generalized variance (KGV) which incorporates the entire spectrum of its associated correlation operator. These functions can be used to perform ICA by looking for

demixing matrices such that the estimated sources are uncorrelated in the feature space induced by a universal kernel. Figure 2.3 shows a toy example of ICA using kernels. In this case, five source signals were mixed with a random mixing matrix. The estimated independent components were found using the KCC contrast function.

Remark 2.5 (Other kernel based independence measures)**.** *Different kernel-based measures of independence have been proposed in Gretton et al. [49]. Namely, the constrained covariance (COCO) and the kernel mutual information (KMI). The COCO corresponds to the spectral norm of the covariance operator between RKHSs while the KMI is a function of the entire spectrum of the covariance operator. The KMI can also be interpreted as an upper bound near independence on a Parzen window estimate of the mutual information. The KCC and the KGV are related to the COCO and the KMI since correlation and covariance differs only in normalization which becomes irrelevant at independence when all four measures are exactly zero. However, the COCO and the KMI do not need the ad-hoc regularization scheme otherwise essential to the KCC and the KGV.*

Remark 2.6 (Pairwise independence)**.** *In the case of more than two datasets, the $\mathcal{F}$-correlation and the covariance operator between RKHSs characterize pairwise independence. However in the context of ICA, pairwise independence is usually assumed to correspond to mutual independence provided that at most one of the sources is Gaussian [27]. Table 2.1 shows the corresponding generalized eigenvalue problems arising in the computation of the KCC, KGV, COCO and KMI contrast functions for ICA in the case of m sources.*

2.5 Spectral Clustering

Spectral clustering refers to a class of methods that make use of the eigenspectrum of a matrix derived from pairwise similarities to find useful partitions of the data. This family of algorithms can be interpreted as relaxations of combinatorial problems arising in graph partitioning problems. Spectral clustering methods have been reported to be successful in cases where classical clustering schemes such as k-means and hierarchical clustering fail. The main concepts are summarized as follows.

Measure	$A\alpha = \lambda B\alpha$		C
	A	B	
COCO KMI	$\begin{bmatrix} 0 & \Omega_c^{(1)}\Omega_c^{(2)} & \dots & \Omega_c^{(1)}\Omega_c^{(m)} \\ \Omega_c^{(2)}\Omega_c^{(1)} & 0 & \dots & \Omega_c^{(2)}\Omega_c^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_c^{(m)}\Omega_c^{(1)} & \Omega_c^{(m)}\Omega_c^{(2)} & \dots & 0 \end{bmatrix}$	$\begin{bmatrix} \Omega_c^{(1)} & 0 & \dots & 0 \\ 0 & \Omega_c^{(2)} & \dots & 0r \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Omega_c^{(m)} \end{bmatrix}$	$-\frac{1}{2}$ log
KCC KGV	$\begin{bmatrix} \tilde{R}^{(1)} & \Omega_c^{(1)}\Omega_c^{(2)} & \dots & \Omega_c^{(1)}\Omega_c^{(m)} \\ \Omega_c^{(2)}\Omega_c^{(1)} & \tilde{R}^{(2)} & \dots & \Omega_c^{(2)}\Omega_c^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_c^{(m)}\Omega_c^{(1)} & \Omega_c^{(m)}\Omega_c^{(2)} & \dots & \tilde{R}^{(m)} \end{bmatrix}$	$\begin{bmatrix} \tilde{R}^{(1)} & 0 & \dots & 0 \\ 0 & \tilde{R}^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \tilde{R}^{(m)} \end{bmatrix}$	

Table 2.1: Kernel-based measures of independence and corresponding contrast function
$(\Omega_c^{(l)} + \frac{Nc_1}{2} I_N)^2, l = 1, \dots, m$ and c_1 is a parameter that depends on the sample size. In
methods such as the incomplete Cholesky decomposition are needed in order to reduce the

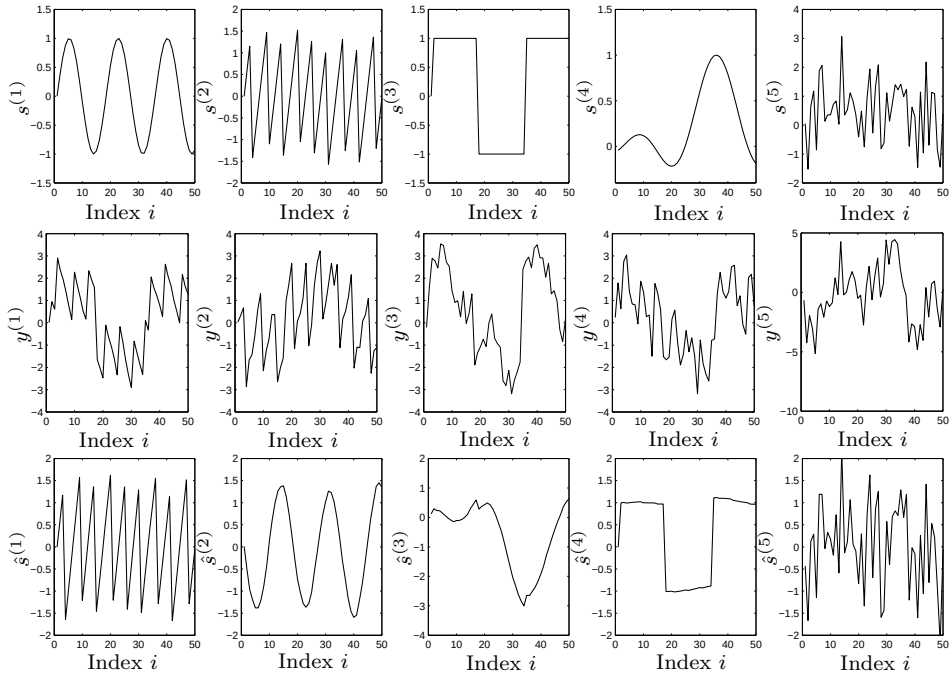


Figure 2.3: Toy example of ICA using kernels. **Top:** Original source signals $S \in \mathbb{R}^{50\times 5} = [s^{(1)}, \ldots, s^{(5)}]$. **Center:** Observed mixtures $Y = [y^{(1)}, \ldots, y^{(5)}]$ with $Y = MS$ where M is an arbitrary invertible mixing matrix. **Bottom:** Estimated independent components $\hat{S} = [\hat{s}^{(1)}, \ldots, \hat{s}^{(5)}]$ using the KCC contrast function for ICA.

2.5.1 Preliminaries

In graph theory, a set of training data points $\mathcal{D} = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$ and a similarity (also called affinity) measure $a_{ij} \in \mathbb{R}^+$ between each pair of points x_i and x_j define an undirected graph if the similarity measure is symmetric $a_{ij} = a_{ji}$. The vertices of the graph are represented by the data points and the weight of the edges is determined by a_{ij}. A classical problem is the graph bipartitioning problem which is to split the graph into two disjoint parts by cutting edges with low weight. This problem has been studied in [26, 45] and several cut criteria have been proposed to obtain meaningful partitions of the graph. Typical criteria include the normalized cut *NCut* [98] and the ratio cut *RCut* [51].

The degree of the i-th vertex is the sum of the edge weights connected to it:

$$\deg_i = \sum_{j=1}^{N} a_{ij},$$

the degree matrix D becomes the diagonal matrix $D = \text{diag}(\deg_1, \ldots, \deg_N)$. The problem of minimizing the normalized cut criterion can be written as [98]:

$$\begin{aligned} \min_{q} \quad & \frac{q^T L q}{q^T D q} \\ \text{such that} \quad & q \in \{-c_1, 1\}^N \end{aligned} \tag{2.18}$$

where $L = D - A$ is the graph Laplacian, A is the similarity matrix with ij-entry $A_{ij} = a_{ij}$, q is a clustering indicator vector and c_1 is a constant that depends of the number of data points assigned to each partition.

Minimizing the normalized cut is NP-hard. However, approximate solutions can be found by relaxing the discrete constraint on q. Letting q take real values, the relaxed solution is the eigenvector corresponding the second smallest eigenvalue (called the Fiedler vector) of the following problem:

$$L\tilde{q} = \nu D\tilde{q}$$

where $\tilde{q} \in \mathbb{R}^N$ and $\tilde{q}^T\tilde{q} = 1$. The Fiedler vector $\tilde{q}^\star$ can be binarized to obtain the clustering indicators:

$$q^\star = \text{sign}(\tilde{q}^\star - \theta)$$

where θ is a threshold to be determined.

2.5.2 Markov Random Walks

A random walk on a graph consists of random jumps from vertex to vertex. Many properties of spectral clustering methods can be expressed in terms of a transition matrix P obtained by normalizing the affinity matrix such that its rows sum to 1. The ij-th entry of P represents the probability of moving from node i to node j in one step [77, 78]. This transition matrix can be defined as $P = D^{-1}A$. The corresponding eigenvalue problem becomes

$$Pr = \xi r.$$

The eigenvectors of P are identical to the eigenvectors of the *NCut* algorithm and the eigenvalues are related by $\xi_i = 1-\nu_i, i = 1, \ldots, N$ [5]. Therefore, the Fiedler vector in this case correspond to the eigenvector with second largest eigenvalue. Maximizing the random walks algorithm can be interpreted as finding a partition of the graph in such a way that the random walk remains most of the time in the same cluster with few jumps to the other cluster.

2.5.3 The k-way *NCut* Relaxation

A more general problem is the so called k-way partioning problem. Instead of partitioning the graph into two sets, k-way cuts aim to split the graph into k disjoint sets $\mathcal{A}_1, \ldots, \mathcal{A}_k$ with $\mathcal{A}_1 \cap \mathcal{A}_2 \cap \ldots \cap \mathcal{A}_k = \emptyset$ and $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \ldots \cup \mathcal{A}_k = \mathcal{D}$.

Consider $f_l \in \{0,1\}^N$ as the cluster indicator vector for the l-th cluster such that f_l has a 1 in the entries corresponding to the data points in the l-th cluster. The cluster indicator matrix becomes $F = [f_1, \ldots, f_k] \in \{0,1\}^{N \times k}$. Hence, the k-way *NCut* can be expressed as [50]:

$$\mathrm{NCut}(\mathcal{A}_1, \ldots, \mathcal{A}_k) \quad = \quad k - \left(\frac{f_1^T W f_1}{f_1^T D f_1} + \ldots + \frac{f_k^T W f_k}{f_k^T D f_k} \right).$$

Defining

$$g_l = \frac{D^{1/2} f_l}{||D^{1/2} f_l||_2}, l = 1, \ldots, k$$

where k is the number of clusters and $G = [g_1, \ldots, g_k]$ leads to

$$\mathrm{NCut}(\mathcal{A}_1, \ldots, \mathcal{A}_k) = k - \mathrm{tr}(G^T \hat{L} G)$$

where $\hat{L}$ is the normalized Laplacian defined as $\hat{L} = D^{-1/2} A D^{-1/2}$ and $G^T G = I_k$.

Minimizing this problem is NP-hard due to the fact that G only takes discrete values. Relaxing G by allowing to take real values leads to the k-way *NCut* relaxation [50]:

$$\begin{aligned} \underset{\tilde{G}}{\mathrm{argmin}}\ \mathrm{NCut}(\mathcal{A}_1, \ldots, \mathcal{A}_k) \quad &= \quad \underset{\tilde{G}}{\mathrm{argmax}}\ \mathrm{Tr}(\tilde{G}^T \hat{L} \tilde{G}) \\ \text{such that} \qquad & \qquad \tilde{G}^T \tilde{G} = I_k. \end{aligned}$$

This minimization problem is a special form of Fan's theorem [24, 40, 44]. The optimal solution of the relaxed problem is given by:

$$\tilde{G}^{\star} = BR_2$$

where $B = [b^{(1)}, \ldots, b^{(k)}] \in \mathbb{R}^{N \times k}$ is any orthonormal basis of the k-th principal subspace of $\hat{L}$ and $R_2 \in \mathbb{R}^{k \times k}$ is an arbitrary rotation matrix [16, 44]. In other words, the relaxed clustering information is contained in the eigenvectors corresponding to the k largest eigenvalues of the normalized Laplacian $\hat{L} = D^{-1/2}AD^{-1/2}$ but the eigenvectors can have arbitrary rotations.

2.5.4 Perturbation Analysis and Piecewise Constant Eigenvectors

If a graph has k disconnected components then the transition matrix P will have k eigenvalues equal to 1. The eigenvectors corresponding to these unitary eigenvalues are the indicator functions to the respective disconnected components and the clustering is trivial because each cluster is represented as a single point in $\mathbb{R}^k$. This also corresponds to the piecewise constant property of eigenvectors discussed in [76, 78] where the spectral clustering problem is cast in a probabilistic framework using block stochastic matrices.

Definition 2.1. *[78] A piecewise constant vector $\alpha = [\alpha_1, \ldots, \alpha_N] \in \mathbb{R}^N$ relative to a partition of $\Delta = \{\mathcal{A}_p\}_{p=1}^k$ and a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ is defined as*

$$\alpha_j = c_p, \text{ for all } x_j \in \mathcal{A}_p, p = 1, \ldots, k,$$

where c_p is a constant value corresponding to the p-th cluster.

Perturbation analysis is the study of how the eigenvalues and eigenvectors of a symmetric matrix B change if an additive perturbation E is considered. The perturbed matrix is denoted by $\tilde{B} \equiv B + E$. A fundamental idea in spectral clustering is given by the Davis-Kahan theorem [32]. This theorem states that the distance between the top k eigenvalues and eigenvectors of the perturbed matrix $\tilde{B}$ and the top k eigenvalues and eigenvectors of B is bounded by $||E||_F/\delta$, where $||\cdot||_F$ denotes the Frobenius norm, $\delta = \tilde{\eta}_k - \tilde{\eta}_{k+1}$ corresponds to the additive eigengap and $\tilde{\eta}_k$ is the k-th ordered eigenvalue of $\tilde{B}$. Therefore, if the perturbation is small and the eigengap is large enough, the perturbed eigenvectors will be approximately piecewise constant and consequently easy to cluster [111].

2.5.5 From Eigenvectors to Clusters

As discussed previously, the relaxed solutions provided by spectral clustering methods are real-valued and do not provide cluster indicators. In the case of binary clustering, the relaxed solution is a particular eigenvector and the indicators can be found by binarizing the eigenvector with an appropriate threshold value. The threshold value can be determined by grid search using the original non-relaxed cut cost. However in the case of $k > 2$ clusters, the indicators cannot be determined in a straightforward manner. In [98] two methods were proposed. The first one is based on applying binary cuts in a recursive way. The current partition is subsequently divided if the *NCut* is below some prespecified value. The process is repeated until k clusters have been found. This method is slow and not optimal due to the fact that only the second smallest eigenvector is used when other eigenvectors may contain useful information. The second method called *reclustering* is one of the most popular approaches for k-way spectral clustering and consists of computing the top k eigenvectors of the k-way *NCut* problem and then applying k-means onto them. This technique works well only if the clusters in the new space represented by the eigenvectors are spherical and well-separated. Figure 2.4 shows a simple example of a classical spectral clustering method.

A different approach called *rounding* was introduced in [15]. It is based on the minimization of a metric between the relaxed solution and the set of discrete allowed solutions. This optimization problem can be solved either with respect to the partition resulting in a weighted k-means algorithm [119] or with respect to the similarity matrix yielding an algorithm for learning W given a particular partition. In [41], a method called linearized cluster assignment was discussed. This method transforms the problem of obtaining clusters from the eigenvectors into finding peaks and valley of a 1-D quantity called cluster crossing.

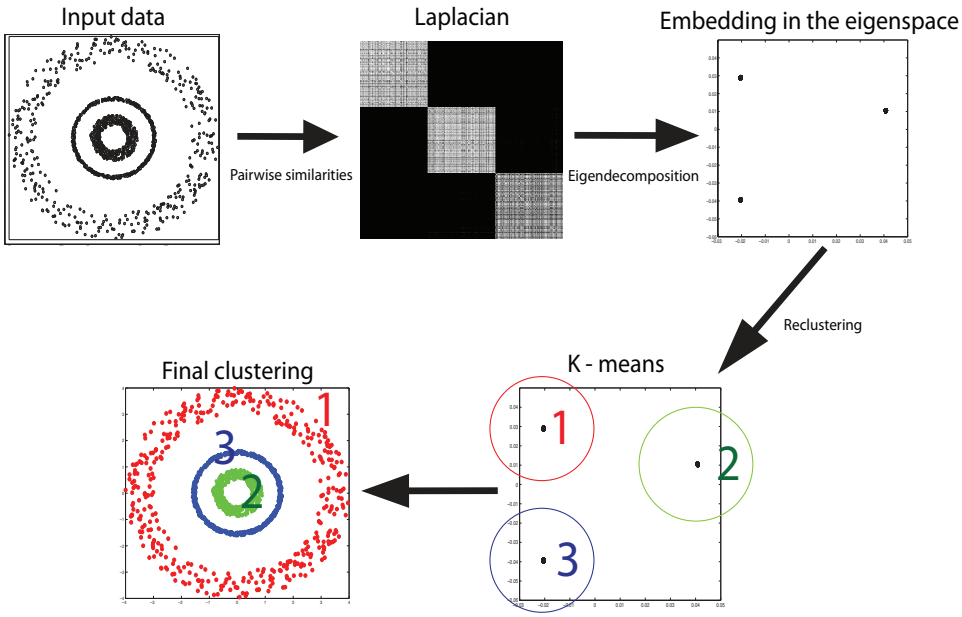


Figure 2.4: Classical spectral clustering algorithm. The eigenvectors of the Laplacian represent an eigenspace where the clusters are mapped to single points in the case of well-chosen kernel parameters. Reclustering is then applied typically using $k-$means to obtain clusters from the eigenvectors.

Chapter 3

Least Squares Support Vector Machines

Least squares support vector machine (LS-SVM) formulations to different problems were discussed in [102]. This type of kernel machines emphasizes primal-dual interpretations in terms of constrained optimization problems. LS-SVMs have been studied with respect to kernel Fisher discriminant analysis, kernel ridge regression, kernel principal component analysis, kernel canonical correlation analysis, kernel partial least-squares, recurrent networks, control and others [102, 103]. In particular, LS-SVM models aim at extending methodologies typical of support vector machines towards a wider range of machine learning problems beyond classification and regression. The primal-dual structure of LS-SVMs allow the core models to be extended by adding additional constraints in the primal formulation or by changing the loss function. This chapter summarizes LS-SVM formulations for supervised learning in the context of classification and regression and unsupervised learning techniques such as kernel PCA and kernel CCA. This chapter is structured as follows. Section 3.1 introduces the LS-SVM model for classification and discusses sparseness issues. Section 3.2 presents the LS-SVM formulation for regression and summarizes an iterative weighted scheme to achieve robustness. Sections 3.3 and 3.4 discuss formulations for unsupervised learning namely

kernel PCA and kernel CCA respectively emphasizing the role of the loss function and the regularization.

3.1 LS-SVM models for classification

LS-SVM models provide a framework for kernel-based learning in a simple and extendable way. Given a training set of data $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ are the input data and $y_i \in \{-1, 1\}$ are the class labels, the convex primal problem of the LS-SVM classifier can be formulated as

$$\begin{aligned} \min_{w,e_i,b} \quad & \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^N e_i^2 \\ \text{such that} \quad & y_i\big(w^T\varphi(x_i) + b\big) = 1 - e_i, \end{aligned} \tag{3.1}$$

$i = 1, \ldots, N$, where an L_2 loss function is used instead of the hinge loss function typical of classical SVMs. Another difference is the use of equality constraints instead of inequality constraints. The e_i are slack variables allowing deviations from the target value 1. This can be considered as a regression to the labels y_i thus, the LS-SVM classifier is related to kernel Fisher discriminant analysis (FDA) [80]. The main result is formalized in the following Lemma.

Lemma 3.1. ***[104]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x, z) = \varphi(x)^T\varphi(z)$ and a regularization constant $\gamma \in \mathbb{R}^+$, the solution to 3.1 is given by the following dual problem:*

$$\left[\begin{array}{c|c} \Omega + \frac{1}{\gamma}I_N & y \\ \hline y^T & 0 \end{array}\right]\left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} 1_N \\ \hline 0 \end{array}\right] \tag{3.2}$$

where $y = [y_1, \ldots, y_N]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$ and $\Omega_{ij} = y_i y_j K(x_i, x_j)$.

Proof: Consider the Lagrangian of the problem (3.1):

$$\mathcal{L}(w, e_i, b; \alpha_i) = \frac{\gamma}{2}\sum_{i=1}^N e_i^2 + \frac{1}{2}w^T w - \sum_{i=1}^N \alpha_i\big(y_i\big(w^T\varphi(x_i) + b\big) - 1 + e_i\big)$$

with Karush-Kuhn-Tucker conditions for optimality:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \displaystyle\sum_{i=1}^{N} \alpha_i y_i \varphi(x_i) \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, i = 1, \ldots, N \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \displaystyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow y_i\big(w^T \varphi(x_i) + b\big) - 1 + e_i = 0, i = 1, \ldots, N. \end{cases} \tag{3.3}$$

Eliminating w, e_i leads to

$$\begin{cases} \dfrac{\alpha_i}{\gamma} + \displaystyle\sum_{j=1}^{N} y_i y_j \alpha_j \varphi(x_j)^T \varphi(x_i) + b y_i = 1, i = 1, \ldots, N \\ \displaystyle\sum_{i=1}^{N} \alpha_i y_i = 0, \end{cases}$$

applying the kernel trick and writing the above equations in matrix form gives the linear system in (3.2). ■

The classifier model in the dual representation becomes:

$$y(x) = \operatorname{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \right).$$

Remark 3.1 (Sparseness)**.** *The use of the equality constraints and the L_2 loss simplifies the formulation in the LS-SVM model for classification. The LS-SVM classifier leads to solving a linear system instead of a quadratic programming (QP) problem which is the case for SVMs. However, the model is typically expressed in terms of dense kernel expansions where each training data point contributes (which can be seen from the KKT optimality condition $\alpha_i = \gamma e_i$). This differs from the classical SVM formulation where the model is formulated in terms of sparse expansions. Several methods have been developed to achieve sparse LS-SVM models. A simple scheme consists of removing data points from the training pool with corresponding small $|\alpha_i|$ value. Fixed-size LS-SVMs [102] aim at introducing sparseness by using a reduced set of training points and performing estimation in the primal via*

an approximated feature map. This is based on the Nyström method [115] which was used in the area of Gaussian processes. The selected reduced set points are obtained through a greedy scheme maximizing the quadratic Renyi entropy, hence acting as prototypes of the underlying density distribution of the data.

3.2 LS-SVM models for regression

Considering data $\{(x_i, y_i)\}_{i=1}^N$ with input data $x_i \in \mathbb{R}^d$ and output data $y_i \in \mathbb{R}$, the goal in regression is to estimate a model of the form

$$y(x) = w^T \varphi(x) + b.$$

The convex primal LS-SVM model for regression is formulated as:

$$\begin{aligned} \min_{w,e_i,b} \quad & \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^N e_i^2 \\ \text{such that} \quad & y_i = w^T \varphi(x_i) + b + e_i, \end{aligned} \tag{3.4}$$

$i = 1, \ldots, N$, where e_i represent error terms assumed to be i.i.d. with zero mean and finite variance. The dual solution is formulated in the following Lemma.

Lemma 3.2. ***[102]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x, z) = \varphi(x)^T \varphi(z)$ and a regularization constant $\gamma \in \mathbb{R}^+$, the solution to* (3.4) *is given by the following dual problem:*

$$\left[\begin{array}{c|c} \Omega + \frac{1}{\gamma} I_N & 1_N \\ \hline 1_N & 0 \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} y \\ \hline 0 \end{array}\right] \tag{3.5}$$

where $y = [y_1, \ldots, y_N]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$ and $\Omega_{ij} = K(x_i, x_j)$.

Proof: Consider the Lagrangian of the problem (3.4):

$$\mathcal{L}(w, e_i, b; \alpha_i) = \frac{\gamma}{2} \sum_{i=1}^N e_i^2 + \frac{1}{2} w^T w - \sum_{i=1}^N \alpha_i (w^T \varphi(x_i) + b + e_i - y_i)$$

with Karush-Kuhn-Tucker conditions for optimality:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \displaystyle\sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, i = 1, \ldots, N \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \displaystyle\sum_{i=1}^{N} \alpha_i = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow y_i = w^T \varphi(x_i) + b + e_i, i = 1, \ldots, N. \end{cases} \tag{3.6}$$

Applying the kernel trick and eliminating w, e_i leads to

$$\begin{cases} y_i = \displaystyle\sum_{j=1}^{N} \alpha_j K(x_i, x_j) + b + \frac{\alpha_i}{\gamma}, i = 1, \ldots, N \\ \displaystyle\sum_{i=1}^{N} \alpha_i y_i = 0, \end{cases}$$

and writing these equations in matrix form gives the linear system in (3.5). ■

The model in the dual form becomes:

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b.$$

Remark 3.2 (Relation to other methods)**.** *In the case of static regression without a bias term, it is known that kernel ridge regression [93], Gaussian processes [114] and LS-SVM lead to the same set of linear equations* (3.5)*.*

Remark 3.3 (Weighted extension for robustness)**.** *The classical SVM model for regression is known to be robust to outliers due to Vapnik's ϵ-insensitive loss function. This loss function has a bounded influence function leading to robust estimation. Although, the L_2 loss function used in LS-SVMs simplifies the model formulation, it also leads to non-robust estimation. Iterative schemes for incorporating robustness into LS-SVMs by considering weighted versions have been proposed in [34, 101, 102]. The primal problem of the*

weighted LS-SVM algorithm for regression is formulated as follows:

$$\begin{aligned} \min_{w,e_i,b} \quad & \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{N} v_i e_i^2 \\ \text{such that} \quad & y_i = w^T\varphi(x_i) + b + e_i, \end{aligned} \tag{3.7}$$

where the weights v_i in the first iteration are determined from the unweighted error variables e_i using robust estimation [58, 90]. The problem is then iterated until convergence by recomputing the error variables and the weights. In practice, few iterations have been reported to be sufficient to incorporate robustness for most problems.

3.3 LS-SVM formulation to kernel PCA

Classical kernel PCA as shown in Section 2.2 can be derived as the dual problem to a primal optimization problem formulated in a kernel induced feature space [103]. With this formulation the underlying loss function associated to classical kernel PCA was shown to be L_2. In the previously discussed LS-SVM models for classification and regression, the primal is formulated as a convex problem with a unique dual solution characterized by a linear system. However, in the case of kernel PCA, the primal problem is non-convex and the dual solutions characterize stationary points. The main results of the LS-SVM formulation to kernel PCA are summarized as follows.

3.3.1 Unconstrained Formulation

Consider the following unconstrained optimization problem:

$$\min_{w} J(w) = \frac{\gamma}{2N}\sum_{i=1}^{N}\left(w^T(\varphi(x_i) - \hat{\mu}_\varphi)\right)^2 - \frac{1}{2}w^T w \tag{3.8}$$

where γ is a positive regularization constant.

Lemma 3.3. *Each eigenvector $w^\star$ with corresponding eigenvalue ρ of the sample covariance matrix in the feature space $C_\varphi = (1/N)\sum_{j=1}^{N}(\varphi(x_j) - \hat{\mu}_\varphi)(\varphi(x_j) - \hat{\mu}_\varphi)^T$ is a stationary point of $J(w)$ given that $\gamma = 1/\rho$. The objective function $J(w)$ evaluated at any eigenvector $w^\star$ equals zero.*

Proof: The optimization problem in (3.8) can be written as:

$$\min_w J(w) = \frac{\gamma}{2} w^T C_\varphi w - \frac{1}{2} w^T w = \frac{1}{2} w^T (\gamma C_\varphi - I_h) w.$$

Using the first-order optimality condition $\partial J(w)/\partial w = 0$ leads to

$$C_\varphi w^\star = \rho w^\star, \tag{3.9}$$

which is an eigenvalue problem of C_φ with $\rho = 1/\gamma$ and corresponding eigenvector $w^\star$ such that $(w^\star)^T w^\star = 1$. The objective function becomes

$$J(w^\star) = \frac{\gamma}{2}(w^\star)^T \rho w^\star - \frac{1}{2}(w^\star)^T w^\star = \big(\frac{\gamma\rho}{2} - \frac{1}{2}\big)(w^\star)^T w^\star = 0$$

■

Remark 3.4 (Intractability of the primal problem)**.** *Note that solving (3.9) becomes intractable for high dimensional feature spaces and the size of the sample covariance matrix C_φ may become too large.*

3.3.2 Constrained Formulation

Now consider the following constrained optimization problem obtained by introducing additional variables e_i into (3.8):

$$\begin{aligned} \min_{w,e} J_p(w,e) &= \frac{\gamma}{2N}\sum_{i=1}^N e_i^2 - \frac{1}{2} w^T w \\ \text{such that} \quad & e_i = w^T(\varphi(x_i) - \hat{\mu}_\varphi),\ i = 1, ..., N, \end{aligned} \tag{3.10}$$

which corresponds to the LS-SVM formulation to kernel PCA [102]. The dual solution is formalized in the following Lemma.

Lemma 3.4. ***Alzate & Suykens [8]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x,z) = \varphi(x)^T\varphi(z)$ and a regularization constant $\gamma \in \mathbb{R}^+$, the Karush-Kuhn-Tucker (KKT) conditions of the Lagrangian of* (3.10) *are satisfied by the eigenvectors $\alpha^\star$ of the following dual eigenvalue problem:*

$$\Omega_c \alpha = \lambda \alpha \tag{3.11}$$

where Ω_c denotes the centered kernel matrix with ij-th entry: $\Omega_{c,ij} = K(x_i,x_j) - \frac{1}{N}\sum_{r=1}^N K(x_i,x_r) - \frac{1}{N}\sum_{r=1}^N K(x_j,x_r) + \frac{1}{N^2}\sum_{r=1}^N\sum_{s=1}^N K(x_r,x_s)$ and $\lambda = N/\gamma$. The dual objective function $\mathcal{L}(\alpha) = \frac{1}{2}\alpha^T\big(\Omega_c - \lambda I_N\big)\alpha$ evaluated at any eigenvector $\alpha^\star$ equals zero.

Proof: Consider the Lagrangian of the problem (3.10):

$$\mathcal{L}(w, e_i; \alpha) = \frac{\gamma}{2N} \sum_{i=1}^{N} e_i^2 - \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i (e_i - w^T (\varphi(x_i) - \hat{\mu}_\varphi))$$

with KKT optimality conditions:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \displaystyle\sum_{i=1}^{N} \alpha_i (\varphi(x_i) - \hat{\mu}_\varphi) \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow e_i = \dfrac{N}{\gamma} \alpha_i, i = 1, \ldots, N \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow e_i = w^T (\varphi(x_i) - \hat{\mu}_\varphi), i = 1, \ldots, N. \end{cases}$$

Eliminating w and e_i leads to:

$$\frac{N}{\gamma} \alpha_i = \sum_{l=1}^{N} \alpha_l (\varphi(x_l) - \hat{\mu}_\varphi)^T (\varphi(x_i) - \hat{\mu}_\varphi), i = 1, \ldots, N.$$

By making use of the positive definite kernel function:

$$\begin{aligned} (\varphi(x_i) - \hat{\mu}_\varphi)^T (\varphi(x_j) - \hat{\mu}_\varphi) = K(x_i, x_j) &- \frac{1}{N} \sum_{r=1}^{N} K(x_i, x_r) \\ &- \frac{1}{N} \sum_{r=1}^{N} K(x_j, x_r) + \frac{1}{N^2} \sum_{r=1}^{N} \sum_{s=1}^{N} K(x_r, x_s) \\ &= \Omega_{c,ij} \end{aligned}$$

and defining $\lambda = N/\gamma$, $\alpha = [\alpha_1, \ldots, \alpha_N]$ leads to the eigenvalue problem (3.11).

Expressing the Lagrangian in terms of the dual variables α leads to the dual objective function:

$$\mathcal{L}(\alpha) = \frac{1}{2} \alpha^T (\Omega_c - \lambda I_N) \alpha.$$

Given that $\alpha^\star$ is an eigenvector of Ω_c with eigenvalue λ, the dual objective function becomes $\mathcal{L}(\alpha^\star) = 1/2(\lambda - \lambda) = 0$. ■

Note that every eigenvalue/eigenvector pair of Ω_c is a stationary point of (3.10) with objective function value equal to zero. Therefore, the principal components can be seen as a pool of candidate solutions. The component selection problem from this pool can be formulated as:

$$\max_{j\in\{1,\ldots,N\}} \frac{\gamma_j}{2N}\sum_{i=1}^{N}(e_i^{(j)})^2 = \frac{N}{2\gamma_j}\sum_{i=1}^{N}(\alpha_i^{(j)})^2 = \frac{\lambda_j}{2} \tag{3.12}$$

which is equivalent to sorting the components in decreasing order with respect to the eigenvalue λ_j. This differs from classical PCA where the objective function value at the optima is the eigenvalue and reflects the amount of variance captured by the projection onto the corresponding eigenvector.

Remark 3.5 (Equivalence to classical kernel PCA)**.** *Note that the dual solutions (3.11) correspond to kernel PCA as proposed by Schölkopf* et al. *in [97], where [103] showed that a similar form of (3.10) is the underlying optimization problem to which kernel PCA is the dual. Equation (3.11) is regularized by restricting the eigenvectors to the unit ball while regularization is explicitly present on the other hand in the primal problem in (3.10) through the regularization term.*

Remark 3.6 (Centering via the bias term)**.** *Considering a bias term in (3.10) leads to centering the kernel matrix Ω in the feature space such that the mapped data have zero mean, see [103]. The bias term becomes*

$$b = -\frac{1}{N}1_N^T\Omega\alpha. \tag{3.13}$$

Remark 3.7 (Stationary point solution)**.** *Note that changing the minimization operator to maximization in (3.10) leads to the same set of stationary points (3.11).*

Remark 3.8 (Projections)**.** *The projections of a data point x onto an eigenvector (also called the score variables) become:*

$$\begin{aligned} z(x) &= w^T(\varphi(x) - \hat{\mu}_\varphi) \\ &= \sum_{l=1}^{N}\alpha_l\Big(K(x_l,x) - \frac{1}{N}\sum_{r=1}^{N}K(x_r,x) - \\ &\quad \frac{1}{N}\sum_{r=1}^{N}K(x_r,x_l) + \frac{1}{N^2}\sum_{r=1}^{N}\sum_{s=1}^{N}K(x_r,x_s)\Big). \end{aligned} \tag{3.14}$$

3.4 LS-SVM formulation to kernel CCA

An LS-SVM formulation to kernel CCA was introduced in [102]. This formulation emphasizes the use of regularization in the primal problem to avoid overfitting problems common in classical kernel CCA formulations (Remark 2.4). The LS-SVM kernel CCA method has the following primal form:

$$\max_{w,v,e,r} \gamma e^T r - \nu_1 \frac{1}{2} e^T e - \nu_2 \frac{1}{2} r^T r - \frac{1}{2} w^T w - \frac{1}{2} v^T v \quad \text{such that} \quad \begin{cases} e = \Phi_c^{(1)} w \\ r = \Phi_c^{(2)} v. \end{cases} \tag{3.15}$$

where the objective function is related to $||e-r||_2^2$ which is known in CCA. The dual solutions are formulated in the following Lemma.

Lemma 3.5. *[102] Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x,z) = \varphi(x)^T \varphi(z)$ and regularization constants $\gamma, \nu_1, \nu_2 \in \mathbb{R}^+$, the KKT conditions of the Lagrangian of (3.15) are satisfied by the solutions of the following eigenvalue problem:*

$$\left[\begin{array}{c|c} 0 & \Omega_c^{(2)} \\ \hline \Omega_c^{(1)} & 0 \end{array}\right] \left[\begin{array}{c} \xi \\ \hline \tau \end{array}\right] = \lambda \left[\begin{array}{c|c} \nu_1 \Omega_c^{(1)} + I_N & 0 \\ \hline 0 & \nu_2 \Omega_c^{(2)} + I_N \end{array}\right] \left[\begin{array}{c} \xi \\ \hline \tau \end{array}\right]. \tag{3.16}$$

Proof: Consider the Lagrangian of (3.15):

$$\mathcal{L}(w,v,e,r;\xi,\tau) = \gamma e^T r - \nu_1 \frac{1}{2} e^T e - \nu_2 \frac{1}{2} r^T r$$
$$- \frac{1}{2} w^T w - \frac{1}{2} v^T v - \xi^T (e - \Phi_c^{(1)} w) - \tau^T (r - \Phi_c^{(2)} v)$$

where ξ, τ are Lagrange multiplier vectors. The Karush-Kuhn-Tucker (KKT) conditions for optimality are:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 & \to \quad w = {\Phi_c^{(1)}}^T \xi \\ \frac{\partial \mathcal{L}}{\partial v} = 0 & \to \quad v = {\Phi_c^{(2)}}^T \tau \\ \frac{\partial \mathcal{L}}{\partial e} = 0 & \to \quad \gamma r = \nu_1 e + \xi \\ \frac{\partial \mathcal{L}}{\partial r} = 0 & \to \quad \gamma e = \nu_2 r + \tau \\ \frac{\partial \mathcal{L}}{\partial \xi} = 0 & \to \quad e = \Phi_c^{(1)} w \\ \frac{\partial \mathcal{L}}{\partial \beta} = 0 & \to \quad r = \Phi_c^{(2)} v. \end{cases}$$

Eliminating the primal variables w, v, r, e gives:

$$\gamma \Phi_c^{(2)} {\Phi_c^{(2)}}^T \tau = \nu_1 \Phi_c^{(1)} {\Phi_c^{(1)}}^T \xi + \xi$$
$$\gamma \Phi_c^{(1)} {\Phi_c^{(1)}}^T \xi = \nu_2 \Phi_c^{(2)} {\Phi_c^{(2)}}^T \tau + \tau$$

defining $\lambda = 1/\gamma$ and applying the definition of a positive definite kernel function leads to the following set of equations:

$$\begin{cases} \Omega_c^{(2)} \tau = & \lambda(\nu_1 \Omega_c^{(1)} \xi + \xi) \\ \Omega_c^{(1)} \xi = & \lambda(\nu_2 \Omega_c^{(2)} \tau + \tau) \end{cases}$$

which can be written as the generalized eigenvalue problem in (3.16).

■

The projections of the mapped training data onto the i-th eigenvector become

$$z_e^{(i)} = \Phi_c^{(1)} w = \Omega_c^{(1)} \xi^{(i)}, \tag{3.17}$$
$$z_r^{(i)} = \Phi_c^{(2)} v = \Omega_c^{(2)} \tau^{(i)}, \tag{3.18}$$

$i = 1, \ldots, 2N$.

Remark 3.9 (Relation to canonical correlation and regularization)**.** *Note that the primal objective function (3.15) does not have the same expression as the correlation coefficient but takes the numerator with a positive sign and the denominator with a negative sign. Moreover, it can be considered as a generalization of the problem* $\min_{w,v} ||e - r||_2^2$ *which is known in CCA [47]. Also, the regularization is incorporated in the primal optimization problem in a natural way without requiring ad-hoc schemes.*

Remark 3.10 (Selecting the eigenvector solution)**.** *The optimal* $\lambda^\star, \xi^\star, \tau^\star$ *can be selected such that the correlation coefficient in terms of the projections of training data is maximized:*

$$\max_{i \in \{1,\ldots,2N\}} \frac{{z_e^{(i)}}^T z_r^{(i)}}{||z_e^{(i)}||_2 ||z_r^{(i)}||_2}.$$

Remark 3.11 (Normalization)**.** *Note that there is no normalization in the primal problem* (3.15)*, but the normalization is through the dual* (3.16)*, as the solution to the generalized eigenvalue problem.*

3.5 Conclusions

LS-SVM models are powerful kernel-based learning methods for classification, regression, unsupervised learning and beyond. These models aim at solving machine learning problems in a systematic way emphasizing the role of a constrained optimization framework with explicit use of regularization. The primal and dual model representations can be considered as core models allowing extensions in order to achieve desirable properties such as robustness and sparseness. In the case of classification and regression, LS-SVMs lead to solving linear systems instead of quadratic programming problems typical of classical SVMs. For unsupervised learning problems as kernel PCA, the LS-SVM formulation provides insights about the relation between the eigenvalues and regularization together with the L_2 loss function underlying the principal components in the feature space. The LS-SVM formulation to kernel CCA provides an alternative correlation measure where regularization is explicitly present in the primal problem leading to better-conditioned methods and avoiding overfitting issues commonly present in kernel CCA.

Chapter 4

Kernel Component Analysis

This chapter presents one of the contributions of this thesis. The LS-SVM formulation to kernel PCA is extended to a generalized formulation of kernel component analysis (KCA) where the associated loss function is explicit and can easily be changed. In the context of robust statistics, it is known that the L_2 loss function is not robust due to its unbounded influence function [58, 90]. Since kernel PCA has an associated L_2 loss function, the principal components can be arbitrarily skewed by outliers. Another issue with kernel PCA is the lack of sparseness because the projections are expressed in terms of dense kernel expansions associated with every data point. The proposed extension consists of incorporating an epsilon insensitive robust loss function into the LS-SVM kernel PCA formulation in order to impose robustness and sparseness. Two different algorithms are proposed. The first one solves a set of nonlinear equations using e.g. a sequential quadratic programming (SQP) method with classical kernel PCA eigenvalues/vectors as starting points. The second proposed algorithm is a simplified weighting scheme that results in solving a sequence of generalized eigenvalue problems. With this scheme, every data point in the training set has an associated weight which determines its importance in the component calculation. A model selection method on validation data is also proposed. This tuning method looks for kernel parameters and number of components that maximize a

criterion based on the proposed loss function. A heuristic to obtain good choices of the proposed loss function parameters is also proposed. This chapter is organized as follows: Section 4.1 describes two formulations to extend the kernel PCA formulation to other loss functions. Section 4.2 discusses robustness and sparseness in relation to the chosen loss function. Section 4.3 contains two algorithms to find components with a general loss function. Section 4.4 describes a model selection method and in Section 4.5, empirical results are presented.

4.1 Extension to general loss functions

4.1.1 Set of Nonlinear Equations

As already discussed in section 3.3, the loss function underlying kernel PCA is L_2. By extending the formulation now to a general loss function $L : \mathbb{R} \to \mathbb{R}$ and introducing a bias term b, the following primal problem can be formulated:

$$\begin{aligned} \min_{w,e_i,b} \quad & \frac{\gamma}{2}\sum_{i=1}^{N} L(e_i) - \frac{1}{2}w^T w \\ \text{such that} \quad & e_i = w^T\varphi(x_i) + b,\ i = 1, ..., N. \end{aligned} \tag{4.1}$$

The stationary solutions are formulated in the following Lemma.

Lemma 4.1. ***Alzate & Suykens [8]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x, z) = \varphi(x)^T\varphi(z)$, a differentiable loss function $L : \mathbb{R} \to \mathbb{R}$ and positive regularization constant γ, the stationary solutions to the problem (4.1) can be obtained as the solution to the following set of nonlinear equations with unknowns $\alpha_i, i = 1, \ldots, N$ (dual problem) and b:*

$$\begin{cases} \alpha_i - \frac{\gamma}{2}L'\Big(\sum_{l=1}^{N}\alpha_l K(x_l, x_i) + b\Big) = 0,\ i = 1, ..., N \\ \sum_{i=1}^{N} L'\Big(\sum_{l=1}^{N}\alpha_l K(x_l, x_i) + b\Big) = 0. \end{cases} \tag{4.2}$$

Proof: Consider the Lagrangian of the problem (4.1):

$$\mathcal{L}(w, e_i, b; \alpha) = \frac{\gamma}{2}\sum_{i=1}^{N} L(e_i) - \frac{1}{2}w^T w - \sum_{i=1}^{N}\alpha_i(e_i - w^T\varphi(x_i) - b) \tag{4.3}$$

with Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial w} = 0 \rightarrow w = \displaystyle\sum_{i=1}^{N}\alpha_i\varphi(x_i) \\ \dfrac{\partial\mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \dfrac{\gamma}{2}L'(e_i),\ i = 1, ...N \\ \dfrac{\partial\mathcal{L}}{\partial b} = 0 \rightarrow \displaystyle\sum_{i=1}^{N}\alpha_i = 0 \\ \dfrac{\partial\mathcal{L}}{\partial\alpha_i} = 0 \rightarrow e_i = w^T\varphi(x_i) + b,\ i = 1, ...N. \end{cases} \tag{4.4}$$

Eliminating w, e_i leads to:

$$\begin{cases} \dfrac{\gamma}{2}L'\Big(\displaystyle\sum_{l=1}^{N}\alpha_l\varphi(x_l)^T\varphi(x_i) + b\Big) = \alpha_i,\ i = 1, ..., N, \\ \displaystyle\sum_{i=1}^{N} L'\Big(\sum_{l=1}^{N}\alpha_l\varphi(x_l)^T\varphi(x_i) + b\Big) = 0 \end{cases}$$

and applying the definition of a positive definite kernel function gives the set of nonlinear equations (4.2).

The score variables become:

$$z(x) = w^T\varphi(x) + b = \sum_{l=1}^{N}\alpha_l K(x_l, x) + b. \tag{4.5}$$

■

Note that in general (4.1) is a non-convex problem, therefore the KKT optimality conditions are necessary but not sufficient. The solutions to the nonlinear set of equations (4.2) are thus stationary points.

Remark 4.1 (Different centering)**.** *Note that (4.2) leads to a different centering in the feature space, because the bias term is also affected by the loss function. This is important for robustness because the bias term is related to the estimation of the mean in the feature space as shown in the L_2 case.*

4.1.2 Weighting scheme

Another approach to extend kernel PCA to general loss functions is to consider a loss function of the form $L(e_i) = v_i^{(k)} e_i^2$ where $v_i^{(k)}, i = 1, \ldots, N$ are user-defined weights and k is the iteration step. This scheme is similar to the weighted LS-SVM formulation for robust regression discussed in Remark 3.3 and can also be employed in KCA as a simplified technique to obtain robust and sparse solutions. Introducing weighting factors $v_i^{(k)}$ in the formulation leads to the following primal optimization problem:

$$\begin{aligned} \min_{w,e_i} \quad & \gamma \frac{1}{2} \sum_{i=1}^{N} v_i^{(k)} e_i^2 - \frac{1}{2} w^T w \\ \text{such that} \quad & e_i = w^T(\varphi(x_i) - \hat{\mu}_\varphi),\ i = 1, ..., N. \end{aligned} \tag{4.6}$$

A simple method is to make a two-step approach. In general it can be more steps as well. In the first step, an unweighted kernel PCA solution is obtained ($v_i^{(1)} = 1, i = 1, \ldots, N$). Then the weights are calculated in the second step in such a way that $v_i^{(2)} e_i^2 = L(e_i)$. The advantage of this approach is that kernel PCA is extended to general loss functions by solving a sequence of generalized eigenvalue problems which can be solved efficiently, while (4.2) results in a set of nonlinear equations which may lead to local optima and slow convergence.

The first weighted principal component is the solution to (4.6). The second weighted component is the solution to (4.6) with an additional constraint. This constraint imposes orthogonality with respect to the score variables associated to the previously found weighted component. For the third weighted component, two constraints are added to ensure orthogonality with respect to the score variables for the second and first weighted principal components. This iterative process is repeated until all required weighted components are obtained. Hence, the m-th weighted component

is the solution to the following primal problem with a number of $m-1$ orthogonality constraints:

$$\begin{aligned} \min_{w,e_i} \quad & \gamma \frac{1}{2} \sum_{i=1}^{N} v_i^{(k)} e_i^2 - \frac{1}{2} w^T w \\ \text{such that} \quad & \begin{cases} e_i = w^T(\varphi(x_i) - \hat{\mu}_\varphi),\ i = 1, ..., N \\ \sum_{i=1}^{N} e_i e_i^{(1)} = 0 \\ \sum_{i=1}^{N} e_i e_i^{(2)} = 0 \\ \vdots \\ \sum_{i=1}^{N} e_i e_i^{(m-1)} = 0 \end{cases} \end{aligned} \tag{4.7}$$

where $k = 1, \ldots, N_{\text{iter}}$ is the iteration step, $e_i^{(l)}, l = 1, \ldots, m-1$ are the score variables corresponding to the l-th previously found weighted principal component, which are assumed to be given. The dual form of (4.7) is formalized in the following Lemma.

Lemma 4.2. ***Alzate & Suykens [8]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x,z) = \varphi(x)^T \varphi(z)$, score variables $e_i^{(1)}, e_i^{(2)}, \ldots, e_i^{(m-1)} \in \mathbb{R},\ i = 1, \ldots, N$ and positive weighting factors $v_i^{(k)} \in \mathbb{R}^+,\ i = 1, \ldots, N$, the solution to the problem (4.7) is given by the following generalized eigenvalue problem:*

$$\left[\begin{array}{c|c} \Omega_c^V & 0_{N\times(m-1)} \\ \hline 0_{N\times(m-1)}^T & 0_{(m-1)\times(m-1)} \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline \beta \end{array}\right] = \lambda \left[\begin{array}{c|c} I_N & E \\ \hline E^T & E_V \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline \beta \end{array}\right] \tag{4.8}$$

where $V^{(k)} = \text{diag}(v_1^{(k)}, \ldots, v_N^{(k)})$, $\Omega_c^V = V^{(k)}\Omega_c$, $0_{p\times q}$ is the $p \times q$ zero matrix, $\alpha = [\alpha_1; \ldots; \alpha_N]$, $\beta = [\beta_1; \beta_2; \ldots; \beta_{m-1}]$, $\lambda = 1/\gamma$, I_N is the $N \times N$ identity matrix and E, E_V are defined as follows:

$$E = \begin{bmatrix} \frac{e_1^{(1)}}{v_1^{(k)}} & \frac{e_1^{(2)}}{v_1^{(k)}} & \cdots & \frac{e_1^{(m-1)}}{v_1^{(k)}} \\ \frac{e_2^{(1)}}{v_2^{(k)}} & \frac{e_2^{(2)}}{v_2^{(k)}} & \cdots & \frac{e_2^{(m-1)}}{v_2^{(k)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{e_N^{(1)}}{v_N^{(k)}} & \frac{e_N^{(2)}}{v_N^{(k)}} & \cdots & \frac{e_N^{(m-1)}}{v_N^{(k)}} \end{bmatrix}$$

$$E_V = \begin{bmatrix} \sum_i \frac{(e_i^{(1)})^2}{v_i^{(k)}} & \sum_i \frac{e_i^{(1)} e_i^{(2)}}{v_i^{(k)}} & \cdots & \sum_i \frac{e_i^{(1)} e_i^{(m-1)}}{v_i^{(k)}} \\ \sum_i \frac{e_i^{(1)} e_i^{(2)}}{v_i^{(k)}} & \sum_i \frac{(e_i^{(2)})^2}{v_i^{(k)}} & \cdots & \sum_i \frac{e_i^{(2)} e_i^{(m-1)}}{v_i^{(k)}} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_i \frac{e_i^{(1)} e_i^{(m-1)}}{v_i^{(k)}} & \sum_i \frac{e_i^{(2)} e_i^{(m-1)}}{v_i^{(k)}} & \cdots & \sum_i \frac{(e_i^{(m-1)})^2}{v_i^{(k)}} \end{bmatrix}.$$

Proof: Consider the Lagrangian of the problem (4.7):

$$\mathcal{L}(w,e_i;\alpha_i,\beta_j) = \gamma \frac{1}{2} \sum_{i=1}^{N} v_i^{(k)} e_i^2 - \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i (e_i - w^T(\varphi(x_i) - \hat{\mu}_\varphi)) - \beta_1 \sum_{i=1}^{N} e_i e_i^{(1)} - \beta_2 \sum_{i=1}^{N} e_i e_i^{(2)} - \ldots - \beta_{i-1} \sum_{i=1}^{N} e_i e_i^{(m-1)}$$

with KKT optimality conditions:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i (\varphi(x_i) - \hat{\mu}_\varphi) \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow e_i = \frac{1}{\gamma v_i^{(k)}} (\alpha_i + \beta_1 e_i^{(1)} + \ldots + \beta_{i-1} e_i^{(m-1)}), i = 1, \ldots, N \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow e_i = w^T (\varphi(x_i) - \hat{\mu}_\varphi), \ i = 1, \ldots, N \\ \frac{\partial \mathcal{L}}{\partial \beta_1} = 0 \rightarrow \sum_{i=1}^{N} e_i e_i^{(1)} = 0 \\ \frac{\partial \mathcal{L}}{\partial \beta_2} = 0 \rightarrow \sum_{i=1}^{N} e_i e_i^{(2)} = 0 \\ \quad \vdots \\ \frac{\partial \mathcal{L}}{\partial \beta_{i-1}} = 0 \rightarrow \sum_{i=1}^{N} e_i e_i^{(m-1)} = 0. \end{cases}$$

Using $\partial \mathcal{L}/\partial w = 0, \partial \mathcal{L}/\partial e_i = 0$, $\partial \mathcal{L}/\partial \alpha_i = 0$ and applying the kernel trick leads to:

$$\frac{1}{\gamma}(\alpha_i + \beta_1 e_i^{(1)} + \beta_2 e_i^{(2)} + \ldots + \beta_{m-1} e_i^{(m-1)}) = v_i^{(k)} \sum_{j=1}^{N} \alpha_j \Omega_c(x_j, x_i), \quad (4.9)$$

$i = 1, \ldots, N$. Substituting $\partial \mathcal{L}/\partial \alpha_i = 0$ into the orthogonality constraints leads to the following set of equations:

$$\begin{cases} \sum_{j=1}^{N} \frac{\alpha_j e_j^{(1)}}{v_j^{(k)}} + \beta_1 \sum_{j=1}^{N} \frac{(e_j^{(1)})^2}{v_j^{(k)}} + \ldots + \beta_{m-1} \sum_{j=1}^{N} \frac{e_j^{(1)} e_j^{(m-1)}}{v_j^{(k)}} = 0 \\ \sum_{j=1}^{N} \frac{\alpha_j e_j^{(2)}}{v_j^{(k)}} + \beta_1 \sum_{j=1}^{N} \frac{e_j^{(1)} e_j^{(2)}}{v_j^{(k)}} + \ldots + \beta_{m-1} \sum_{j=1}^{N} \frac{e_j^{(1)} e_j^{(m-1)}}{v_j^{(k)}} = 0 \\ \vdots \\ \sum_{j=1}^{N} \frac{\alpha_j e_j^{(m-1)}}{v_j^{(k)}} + \beta_1 \sum_{j=1}^{N} \frac{e_j^{(1)} e_j^{(m-1)}}{v_j^{(k)}} + \ldots + \beta_{m-1} \sum_{j=1}^{N} \frac{(e_j^{(m-1)})^2}{v_j^{(k)}} = 0. \end{cases} \quad (4.10)$$

The sets of equations (4.9) and (4.10) give the generalized eigenvalue problem (4.8).

■

Remark 4.2 (Real eigenvalues)**.** *The generalized eigenvalue problem* (4.8) *has real eigenvalues since* Ω_c^V *is the product of two positive definite matrices* $V^{(k)}$*,* Ω_c *and the matrix in the right hand side of the generalized eigenequation is symmetric [56].*

4.2 A robust and sparse loss function

In this Section, a loss function is proposed in order to incorporate robustness and sparseness into kernel component analysis.

4.2.1 Robustness in PCA and kernel PCA

It is known that classical PCA is heavily influenced by outlying observations. A single outlier can by itself determine an entire component [65]. A small deviation from the assumptions produces a gross error in the principal

component estimation. Although several approaches to robustify PCA have been developed [116], [33], [55], [31], [60], that is not the case for kernel PCA. Robustness theory from the point of view of influence function is summarized as follows.

Let T_N be a target statistic based on the training data $\{x_i\}_{i=1}^N$ assuming that T_N can be expressed as a functional, say $T(\bar{F}_N)$, where $\bar{F}_N$ indicates the empirical distribution function of the data. The influence function of $T(\bar{F}_N)$ against an outlier ξ is defined as:

$$IF(\xi; \bar{F}_N, T) = \lim_{t \to 0} \frac{T[(1-t)\bar{F}_N + t\delta_\xi] - T(\bar{F}_N)}{t}. \tag{4.11}$$

where δ_ξ is the pointmass 1 distribution at ξ. If the influence function is bounded, then the influence of the functional $T(\bar{F}_N)$ by any ξ should have a reduced effect.

The i-th principal component $\alpha^{(i)}$ can be expressed as a functional $\alpha^{(i)}(\bar{F}_N)$ and according to [30], [65], [54] its influence function becomes:

$$IF(\xi; \bar{F}_N, \alpha^{(i)}) = -\beta_1(\xi) \sum_{j \neq i}^{d} \beta_j(\xi)(\lambda_j - \lambda_i)^{-1} \alpha^{(j)} \tag{4.12}$$

where $\beta_j(\xi) = (\alpha^{(j)})^T(\xi - \hat{\mu})$, λ_j is the j-th eigenvalue, $\hat{\mu}$ is the sample mean of the data and $\alpha^{(j)}$ is the j-th principal component. Note that (4.12) depends of all the components and is unbounded which confirms the non-robust nature of PCA.

For *M-estimators* [58], the influence function of the functional $T(\bar{F}_N)$ is defined as:

$$IF(\xi; \bar{F}_N, T) = \frac{L'(\xi)}{\int L'(\xi) d\bar{F}_N}. \tag{4.13}$$

In other words, the influence function of an *M-estimator* is proportional to the first derivative of the loss function (also called the score function). Several loss functions with bounded score functions have been proposed. Well-known examples are the Huber, Hampel, Welsch and Tukey loss functions. Kernel PCA has an underlying L_2 loss function with an unbounded score function (Figure 4.1). Hence, kernel PCA is not robust against outlying data points. Note also that kernel PCA is linear PCA in a kernel-induced feature space, therefore outliers in the feature space have a large influence. Christmann *et al.* showed in [25] that the kernel and

the score function should be bounded in convex risk minimization methods to achieve robustness. Bounded kernels (e.g. the RBF kernel) will lead to models where outliers have only a local influence rather than global. An explicit form of the influence function for kernel PCA has been described in [35] showing also the key role of bounded kernels for robustness.

4.2.2 Sparseness

As can be seen in (4.5), the score variables are expressed in terms of dense kernel expansions in which every training point contributes. Different methods have been proposed to overcome the lack of sparseness in kernel PCA. Smola *et al.* in [99] proposed a method for obtaining sparse components employing an L_1 penalty on the expansion coefficients α_i. Schölkopf *et al.* in [95] introduced a compact representation for the expansions leading to a reduced set method.

Because the loss function is explicit in our formulation, it is possible to introduce an epsilon insensitive zone into a robust loss function and achieve sparseness and robustness at the same time due to the conditions for optimality in (4.4). The proposed loss function (Figure 4.1) is a combination of the Huber loss and the Vapnik epsilon insensitive:

$$L(e) = \begin{cases} 0 & \text{if } |e| \leq \epsilon \\ e^2 - 2|e| + \epsilon^2 & \text{if } \epsilon < |e| < h \\ 2(h-\epsilon)|e| - h^2 + \epsilon^2 & \text{if } |e| \geq h. \end{cases} \tag{4.14}$$

with tunable parameters ϵ, h and $h > \epsilon$. The score function becomes:

$$\psi(e) = L'(e) = \begin{cases} 0 & \text{if } |e| \leq \epsilon \\ 2\text{sign}(e)(|e| - \epsilon) & \text{if } \epsilon < |e| < h \\ 2\text{sign}(e)(h - \epsilon) & \text{if } |e| \geq h. \end{cases} \tag{4.15}$$

Taking an epsilon zone in the loss function leads to sparseness because of the conditions for optimality (4.4). Note that setting ϵ to zero leads to the Huber loss function and $\epsilon > 0, h \to \epsilon$ approaches to the Vapnik epsilon insensitive loss function.

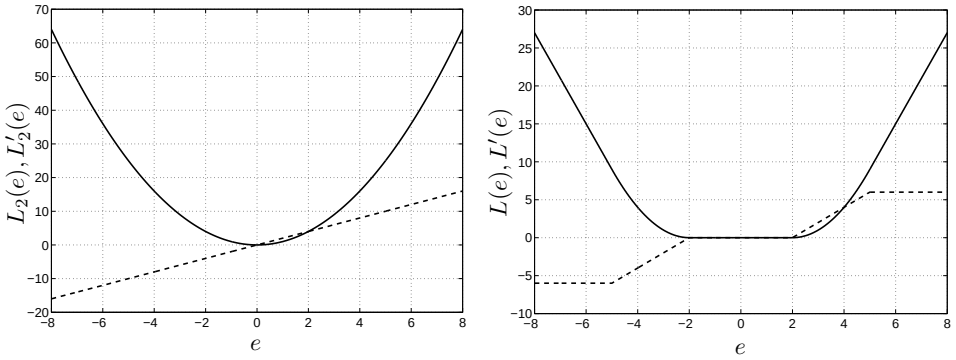


Figure 4.1: **Left:** L_2 loss function and its score function. **Right:** Proposed loss function with $\epsilon = 2, h = 5$ and its score function. Note that $L'(e)$ is bounded which is important for robustness.

4.3 Algorithms

We propose two algorithms to obtain the components with the Huber epsilon-insensitive loss function. The first algorithm solves (4.2) using a sequential quadratic programming (SQP) approach [85] and smooth approximation of $L'(e)$. The second algorithm corresponds to an implementation of the weighted scheme (4.8).

4.3.1 SQP Algorithm

The Huber epsilon-insensitive score function $L'(e)$ is not differentiable at $|e| = \epsilon$ and $|e| = h$ therefore to use a gradient-based approach such as SQP, a smooth approximation $L'_s(e)$ to $L'(e)$ is taken:

$$L'_s(e) = \begin{cases} \frac{2}{s}\log\left(1+\exp((s(e+h)))\right) - 2(h-\epsilon) & \text{if } e \leq \frac{\epsilon-h}{2} - \epsilon \\ \frac{2}{s}\log\left(\frac{1+\exp((s(e-\epsilon)))}{1+\exp((-s(e+\epsilon)))}\right) & \text{if } \frac{\epsilon-h}{2} - \epsilon < |e| < \frac{h-\epsilon}{2} + \epsilon \\ -\frac{2}{s}\log\left(1+\exp((-s(e-h)))\right) + 2(h-\epsilon) & \text{if } e \geq \frac{h-\epsilon}{2} + \epsilon \end{cases} \tag{4.16}$$

where s is a constant that controls the smoothness of the approximation. Figure 4.2 shows $L'_s(e)$ and $L'(e)$ for $\epsilon = 2, h = 5, s = 20$.

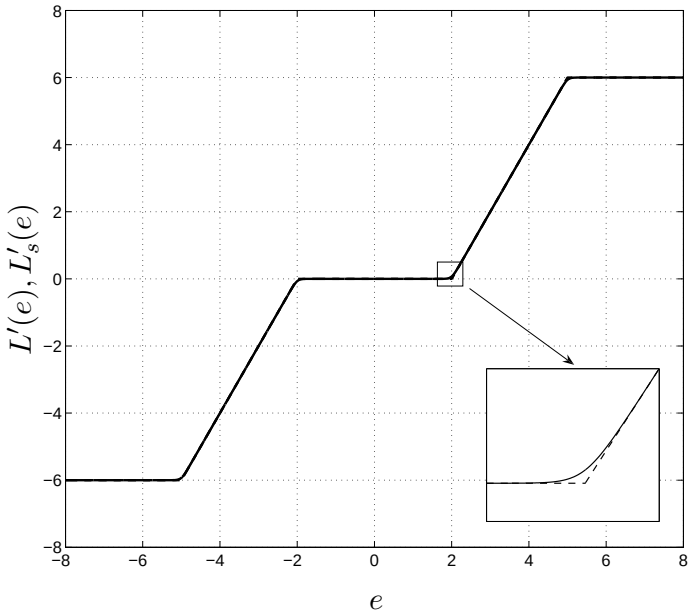


Figure 4.2: Smooth approximation of the Huber epsilon-insensitive score function. The dashed line corresponds to $L'(e)$ and the solid line to $L'_s(e)$.

Replacing $L'(\cdot)$ by $L'_s(\cdot)$ in (4.2) leads to:

$$f(\lambda, \alpha, b) = \begin{bmatrix} \lambda\alpha - \frac{1}{2}\tilde{L}'_s(z(\alpha, b)) \\ 1_N^T \tilde{L}'_s(z(\alpha, b)) \end{bmatrix} = 0 \tag{4.17}$$

where $f : \mathbb{R}^+ \times \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^{N+1}$, $z(\alpha, b) = [z_1(\alpha, b); z_2(\alpha, b); \ldots; z_N(\alpha, b)]$,$\lambda = 1/\gamma$, $\alpha = [\alpha_1; \alpha_2; \ldots; \alpha_N]$, $z_i(\alpha, b) = \Omega^{(i)}\alpha + b1_N$, $\Omega^{(i)}$ denotes the i-th row of the kernel matrix Ω and

$$\tilde{L}'_s(z(\alpha, b)) = \big[L'_s(z_1(\alpha, b)); \ldots; L'_s(z_N(\alpha, b))\big].$$

The set of nonlinear equations (4.17) is usually solved via minimization of a sum of squares of the residuals. We impose two additional constraints $\alpha^T\alpha \geq 1$ and $\lambda \geq c$ where c is a small positive constant (e.g. $c = 10^{-6}$. The first constraint prevents the trivial solution $\alpha = 0$ and non-interesting solutions with small norm causing the score variables $z_i(\alpha, b)$ to lie in the range $-h < |z_i(\alpha, b)| < h$, $i = 1, \ldots, N$, therefore avoiding the bounded zone $|z_i(\alpha, b)| \geq h$ of the score function. The proposed nonlinear optimization

problem is the following:

$$\begin{aligned} \min_{\lambda,\alpha,b} \quad & g(\lambda,\alpha,b) = f(\lambda,\alpha,b)^T f(\lambda,\alpha,b) \\ \text{such that} \quad & \begin{cases} \alpha^T\alpha \geq 1 \\ \lambda \geq c \end{cases} \end{aligned} \tag{4.18}$$

where

$$\begin{aligned} g(\lambda,\alpha,b) &= \lambda^2\alpha^T\alpha - \lambda\alpha^T\tilde{L}'_s(z(\alpha,b)) \\ &+ \frac{1}{4}\big(\tilde{L}'_s(z(\alpha,b))\big)^T\tilde{L}'_s(z(\alpha,b)) + \big(1_N^T\tilde{L}'_s(z(\alpha,b))\big)^2. \end{aligned}$$

All $N+1$ equations have to be satisfied at the local solution $(\lambda^\star,\alpha^\star,b^\star)$ rather than just minimizing the sum of the squares of the equations. This means that the value of $g(\lambda^\star,\alpha^\star,b^\star)$ should be zero in order to $(\lambda^\star,\alpha^\star,b^\star)$ to be a solution to the problem (4.17). In the same fashion as in the weighting scheme, additional constraints can be added to impose orthogonality with previously found components. Note that sparseness is achieved *component-wise* due to $\alpha_i = \frac{\gamma}{2}L'(e_i)$ which is one of the optimality conditions (4.4). Therefore, each component vector may have a different number of zero α_i values.

An SQP method approximates in every iteration the constrained optimization problem (4.18) as a quadratic programming problem of the form:

$$\begin{aligned} \min_{d} \quad & \frac{1}{2}d^T H^{(k)} d + \nabla g(\theta^{(k)})^T d \\ \text{such that} \quad & \begin{cases} \nabla c_1(\theta^{(k)})^T d + c_1(\theta^{(k)}) \geq 0 \\ \nabla c_2(\theta^{(k)})^T d + c_2(\theta^{(k)}) \geq 0 \end{cases} \end{aligned} \tag{4.19}$$

where k denotes the iteration step, H is the Hessian matrix (or an approximation to it) of the Lagrangian of (4.18), $\theta = [\lambda;\alpha;b]$ is the vector of parameters, $c_1(\theta) = \alpha^T\alpha - 1$ is the first constraint, $c_2(\theta) = \lambda - c$ is the second constraint, $g(\theta) = g(\lambda;\alpha;b)$ and the gradient $\nabla g(\theta) \in \mathbb{R}^{N+2}$ is defined as:

$$\nabla g(\theta) = \nabla g(\lambda,\alpha,b) = \begin{bmatrix} \nabla_\lambda g(\lambda,\alpha,b) \\ \nabla_\alpha g(\lambda,\alpha,b) \\ \nabla_b g(\lambda,\alpha,b) \end{bmatrix} \tag{4.20}$$

where

$$\nabla_\lambda g(\lambda,\alpha,b) = 2\lambda\alpha^T\alpha - \alpha^T\tilde{L}'_s(z(\alpha,b)),$$

$$\begin{aligned}
\nabla_\alpha g(\lambda,\alpha,b) &= 2\lambda^2\alpha \\
&\quad - \lambda\Big(\Omega\mathrm{diag}\big(\tilde{L}_s''(z(\alpha,b))\big)\alpha + \tilde{L}_s'(z(\alpha,b))\Big) \\
&\quad + \frac{1}{2}\Omega\mathrm{diag}\big(\tilde{L}_s''(z(\alpha,b))\big)\tilde{L}_s'(z(\alpha,b)) + 2r_1\Omega\tilde{L}_s''(z(\alpha,b)), \\
\nabla_b g(\lambda,\alpha,b) &= -\lambda\alpha^T\tilde{L}_s''(z(\alpha,b)) + \\
&\quad \frac{1}{2}\big(\tilde{L}_s''(z(\alpha,b))\big)^T\tilde{L}_s'(z(\alpha,b)) + 2r_1r_2,
\end{aligned}$$

$r_1 = 1_N^T\tilde{L}_s'(z(\alpha,b)), r_2 = 1_N^T\tilde{L}_s''(z(\alpha,b))$. The solution vector $d^{(k)}$ to the QP subproblem (4.19) can then be used to form a new iterate:

$$\theta^{(k+1)} = \theta^{(k)} + \mu^{(k)}d^{(k)} \tag{4.21}$$

where $\mu^{(k)}$ is the step length parameter of the line search and can be obtained using a merit function.

A critical issue in nonlinear optimization is the selection of good starting points. In the proposed SQP approach, kernel PCA eigenvectors and eigenvalues are used as initial values for the nonlinear solver. Therefore, we are considering only the first local solution obtained through the proposed initialization. Given a set of N $d-$dimensional data points $\mathcal{D} = \{x_1,\ldots,x_N\}$, the algorithm gives a set of N_c component vectors $\mathcal{A} = \{\alpha^{(1)\star},\ldots,\alpha^{(N_c)\star}\}$ as follows:

Algorithm 1 KCA: SQP Approach

1: Set the kernel parameters and the loss function parameters ϵ, h.
2: Perform an eigendecomposition of the non-centered kernel matrix to obtain initial eigenvalues $\{\lambda_1,\ldots,\lambda_N\}$ and initial eigenvectors $\{\alpha^{(1)},\ldots,\alpha^{(N)}\}$.
3: **for** $l = 1$ to N_c **do**
4: Obtain the initial values for the bias term $b_l = (1/N)1_N^T\Omega\alpha^{(l)}$
5: Solve the SQP problem (4.19), (4.21) with initial values λ_l, $\alpha^{(l)}$ and b_l to obtain the solutions $\lambda_l^\star$, $\alpha^{(l)\star}$ and $b_l^\star$
6: **end for**
7: Sort the solutions according to the value of $\lambda_l^\star$ in decreasing order.
8: Normalize the components $\alpha^{(l)\star} \leftarrow \alpha^{(l)\star}/||\alpha^{(l)\star}||_2, l = 1,\ldots,N_c$.

4.3.2 Weighted Algorithm

The second algorithm is an implementation of the iterative weighting scheme (4.7). Given $\mathcal{D}$, and the number of iterations N_{iter}, the algorithm 2 gives a

set of N_c components $\{\alpha^{(1)}, \ldots, \alpha^{(N_c)}\}$, where $N_r \leq N$.

Algorithm 2 KCA: Weighted Kernel PCA

1: Set the kernel parameters, the loss function parameters ϵ, h and the number of principal components N_c to be extracted.
2: Compute the unweighted score variables $e_i^{(l)}, i = 1, \ldots, N, l = 1, \ldots, N_c$ associated to the first N_c principal components using classical kernel PCA.
3: Compute the weights $v_i^{(l)}$ such that $v_i^{(l)}(e_i^{(l)})^2 = L(e_i^{(l)})$.
4: Remove from the training dataset the data points whose corresponding weight $v_i^{(l)}$ equals to zero in every component. The new dataset has N_r data points.
5: Compute the unweighted score variables $e_i^{(l)}, l = 1, \ldots, N_c$ of the reduced dataset using classical kernel PCA.
6: **for** $k = 1$ to N_{iter} **do**
7: Recompute the weights $v_i^{(1)}$ such that $v_i^{(1)}(e_i^{(1)})^2 = L(e_i^{(1)})$ with $\epsilon = 0$.
8: Find the first weighted principal component $\alpha^{(1)}$ which is the solution associated to the largest eigenvalue in $V\Omega_c\alpha = \lambda\alpha$, with $V = \text{diag}(v_1, \ldots, v_{N_r})$.
9: Recompute $e_i^{(1)}$ using $\alpha^{(1)}$.
10: **for** $l = 2$ to N_c **do**
11: Recompute the weights $v_i^{(l)}$ such that $v_i^{(l)}(e_i^{(j)})^2 = L(e_i^{(l)})$ with $\epsilon = 0$
12: Find the $l-$th weighted principal component $\alpha^{(l)}$ which is the solution associated to the largest eigenvalue in (4.8).
13: Recompute $e_i^{(l)}$ using $\alpha^{(l)}$.
14: **end for**
15: **end for**

In general, empirical results suggest that few iterations are sufficient to obtain satisfactory results. Sparseness is achieved in step 4 by removing data points from the training set for which the corresponding weight is equal to zero across the components. The computational complexity of this algorithm is $\mathcal{O}(N_{\text{iter}}N_c(\xi N + N_c))$ compared with $\mathcal{O}(\xi N)$ for classical kernel PCA using the Lanczos method for the largest N_c components. The factor ξ is the maximum number of matrix-vector computations required and depends of the eigengap $\lambda_{N_c+1} - \lambda_{N_c}$ [48], [113].

4.4 Model Selection

Model selection is a central issue in unsupervised learning techniques such as kernel PCA where the parameters that define the model are the number of

principal components N_c to be extracted and the kernel parameters (e.g. σ^2 for the RBF kernel). These parameters are typically chosen in a trial-and-error fashion. In the proposed approaches, model selection in a learning framework can be done naturally on the basis of the explicit constrained optimization formulation (4.1). We propose the following model selection criterion:

$$\max_{N_c,\sigma^2} \sum_{l=1}^{N_c} \sum_{j=1}^{N_{\text{val}}} L(\tilde{e}_j^{(l)}) \tag{4.22}$$

where $\tilde{e}_j^{(l)} = w^T \varphi(x_j^{\text{val}}) + b_l = \sum_{i=1}^{N} \alpha_i^{(l)} K(x_i, x_j^{\text{val}}) + b_l$, x_l^{val} denotes the j-th validation data point, $\alpha^{(l)}$ is the l-th component and b_l is the bias term for the l-th component, $l = 1, \ldots, N_c$. The values of ϵ (sparseness) and h (robustness) are problem-dependent and can be obtained heuristically from the empirical distribution of the score variables such that a given amount of the score values lie within the epsilon zone of the loss function. In the same way, h can be set such that a given percentage of the score values lie within the range $-h < e < h$. Figure 4.3 shows the proposed heuristic.

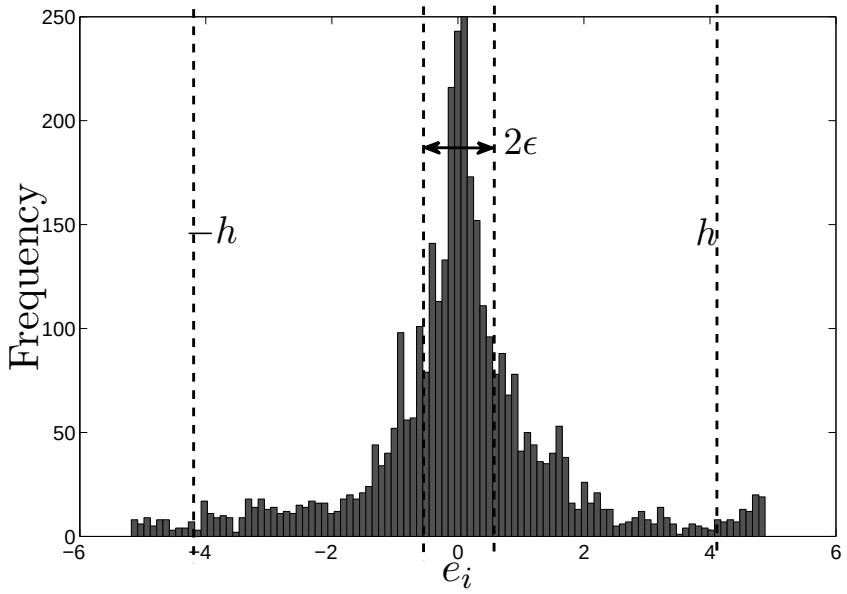


Figure 4.3: Empirical distribution of the score variables associated to the kernel principal components. The parameters of the proposed loss function (ϵ, h) which represent sparseness and robustness respectively can be determined heuristically from this plot.

4.5 Experimental Results

In this section, some experimental results are presented to illustrate the effectiveness of the proposed approaches. We used the MATLAB function *fmincon* to solve (4.19). This function is a SQP solver with a quasi-Newton approximation to the Hessian of the Lagrangian using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. The step length is calculated with line search and the merit function described in [87]. All experiments reported were carried out using the RBF kernel. It is important to mention that the kernel should be bounded for these robust methods to work. The machine used is a Pentium 4, 2.8 GHz, 1GB RAM. The values of ϵ and h were obtained using the heuristic proposed in Figure 4.3, setting ϵ such that 10% of the score values lie within the epsilon insensitive range and setting h such that 90% of the score values lie within the range $-h < e < h$. In general, ϵ and h can also change in each iteration. However in this case, we used the proposed heuristic to find ϵ, h from the unweighted score variables and considered these values fixed during the iterations.

Since the experiments correspond to denoising applications of kernel PCA, we briefly describe the pre-image algorithms used in combination with the proposed method as follows:

- Pre-image algorithm 1: Fixed-point iteration scheme for RBF kernels [81].
- Pre-image algorithm 2: Robust extension of the pre-image algorithm 1. Introduced in [105].
- Pre-image algorithm 3: Non-iterative approach based on MDS concepts. This algorithm can be applied to a larger class of kernels [69].

Simulations were performed with a curve-shaped dataset in two dimensions and with the UCI *multiple features* handwritten digit dataset. The curve dataset consists of 360 data points for training, 420 for validation and 320 for testing. The loss function parameters $\epsilon = 0.34, h = 3.7$ were set according to Figure 4.3. Model selection surfaces for the curve dataset can be seen in Figure 4.4 showing that in the weighted kernel PCA method, the criterion saturates with respect to the number of components N_c. On the other hand, in kernel PCA, the variance of the projected variables is always increasing with respect to N_c. This means that fewer components have to be computed in the weighted kernel PCA algorithm. Obtained parameters are $\sigma^2 =$

$0.45, N_c = 40$ for kernel PCA and $\sigma^2 = 2.0, N_c = 8$ for the SQP algorithm. Sparseness of the proposed method can be seen in Figure 4.5.

Note that sparseness varies from component to component. However, most of the outliers are not support vectors therefore they can be removed from the dataset. Table 4.1 shows the percentage of outliers present in the support vector set. The reduced set selection method [95] is more sparse than the proposed methods but its support vector set contains almost all outliers. The components found by the SQP algorithm are more sparse than the components obtained by the simplified weighting scheme but at a more expensive computational cost. Figure 4.6 shows the denoising results using the SQP approach and classical kernel PCA with three different pre-image algorithms.

The handwritten digit training set is composed of 195 digit images of 15×16 pixels. The validation set consists of 40 digits. We added outliers and Gaussian noise to the dataset and set $\epsilon = 0.1, h = 4$ according to Figure 4.3. Figure 4.7 shows the slices of the model selection surfaces with similar results compared to Figure 4.4. Obtained parameters are $\sigma^2 = 350, N_c = 190$ for kernel PCA and $\sigma^2 = 300, N_c = 120$ for the SQP algorithm. The denoising results can be seen in Figures 4.8 and 4.9 showing improved results of the SQP algorithm compared to kernel PCA in terms of robustness together with visually appealing digits. Table 4.2 shows the mean squared error (MSE) and standard deviation for kernel PCA and the proposed methods with respect to the three pre-image algorithms. The MSE was calculated between the denoised digits and the test set without outliers and Gaussian noise. Figure 4.10 shows sparseness results in terms of accumulated number of support vectors with respect to the number of components.

	% of outliers	Computation times
Reduced Set [95]	98%	5.4 secs
Weighted Algorithm	3%	8.6 secs
SQP Algorithm	0%	24.7 secs

Table 4.1: Fraction of the total number of outliers present on the support vector set. Note that, the reduced set method is more sparse than the proposed methods but the majority of the outliers are in the support vector set.

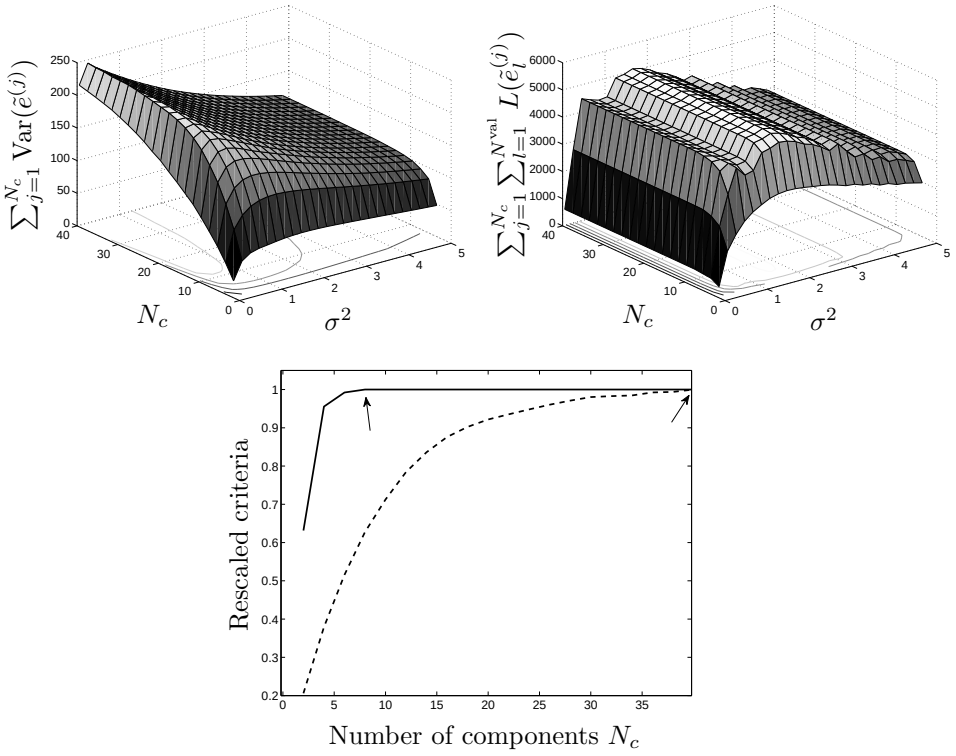


Figure 4.4: Model selection surfaces on validation data. **Top left:** Kernel PCA. **Top right:** KCA with SQP algorithm. **Bottom:** Slice plot of the model selection surfaces for the best σ^2. *Dashed line:* Average of the maximal variance on validation data (kernel PCA). *Solid line:* Average of the maximal $L(e_i^{\mathrm{val}})$ on validation data. The arrows show the selected number of components. The value of the proposed criterion saturates with less components compared to kernel PCA. Obtained parameters are $\sigma^2 = 0.45, N_c = 40$ for kernel PCA and $\sigma^2 = 2, N_c = 8$ for the SQP algorithm.

4.6 Conclusions

A new generalization to the kernel PCA formulation for general loss functions is derived. This formulation is based on the LS-SVM context and can be used to impose desirable properties such as sparseness and robustness in the analysis. These two properties are missing in the classical kernel PCA because it has an associated L_2 loss function, and the solutions are

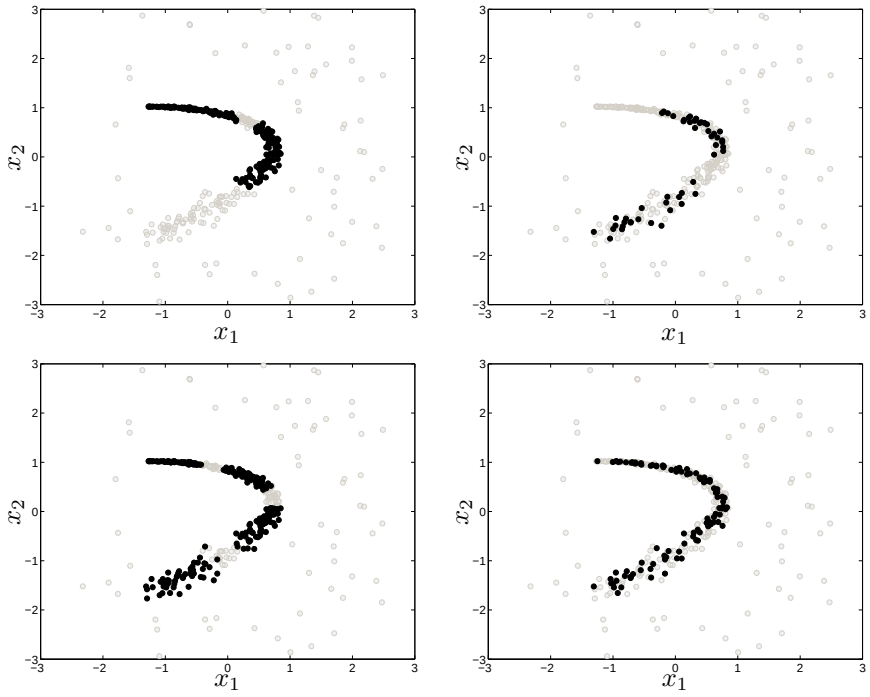


Figure 4.5: Sparseness per component SQP algorithm. The figure shows the support vectors (black points) for the first 4 components. Most of the outliers are not support vectors in any component, therefore they can be removed from the dataset.

	Kernel PCA	KCA:SQP alg.	KCA:Weighted alg.
Preimage alg. 1	0.60 ± 0.40	0.10 ± 0.15	0.28 ± 0.30
Preimage alg. 2	0.70 ± 0.30	0.21 ± 0.20	0.45 ± 0.32
Preimage alg. 3	0.54 ± 0.40	0.30 ± 0.23	0.32 ± 0.28

Table 4.2: MSE and standard deviation on test data of the denoised digits compared to the clean test data. The test set was randomized 10 times.

expressed in terms of dense kernel expansions. To overcome these issues, we proposed a loss function with an epsilon insensitive zone for sparseness and bounded score function for robustness. We have shown two different algorithms to obtain components using this loss function. The first algorithm solves a set of nonlinear equations using an SQP method. This nonlinear solver was initialized using classical kernel PCA eigenvectors/values which were shown to be good initial starting points leading to convergence in

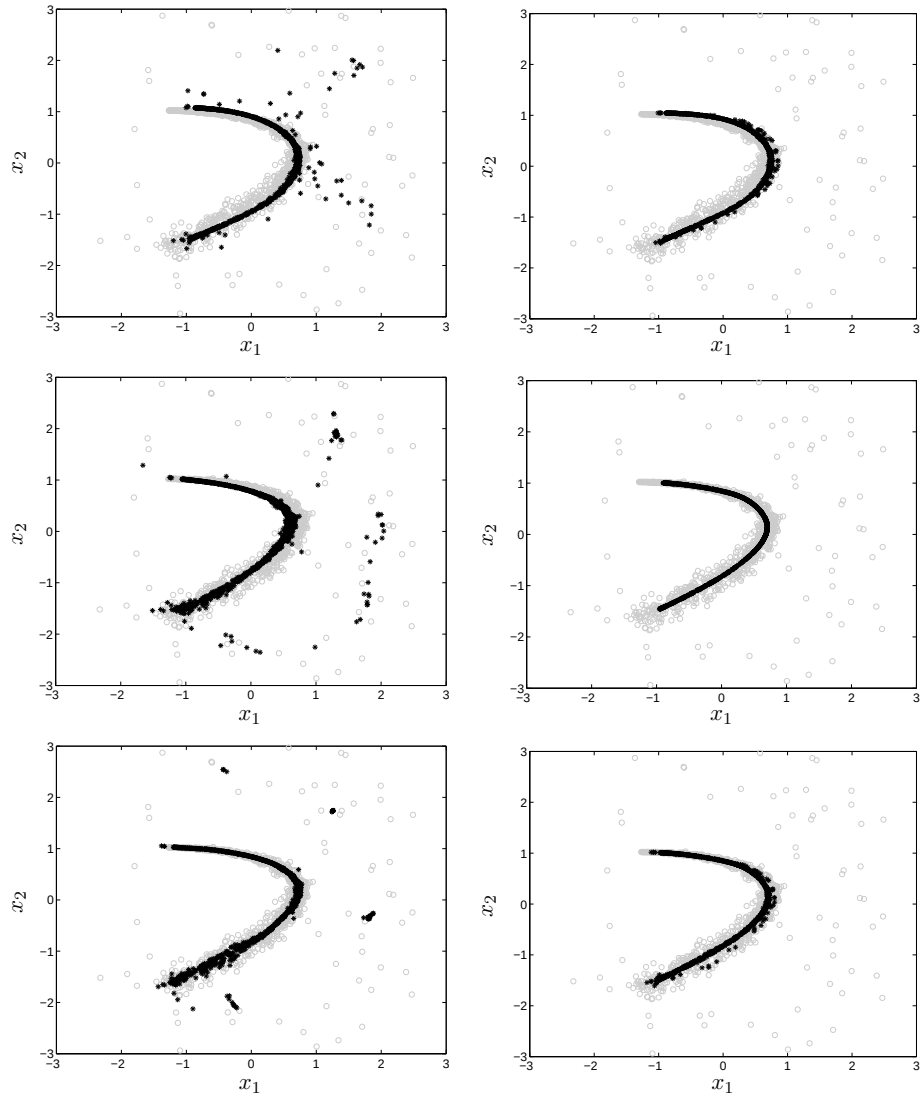


Figure 4.6: Denoising results using three pre-image algorithms. Original data is depicted in gray and denoised data point in black. **Left column:** Kernel PCA with $\sigma^2 = 0.45, N_c = 40$. **Right column:** KCA with SQP algorithm with $\sigma^2 = 2.0, N_c = 8, \epsilon = 0.34, h = 3.7$. **Top:** Pre-image algorithm 1. **Center:** Pre-image algorithm 2. **Bottom:** Pre-image algorithm 3. In all cases, the SQP approach performs better in terms of robustness to outliers.

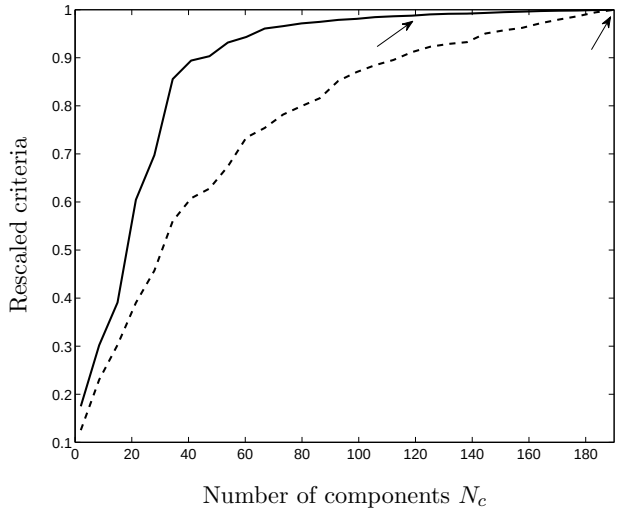


Figure 4.7: Slice plot of the model selection surfaces for the best RBF kernel parameter σ^2. **Dashed line:** Average of the maximal variance on validation data (kernel PCA). **Solid line:** Average of the maximal $L(e_i^{\text{val}})$ on validation data (KCA with SQP algorithm). The arrows show the selected number of components. The value of the proposed criterion saturates with less components compared to kernel PCA. Obtained parameters are $\sigma^2 = 0.45, N_c = 40$ for kernel PCA and $\sigma^2 = 2, N_c = 8$ for the SQP algorithm.

most cases. However, we only consider the first local solution obtained through this initialization while other local optima may be interesting as well. The second algorithm is a simplified iterative weighting scheme that leads to a sequence of generalized eigenvalue problems which can be solved efficiently. We proposed a model selection criterion based on the primal/dual constrained optimization framework to obtain the kernel parameters and the number of components. A heuristic to find good choices of the loss function parameters was also discussed. From simulations results, the proposed methods showed good performance in terms of robustness together with a sparse representation. The components obtained by the SQP algorithm were shown to be sparser than the components found by the weighted approach. However, the latter algorithm was shown to be computationally faster.

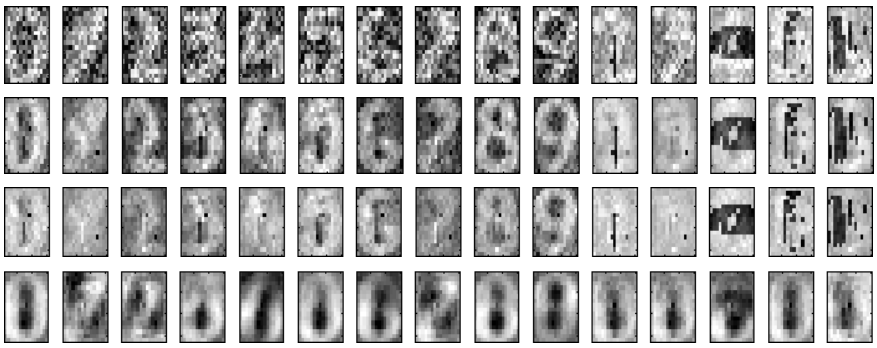


Figure 4.8: Denoising results using kernel PCA with $\sigma^2 = 350, N_c = 190$. **First row:** Test set corrupted with Gaussian noise and outliers. **Second row:** Denoising with pre-image algorithm 1. **Third row:** Denoising with pre-image algorithm 2. **Fourth row:** Denoising with pre-image algorithm 3.

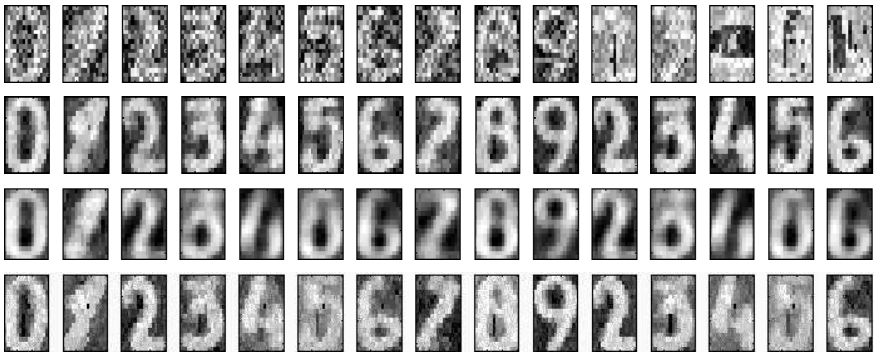


Figure 4.9: Denoising results using the SQP algorithm with $\sigma^2 = 300, N_c = 120$. **First row:** Test set corrupted with Gaussian noise and outliers. **Second row:** Denoising with pre-image algorithm 1. **Third row:** Denoising with pre-image algorithm 2. Fourth row: Denoising with pre-image algorithm 3. The SQP algorithm show improved results in terms of robustness compared to Figure 4.8.

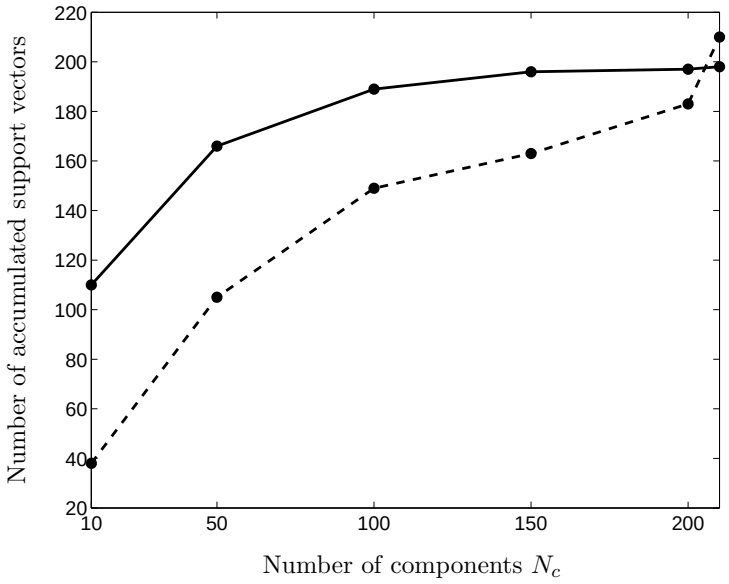


Figure 4.10: Number of accumulated support vectors with respect to the number of components. **Solid line:** SQP algorithm. **Dashed line:** Reduced set selection method [95]. The reduced set method contains less support vectors but outliers are not removed.

Chapter 5

ICA via Kernel Regularized Correlation

This chapter presents another contribution of this work. The LS-SVM formulation for kernel CCA discussed in Section 3.4 is extended to the multivariate case when more than two sets of data are considered. The new formulation leads to a regularized correlation measure in high dimensional feature spaces induced by kernels. The regularization is incorporated naturally in the primal problem leading to a dual generalized eigenvalue problem. The smallest generalized eigenvalue is a measure of correlation in the feature space and corresponds to a measure of pairwise independence in the input space when a universal kernel is used. Due to the primal-dual nature of the proposed approach, the measure of independence can also be extended to out-of-sample points which is important for model selection ensuring statistical reliability of the proposed measure. Computational issues due to the large size of the matrices involved in the eigendecomposition are tackled via the incomplete Cholesky factorization. This factorization can be used to obtain low-rank approximations of the kernel matrix achieving a considerable reduction in the case of a fast decaying eigenspectrum. This chapter is organized as follows: In Section 5.1, we present the proposed kernel regularized correlation (KRC) which can be used as a contrast function for ICA. Section 5.2 discusses an approximation of the

contrast function using the incomplete Cholesky decomposition together with an algorithm. In Section 5.3, we propose a model selection criterion and Section 5.4 contains simulations with toy data, images and speech signals.

5.1 The Kernel Regularized Correlation (KRC)

5.1.1 Multivariate Kernel CCA

In this section, the LS-SVM formulation to kernel CCA discussed in section 3.4 is extended to m sets of variables. Given training data $\{x_i^{(l)}\}, i = 1, \ldots, N, l = 1, \ldots, m$, the multivariate kernel CCA problem can be formulated in the primal as:

$$\max_{w^{(l)}, e^{(l)}} \quad \frac{\gamma}{2} \sum_{l=1}^{m} \sum_{k \neq l}^{m} e^{(l)^T} e^{(k)} - \frac{1}{2} \sum_{l=1}^{m} \nu_l e^{(l)^T} e^{(l)} - \frac{1}{2} \sum_{l=1}^{m} w^{(l)^T} w^{(l)} \quad (5.1)$$
$$\text{such that } e^{(l)} = \Phi_c^{(l)} w^{(l)}, l = 1, \ldots, m$$

where the objective function can be interpreted as a generalization of the expression $||e^{(1)} - e^{(2)}||_2^2$ towards m sets of data $\sum_{l=1}^{m} \sum_{k \neq l}^{m} ||e^{(l)} - e^{(k)}||_2^2$, $e^{(l)}$ is the compact form of the l-th projected variables

$$e_i^{(l)} = w^{(l)^T} (\varphi^{(l)}(x_i^{(l)}) - \hat{\mu}_{\varphi^{(l)}}), i = 1, \ldots, N, l = 1, \ldots, m,$$

the l-th centered feature matrix $\Phi_c^{(l)} \in \mathbb{R}^{N \times n_h}$ is

$$\Phi_c^{(l)} = \begin{bmatrix} \varphi^{(l)}(x_1^{(l)})^T - \hat{\mu}_{\varphi^{(l)}}^T \\ \varphi^{(l)}(x_2^{(l)})^T - \hat{\mu}_{\varphi^{(l)}}^T \\ \vdots \\ \varphi^{(l)}(x_N^{(l)})^T - \hat{\mu}_{\varphi^{(l)}}^T \end{bmatrix},$$

γ, ν_l are regularization parameters and

$$\hat{\mu}_{\varphi^{(l)}} = (1/N) \sum_{i=1}^{N} \varphi^{(l)}(x_i^{(l)}), l = 1, \ldots, m.$$

The dual solutions to (5.1) are formulated as follows.

Lemma 5.1. ***Alzate & Suykens [9]*** *Given m positive definite kernel functions $K^{(l)} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ with $K^{(l)}(x,z) = \varphi^{(l)}(x)^T\varphi^{(l)}(z)$ and regularization constants $\gamma, \nu_l \in \mathbb{R}^+, l = 1,\ldots,m$, the stationary points of the Lagrangian of (5.1) are given by eigenvectors of the following generalized eigenvalue problem:*

$$\mathcal{K}\alpha = \lambda\mathcal{R}\alpha \tag{5.2}$$

with $\mathcal{K}, \mathcal{R}$ and α defined as follows:

$$\mathcal{K}=\left[\begin{array}{c|c|c|c} 0 & \Omega_c^{(2)} & \ldots & \Omega_c^{(m)} \\ \hline \Omega_c^{(1)} & 0 & \ldots & \Omega_c^{(m)} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \Omega_c^{(1)} & \Omega_c^{(2)} & \ldots & 0 \end{array}\right], \mathcal{R}=\left[\begin{array}{c|c|c|c} R^{(1)} & 0 & \ldots & 0 \\ \hline 0 & R^{(2)} & \ldots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \ldots & R^{(m)} \end{array}\right], \alpha=\left[\begin{array}{c} \alpha^{(1)} \\ \hline \alpha^{(2)} \\ \hline \vdots \\ \hline \alpha^{(m)} \end{array}\right],$$

$R^{(l)} = (I_N + \nu_l\Omega^{(l)}), l = 1,\ldots,m$ *and* $\lambda = 1/\gamma$.

Proof: The Lagrangian for this constrained optimization problem is:

$$\begin{aligned}\mathcal{L}(w^{(l)}, e^{(l)}; \alpha^{(l)}) = \frac{\gamma}{2}\sum_{l=1}^{m}\sum_{k\neq l}^{m} e^{(l)^T}e^{(k)} - \frac{1}{2}\sum_{l=1}^{m}\nu_l e^{(l)^T}e^{(l)} \\ -\frac{1}{2}\sum_{l=1}^{m} w^{(l)^T}w^{(l)} - \sum_{l=1}^{m}\alpha^{(l)^T}(e^{(l)} - \Phi_c^{(l)}w^{(l)})\end{aligned} \tag{5.3}$$

with conditions for optimality given by:

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial w^{(l)}} = 0 \rightarrow w^{(l)} = \Phi_c^{(l)^T}\alpha^{(l)} \\ \dfrac{\partial\mathcal{L}}{\partial e^{(l)}} = 0 \rightarrow \gamma\displaystyle\sum_{k\neq l} e^{(k)} = \nu_l e^{(l)} + \alpha^{(l)} \\ \dfrac{\partial\mathcal{L}}{\partial \alpha^{(l)}} = 0 \rightarrow e^{(l)} = \Phi_c^{(l)}w^{(l)}, \end{cases}$$

for $l = 1,\ldots,m$. Eliminating the primal variables $w^{(l)}, e^{(l)}$ gives:

$$\gamma\sum_{k\neq l}^{m}\Phi_c^{(k)^T}\Phi_c^{(k)}\alpha^{(k)} = (\nu_l\Phi_c^{(l)^T}\Phi_c^{(l)} + I_N)\alpha^{(l)}, l = 1,\ldots,m$$

defining $\lambda = 1/\gamma$, applying the definition of a positive definite kernel function and rewritting the above equations in matrix form leads to (5.2). ■

The score variables for training data can be expressed as:

$$[z^{(l)}]_i = \Phi_c^{(l)} w^{(l)} = \Omega^{(l)} [\alpha^{(l)}]_i,$$

where $[\alpha^{(l)}]_i$ is the $\alpha^{(l)}$ corresponding to the i-th eigenvector, $i = 1, \ldots, Nm, l = 1, \ldots, m$. Given new test data $\{\tilde{x}_j^{(l)}\}, j = 1, \ldots, N_{\text{test}}, l = 1, \ldots, m$ the projections $z_{\text{test}}^{(l)}$ can be calculated as:

$$[z_{\text{test}}^{(l)}]_i = \Omega_{\text{test}}^{(l)} [\alpha^{(l)}]_i, \tag{5.4}$$

where $\Omega_{\text{test}}^{(l)} \in \mathbb{R}^{N_{\text{test}} \times N}$ is the kernel matrix evaluated using the test points with ji-entry $\Omega_{\text{test},ji}^{(l)} = K^{(l)}(\tilde{x}_j^{(l)}, x_i^{(l)}), j = 1, \ldots, N_{\text{test}}, i = 1, \ldots, Nm$.

Remark 5.1 (Regularized correlation measure)**.** *Note that the regularized matrix $\mathcal{R}$ differs from the schemes proposed in [14, 49, 117]. These schemes start from an unregularized kernel CCA problem and perform regularization afterwards in an ad-hoc manner (see Remark 2.4). As already discussed, the proposed regularization arises naturally from the primal optimization problem providing insights and leading to better numerically conditioned eigenvalue problems.*

Remark 5.2 (Selecting the eigenvector solution)**.** *The optimal $\lambda^\star, \alpha^\star$ can be selected such that the sum of the correlation coefficients computed on training data is maximized :*

$$\max_{i \in \{1,\ldots,Nm\}} \sum_{l=1}^{m} \sum_{k \neq l}^{m} \frac{\left[z^{(l)}\right]_i^T \left[z^{(k)}\right]_i}{\left\| \left[z^{(l)}\right]_i \right\|_2 \left\| \left[z^{(k)}\right]_i \right\|_2}.$$

5.1.2 The KRC Contrast Function

The largest generalized eigenvalue of the problem (5.2) provides a measure of correlation among the sets of data. Even though this measure may not correspond to the correlation coefficient, it can be used as a contrast function for ICA. Consider the following modification to (5.2):

$$(\mathcal{K} + \mathcal{R})\alpha = \zeta \mathcal{R} \alpha, \tag{5.5}$$

where $\zeta = 1 + \lambda$. The largest eigenvalue of (5.2) is related to the smallest eigenvalue $\zeta_{\min}$ of (5.5) and $\zeta_{\min}$ is bounded between zero and

one. Therefore, the proposed kernel regularized correlation (KRC) contrast function for ICA is defined as follows for training data:

$$\hat{\rho}(\Omega^{(1)}, \ldots, \Omega^{(m)}) = 1 - \zeta_{\min}. \tag{5.6}$$

The KRC for test data is the sum of the correlation coefficients of the projections:

$$\hat{\rho}(\Omega_{\text{test}}^{(1)}, \ldots, \Omega_{\text{test}}^{(m)}) = \sum_{l=1}^{m} \sum_{k \neq l}^{m} \frac{z_{\text{test}}^{(l)^T} z_{\text{test}}^{(k)}}{||z_{\text{test}}^{(l)}||_2 ||z_{\text{test}}^{(k)}||_2}. \tag{5.7}$$

5.2 Approximations and Algorithm

5.2.1 The Incomplete Cholesky Decomposition

In practice, the size of the matrices in (5.5) makes the cost of computing the smallest eigenvalue prohibitive. However, low-rank approximations of the kernel matrices can be found by means of the incomplete Cholesky decomposition. An $N \times N$ kernel matrix Ω can be factorized as $\Omega = GG^T$ where $G \in \mathbb{R}^{N \times N}$ is lower triangular. If the eigenvalues of Ω decay sufficiently rapidly then it is possible to make the approximation $\Omega \approx GG^T$ where $G \in \mathbb{R}^{N \times M}$ such that $||\Omega - GG^T||_2 \leq \eta$ and $M \ll N$. This method is known as the incomplete Cholesky decomposition with symmetric pivoting and was used in [14, 49] to reduce the computational burden of the corresponding eigenvalue problem.

Making use of the incomplete Cholesky decomposition, the centered kernel matrices in (5.5) can be approximated as $\Omega_c^{(l)} \approx G_c^{(l)} G_c^{(l)^T}$. The eigenvalue decomposition of $\Omega_c^{(l)}$ is $U^{(l)} \Lambda^{(l)} U^{(l)^T}$ where $U^{(l)}$ is an $N \times M_l$ orthogonal matrix and $\Lambda^{(l)}$ is an $M_l \times M_l$ diagonal matrix. Therefore we can approximate (5.5) with

$$\tilde{\mathcal{K}}\alpha = \zeta \tilde{\mathcal{R}}\alpha \tag{5.8}$$

where

$$\tilde{\mathcal{K}}=\left[\begin{array}{c|c|c|c} U^{(1)}\Lambda_\nu^{(1)}U^{(1)^T} & U^{(2)}\Lambda^{(2)}U^{(2)^T} & \dots & U^{(m)}\Lambda^{(m)}U^{(m)^T} \\ \hline U^{(1)}\Lambda^{(1)}U^{(1)^T} & U^{(1)}\Lambda_\nu^{(2)}U^{(2)^T} & \dots & U^{(m)}\Lambda^{(m)}U^{(m)^T} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline U^{(1)}\Lambda^{(1)}U^{(1)^T} & U^{(2)}\Lambda^{(2)}U^{(2)^T} & \dots & U^{(m)}\Lambda_\nu^{(m)}U^{(m)^T} \end{array}\right],$$

$$\tilde{\mathcal{R}}=\left[\begin{array}{c|c|c|c} U^{(1)}\Lambda_\nu^{(1)}U^{(1)^T} & 0 & \dots & 0 \\ \hline 0 & U^{(1)}\Lambda_\nu^{(2)}U^{(2)^T} & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & U^{(m)}\Lambda_\nu^{(m)}U^{(m)^T} \end{array}\right],$$

$\Lambda_\nu^{(l)}$ is a diagonal matrix with nn-entry

$$\Lambda_{\nu,nn}^{(l)} = 1 + \nu_l\Lambda_{nn}^{(l)}, n = 1, \dots, M_l.$$

Substituting $\tilde{\alpha}^{(l)} = U^{(l)^T}\alpha^{(l)}, i = l, \dots, m$ and premultiplying both sides of (5.8) by blkdiag$(U^{(1)^T}, \dots, U^{(m)^T})$ leads to the following reduced problem:

$$\tilde{\mathcal{K}}_{\mathcal{R}}\tilde{\alpha} = \zeta\tilde{\alpha} \tag{5.9}$$

where

$$\tilde{\mathcal{K}}_{\mathcal{R}} = \left[\begin{array}{c|c|c|c} I_{M_1} & T_U^{(1)}U_\Lambda^{(2)} & \dots & T_U^{(1)}U_\Lambda^{(m)} \\ \hline T_U^{(2)}U_\Lambda^{(1)} & I_{M_2} & \dots & T_U^{(1)}U_\Lambda^{(m)} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline T_U^{(m)}U_\Lambda^{(1)} & T_U^{(m)}U_\Lambda^{(2)} & \dots & I_{M_m} \end{array}\right], \tilde{\alpha} = \left[\begin{array}{c} \tilde{\alpha}^{(1)} \\ \hline \tilde{\alpha}^{(2)} \\ \hline \vdots \\ \hline \tilde{\alpha}^{(m)} \end{array}\right],$$

$T_U^{(l)} = T^{(l)}U^{(l)^T}$, $T^{(l)}$ is a diagonal matrix $T_{nn}^{(l)} = 1/\Lambda_{nn}^{(l)}$, $U_\Lambda^{(l)} = U^{(l)}\Lambda^{(l)}, l = 1, \dots, m, n = 1, \dots, M_l$. Note that the size of $\tilde{\mathcal{K}}_{\mathcal{R}}$ is $(\sum_{l=1}^m M_l) \times (\sum_{l=1}^m M_l)$ which is much smaller than the original $mN \times mN$ problem in (5.5).

5.2.2 The Stiefel Manifold

The Stiefel manifold is described by the set of all $m \times q$ orthogonal matrices W (i.e. $W^TW = I_q$). The geometry of this special manifold was studied in [42] together with gradient descent algorithms for optimization on the manifold. The tangent space corresponds to the space of all matrices A such that W^TA is a skew-symmetric. A geodesic is defined as the shortest path between two points in the manifold and is determined as $G_W(t) =$

$W\exp(tW^TA)$ where W is the starting point, A is the direction and t is the displacement. The direction A can be determined as $A = D_W - WD_W^TW$ where D_W is the derivative of the contrast function with respect to W. Gradient descent can then be used along the geodesic in the direction A to obtain orthogonal matrices W such that the contrast function is minimized. The kernel-based ICA approaches described in [14,49] apply gradient descent methods with golden search on the Stiefel manifold in order to obtain demixing matrices leading to independent components. We also use this approach to minimize the KRC. The computation of the KRC and the corresponding minimization on the Stiefel manifold are shown in Algorithms 3 and 4.

Algorithm 3 Kernel Regularized Correlation (KRC)

Input: Mixture set of m sources $\{x_i^{(l)}\}_{i=1}^N, l = 1,\ldots,m$, RBF kernel parameter σ^2, regularization parameter ν, Cholesky error threshold η.
Output: $\hat{\rho}(\Omega_c^{(1)},\ldots,\Omega_c^{(m)})$.

1: Compute the Cholesky factor $G_c^{(l)} \in \mathbb{R}^{N\times M_l}$ such that $||\Omega_c^{(l)} - G_c^{(l)}G_c^{(l)^T}||_2 \leq \eta, l = 1,\ldots,m$.
2: Compute the SVD $G_c^{(l)} = U^{(l)}\Lambda^{(l)}U^{(l)^T}, l = 1,\ldots,m$ and build the matrix $\tilde{\mathcal{K}}_{\mathcal{R}}$ (5.9).
3: Compute the smallest eigenvalue $\zeta_{\min}$ of the problem $\tilde{\mathcal{K}}_{\mathcal{R}}\tilde{\alpha} = \zeta\tilde{\alpha}$.
4: **Return** $1 - \zeta_{\min}$.

The computational complexity of the KRC is $\mathcal{O}((mM)^2N)$, where M is the maximal rank considered by the incomplete Cholesky decompositions. However, as noted by [14], the bottleneck of the computation is the incomplete Cholesky decomposition. Therefore, the empirical computational complexity of the KRC is $\mathcal{O}(mM^2N)$.

5.3 Model Selection

Typically, the free parameters in kernel-based approaches to ICA are set heuristically to some values that show good performance on a number of experiments. However, overfitting may occur due to the fact that the estimation of the correlation coefficient is done solely on training data especially when few data are available. One of the main advantages of our proposed approach is a clear constrained optimization problem for kernel

Algorithm 4 Gradient descent on the Stiefel manifold

Input: Whitened mixture set of m sources $\{x_i^{(l)}\}_{i=1}^N, l = 1, \ldots, m$, RBF kernel parameter σ^2, regularization parameter ν, Cholesky error threshold η, initial demixing matrix W_0.
Output: Demixing matrix W.

1: $W \leftarrow W_0$
2: **repeat**
3: Compute D_W the derivative of the KRC with respect to W using first-order differences.
4: Compute the direction $A = D_W - W D_W^T W$.
5: Find a displacement t along the geodesic using golden search in the direction A such that the KRC decreases.
6: Update $W \leftarrow W \exp(tW^T A)$
7: **until** change of W in two consecutive iterations is less than some given threshold.

CCA and the corresponding measure of independence. This allows the computation of projected variables for out-of-sample (test) data points that can be used to select values for the free parameters. The parameters values are chosen such that the normalized correlation coefficient is minimal for validation data. This ensures statistical reliability of the estimated canonical correlation.

The RBF kernel parameters are assumed to be identical $(\sigma^{(l)})^2 = \sigma^2, l = 1, \ldots, m$ which means considering one feature map $\varphi^{(l)} = \varphi$. The regularization parameters are also considered to be identical $\nu_l = \nu$ and fixed to some value. Given validation data $\{\check{x}_h^{(l)}\}, h = 1, \ldots, N_{\text{val}}, l = 1, \ldots, m$, the following model selection criterion is proposed:

$$\hat{\rho}(\Omega_{\text{val}}^{(1)}, \ldots, \Omega_{\text{val}}^{(m)}) = \sum_{l=1}^{m} \sum_{k \neq l}^{m} \frac{z_{\text{val}}^{(l)^T} z_{\text{val}}^{(k)}}{||z_{\text{val}}^{(l)}||_2 ||z_{\text{val}}^{(k)}||_2 ||\Omega_{\text{val}}||_F} \tag{5.10}$$

where $z_{\text{val}}^{(l)} = \Omega_{\text{val}}^{(l)} \alpha^{(l)\star}$, $\Omega_{\text{val}}^{(l)}$ is the kernel matrix evaluated using the validation points with hi-entry $\Omega_{\text{val},hi}^{(l)} = K^{(l)}(\check{x}_h^{(l)}, x_i^{(l)}), h = 1, \ldots, N_{\text{val}}, i = 1, \ldots, N$ and $\Omega_{\text{val}} = \text{blkdiag}(\Omega_{\text{val}}^{(1)}, \ldots, \Omega_{\text{val}}^{(m)})$. Note that, for RBF kernels the correlation measure decreases as σ^2 increases [14]. Therefore, $||\Omega_{\text{val}}||_F$ in (5.10) fixes the scaling introduced by σ^2.

5.4 Experimental Results

In this section, we present empirical results. In all experiments, we assume that the number of sources m is given and the incomplete Cholesky parameter was fixed to $\eta = 1 \times 10^{-4}$. We use the Amari divergence [12] to assess the performance of the proposed approach when the optimal demixing matrix W is known. This error measure compares two demixing matrices and is equal to zero if and only if the matrices represent the same components. Moreover, the Amari divergence is invariant to permutation and scaling which makes it suitable for ICA. We compared four different existing algorithms for ICA: FastICA [62], the KCC contrast [14], the KGV [14], Jade [23] and the proposed KRC. The kernel-based algorithms were initialized with the solution of the KGV using a Hermite polynomial of degree $d = 3$ and width $\sigma = 1.5$. This solution does not characterize independence but it was shown to be a good starting point [14]. The computation time of the eigenvalue decomposition involved in the calculation of the KRC for training data was 0.98 seconds for $N = 28,000$, $m = 3$, $\eta = 10^{-4}$ on a Pentium 4, 2.8 GHz, 1GB RAM using Lanczos algorithm in Matlab. In the case of test and validation data, the average computation time increased one order of magnitude because the projected variables should be calculated in order to compute the correlation coefficient.

5.4.1 Toy Examples - Simple signals

In this experiment we used three simple signals: a sine function, a sawtooth function and a sinc function. The sample size was $N = 400$. The sources were mixed using a product of known Jacobi rotations $\theta_3 = -\pi/3, \theta_2 = -\pi/2, \theta_1 = -\pi/5$. In this way, the optimal demixing matrix is a 3-D rotation matrix with angles $-\theta_1, -\theta_2, -\theta_3$. The first angle was fixed to $\pi/5$ and an estimate of the demixing matrix was made using a grid in the range $[0, \pi]$ for the remaining two angles. Figure 5.1 shows the contour plots of the Amari divergence, the KCC and KGV criteria described in [14] and the proposed KRC criterion. The KCC and KGV parameters were set to the values recommended by the authors. The KRC parameters were set to $\sigma = 5, \nu = 2$. The black dot shows the location of the optimal set of angles which corresponds to a local minimum for all criteria. The effect of the KRC parameters σ^2, ν on the optimization surface can be seen in Figure 5.2. Note that the optimal set of angles corresponds to a local minimum in the KRC

for a wide range of parameter values.

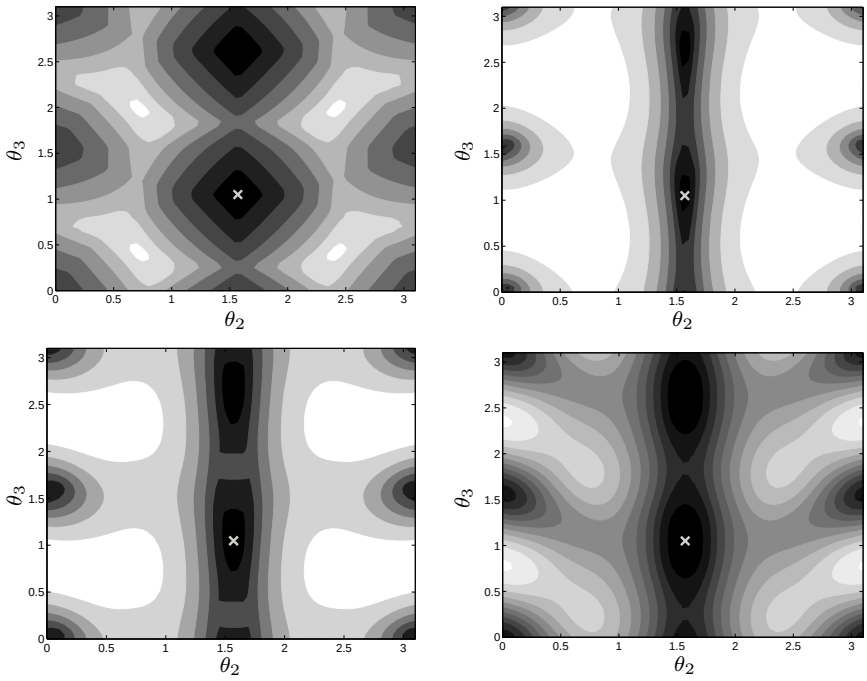


Figure 5.1: Contour plots of optimization surfaces. Three sources were mixed using a product of known Jacobi rotations. The optimal set of angles is shown as a white cross. **Top left:** Amari divergence. **Top right:** Proposed KRC using $\sigma^2 = 5, \nu = 2$. **Bottom left:** KCC criterion [14] using $\sigma^2 = 1, \kappa = 2 \times 10^{-2}$. **Bottom right:** KGV criterion [14] using the same set of parameters as KCCA. Note that all contrast functions are smooth and the white cross corresponds to a local minimum.

5.4.2 Toy Examples - Source distributions

This experiment consisted of demixing six different source distributions. The sample size was $N = 900$. The distributions include a multimodal asymmetric mixture of two Gaussians, a Student's t with 3 degrees of freedom, a uniform distribution, a mixture of a Gaussian and uniform, a beta distribution and a gamma distribution. We performed 20 runs with different random mixing matrices and compute the mean Amari error for the KCC, the KGV, the FastICA algorithm, Jade and the proposed KRC.

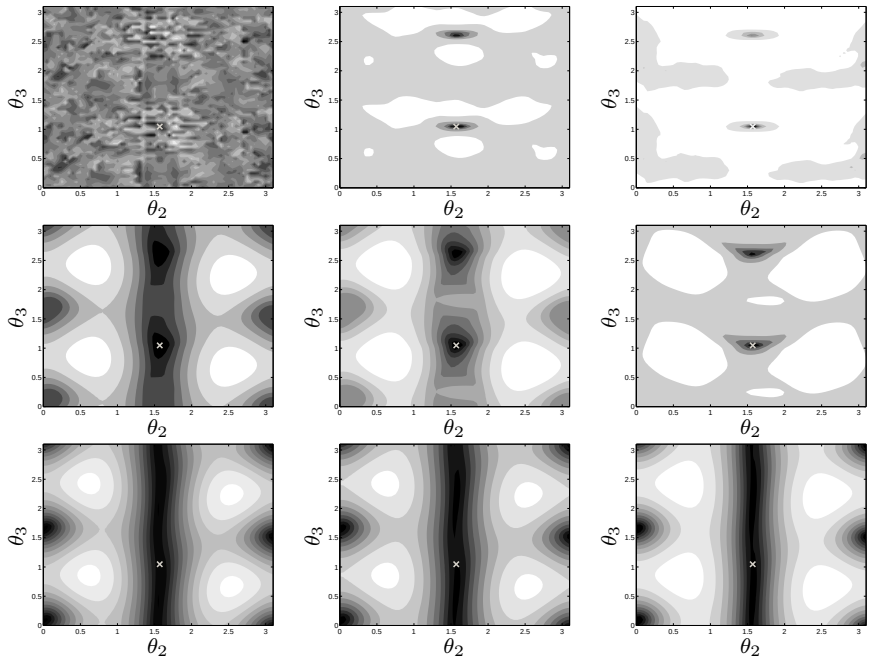


Figure 5.2: Effect of the KRC parameters on the optimization surface. **Top row:** $\sigma^2 = 0.1$. **Middle row:** $\sigma^2 = 1$. **Bottom row:** $\sigma^2 = 10$. **Left column:** $\nu = 0.1$. **Middle column:** $\nu = 1$. **Right column:** $\nu = 10$. The white cross indicates the optimal set of angles. Note that for most parameter values the white cross corresponds to a local minimum in the KRC.

As in the previous toy example, the parameters of the KCC and KGV were set to the default values. The FastICA parameters were set to the default values and the kurtosis-based contrast function was used. Table 5.1 and Figure 5.3 show the results. In all cases the KRC obtained the smallest Amari error. Mean computation times for the KCCA, the KGV and the KRC are shown in Figure 5.4. The KRC computation time is comparable to the KCC. The FastICA algorithm took in average 0.7 seconds but with reduced performance in terms of Amari error as shown in Table 5.1.

5.4.3 Speech signals

We used three fragments of 6.25 seconds of speech sampled at 8 KHz with 8 bit quantization. We performed 10 runs with a random window

# sources	KCC	KGV	KRC	FastICA	Jade
2	4.02	2.77	**2.75**	12.92	4.95
3	7.43	2.89	**2.36**	9.37	6.98
4	6.12	3.55	**2.19**	9.88	8.30
5	6.39	5.75	**4.48**	11.03	8.88
6	8.42	5.57	**4.06**	9.23	7.49

Table 5.1: Mean Amari error (multiplied by 100) of the source distributions experiment after 20 runs. The KCC and KGV parameters were set to the values recommended by the authors ($\sigma^2 = 1, \kappa = 2 \times 10^{-2}$) in [14]. The KRC parameters were set to $\sigma^2 = 0.5, \nu = 1$. The FastICA algorithm was set to the kurtosis-based contrast and the default parameters. In all cases the KRC obtained the smallest Amari error.

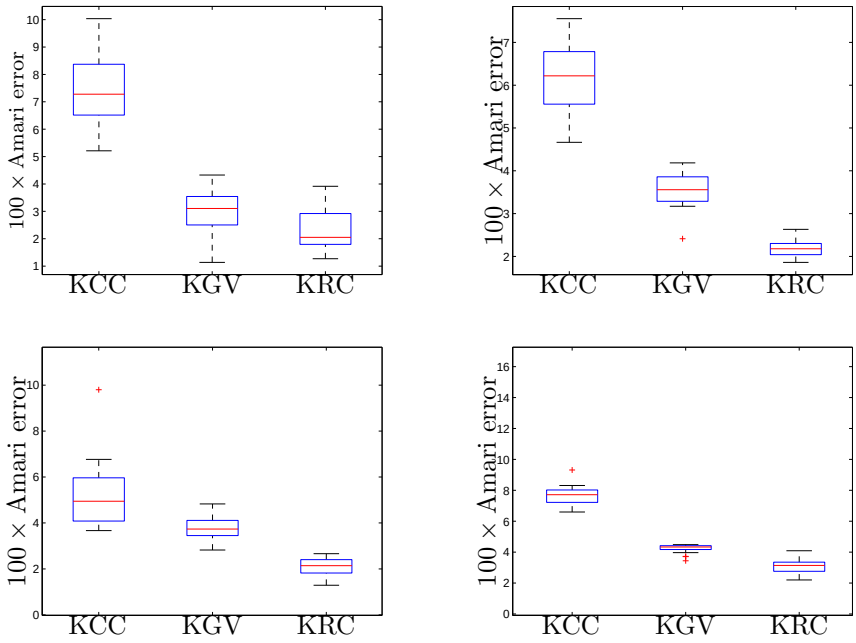


Figure 5.3: Amari errors of the source distributions experiment for the KCC, KGV and the proposed KRC contrast after 20 runs with different mixing matrices. **Top left:** 3 sources. **Top right:** 4 sources. **Bottom left:** 5 sources. **Bottom right:** 6 sources. The KRC performs better with respect to the KCC and KGV.

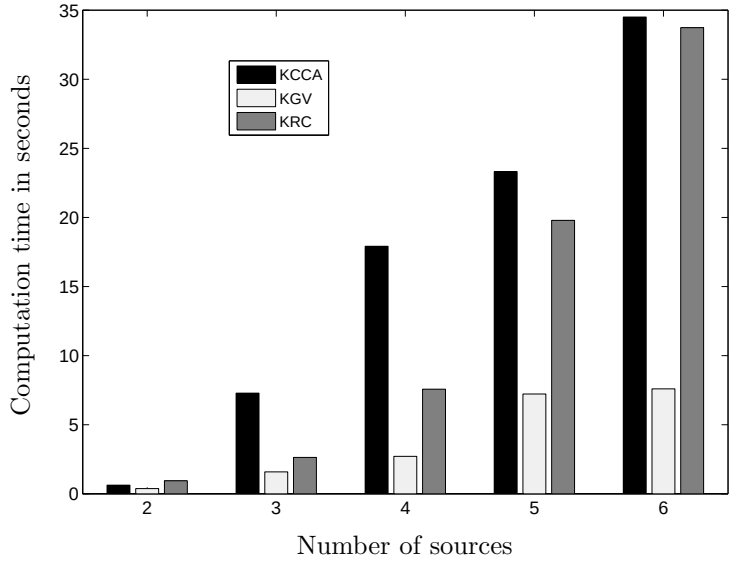


Figure 5.4: Mean computation times of the source distributions experiment using kernel-based contrast functions for ICA. The KRC compares favorably with respect the KCC contrast.

of $N = 10,000$ consecutive data points. The fragments were combined with a randomly generated mixing matrix. The KCC, KGV and the proposed KRC were initialized with Jade and the parameters were fixed to the defaults. Figure 5.5 shows the performance and computation times. The KRC performs better than KCCA and KGV in terms of the Amari error with computation time comparable to the KGV. The Jade algorithm provides a fast and efficient way to initialize the KRC. Even though speech signals violate the i.i.d. data assumption, ICA has been applied to speech demixing with good results.

5.4.4 Image demixing

We used three 140×200 pixels images from the Berkeley image dataset [75]. The mixed images were divided in 4 parts. The upper left part was used as training data and the lower right part as validation data. The two remaining regions were considered as test data. The RBF kernel parameter $\sigma^2 = 0.57$ was tuned using the proposed contrast function on validation data (5.10).

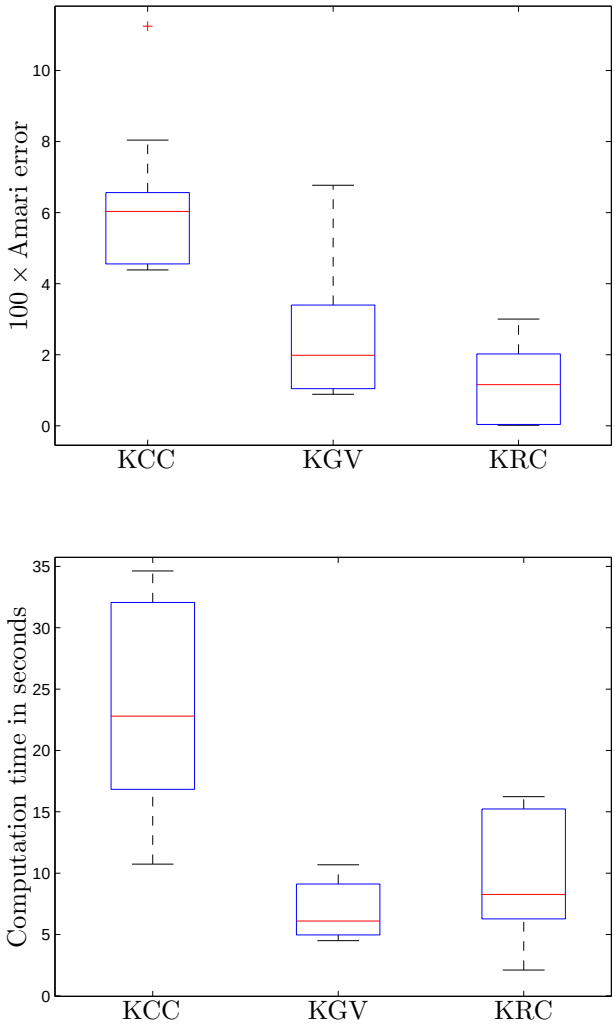


Figure 5.5: Speech demixing results after 10 runs with different segments. **Top:** Amari errors. **Bottom:** Computation times. The KRC outperforms the KCC and KGV in terms of Amari error with comparable computation times with respect to the KGV.

Figure 5.6 shows the model selection curve. The regularization parameter was fixed to $\nu = 1.00$. Figure 5.7 shows the images together with the results. Note that the estimated demixing matrix computed for training data

generalizes for unseen data. The use of validation points to tune the kernel parameter ensures reliability of the proposed measure of independence. The size of the matrix involved in the eigendecomposition was reduced from $84,000 \times 84,000$ to 56×56 using the incomplete Cholesky decomposition with $\eta = 1 \times 10^{-4}$. The mean Amari errors after 10 runs are 0.34 for the KCCA, 0.16 for the KGV, 0.30 for the FastICA algorithm and 0.155 for the proposed KRC.

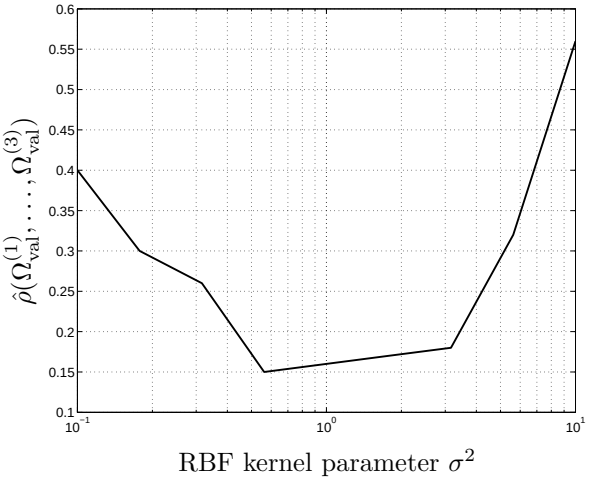


Figure 5.6: Image demixing experiment. Model selection curve for a fixed regularization parameter $\nu = 1.00$. The plot shows the proposed contrast function evaluated on validation data. The obtained tuned RBF kernel parameter is $\sigma^2 = 0.57$.

5.5 Conclusions

A new kernel based contrast function for ICA is proposed. This function called the kernel regularized correlation (KRC) is a measure of independence in the input space and a measure of regularized correlation in the feature space induced by RBF kernels. This formulation is based on LS-SVMs and follows from a clear constrained optimization framework providing primal-dual insights and leading to a naturally regularized measure of independence. This approach also provides out-of-sample extensions for test points which is important for model selection ensuring statistical reliability of the measure for independence. Simulations with toy datasets show improved performance

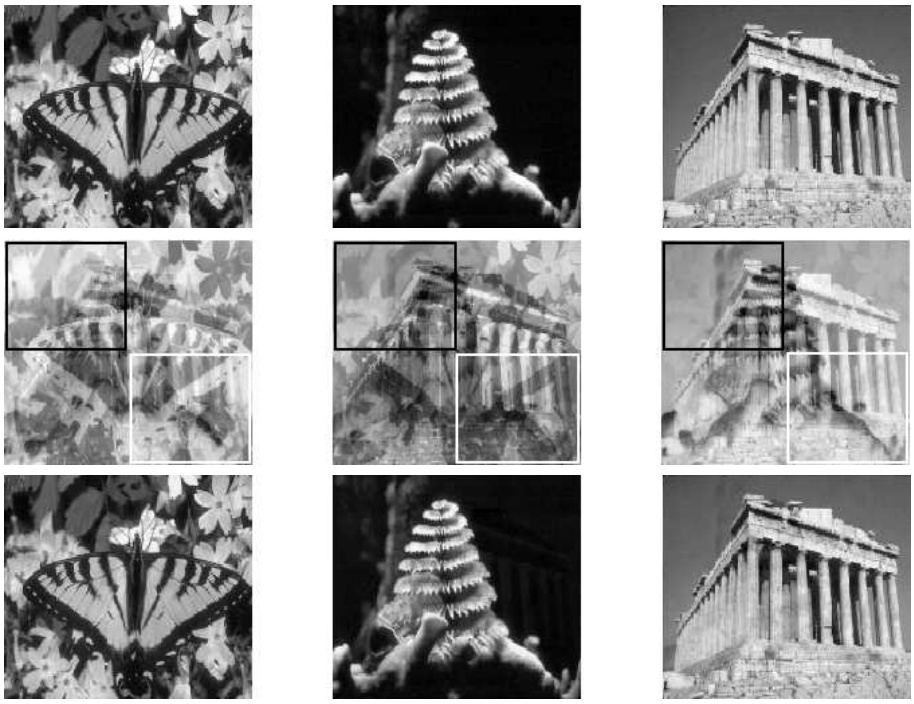


Figure 5.7: Image demixing experiment. **Top row:** Original independent components. **Middle row:** Observed mixtures. The black squares show the regions used for training. Validation regions are shown as white squares. All remaining pixels are considered as test points. **Bottom row:** Estimated independent components using the KRC with tuned RBF kernel width $\sigma^2 = 0.57$. The regularization parameter was fixed to $\nu = 1.00$. Note that the estimated components are visually appealing with an Amari error of 0.15.

compared to existing ICA approaches. The computational complexity of the KRC compares favorably to similar kernel-based methods such as the KCCA and the KGV. Experiments with images and speech using the incomplete Cholesky decomposition to solve a reduced generalized eigenvalue problem also delivered good results.

Chapter 6

Spectral Clustering with Out-of-Sample Extensions

Another contribution of this thesis is presented in this chapter. The weighted kernel PCA framework previously used in chapter 4 for introducing sparseness and robustness into kernel PCA is extended towards spectral clustering methods. One of the main drawbacks of spectral clustering is the fact that the solutions come directly from matrix decomposition problems without clear extension to out-of-sample points. The methods described in [21, 46] allow extensions to new points by approximating the implicit eigenfunction using the Nyström method [18, 115]. The underlying clustering model is usually not known and the parameter selection is done in a heuristic way. In this chapter, a multiway spectral clustering model based on weighted kernel PCA is proposed. This interpretation allows the clustering model to be extended to out-of-sample data points through projections onto the eigenvector solutions. The eigenvectors and projections display a special structure when the clusters are well formed. This structural property is exploited through a proposed model selection criterion called the balanced line fit (BLF). This chapter is organized as follows: Section 6.1 contains a description of the existing weighted kernel PCA approach to binary spectral clustering. In Section 6.2, we propose the multiway spectral clustering model. Section 6.3 contains the

out-of-sample extension together with an algorithm. In Section 6.4, we discuss about model selection and propose the BLF criterion. Section 6.5 contains the toy illustrative examples. Section 6.6 shows an application towards image segmentation. In Section 6.7, we discuss an application towards power load time series clustering and in Section 6.8, we apply the proposed model to scientific journal clustering.

6.1 Preliminaries

Consider the weighted kernel PCA formulation introduced in section 4.1 without centering:

$$\begin{aligned} \min_{w,e_i} \quad & \frac{\gamma}{2N}\sum_{i=1}^{N} v_i e_i^2 - \frac{1}{2}w^T w \\ \text{such that} \quad & e_i = w^T\varphi(x_i),\ i = 1,...,N. \end{aligned} \tag{6.1}$$

From the Lagrangian

$$\mathcal{L}(w, e_i; \alpha_i) = \frac{\gamma}{2N}\sum_{i=1}^{N} v_i e_i^2 - \frac{1}{2}w^T w - \sum_{i=1}^{N}\alpha_i(e_i - w^T\varphi(x_i)),$$

the KKT conditions for optimality are $\partial\mathcal{L}/\partial w = 0$,$\partial\mathcal{L}/\partial e_i = 0$,$\partial\mathcal{L}/\partial\alpha_i = 0$. Eliminating w, e_i, defining $\lambda = N/\gamma$ and applying Mercer's theorem leads to the eigenvalue problem:

$$V\Omega\alpha = \lambda\alpha$$

where $V = \text{diag}([v_1, \ldots, v_N])$ and $\alpha = [\alpha_1; \ldots; \alpha_N]$. Note that, if the weighting matrix V is set to the inverse of the graph degree matrix $V = D^{-1}$, the eigenvalue problem becomes equivalent to the random walks model and the binary NCut discussed in section 2.5. This interpretation of spectral clustering as a special case of weighted kernel PCA [1] allows extensions to out-of-sample data by means of projections onto the eigenvectors and will be discussed in the sequel.

[1] A different interpretation of spectral clustering as a special case of weighted kernel k-means has been discussed in [37,38]. This interpretation allows to solve spectral clustering algorithms in an iterative way without computing eigenvectors. However, out-of-sample extensions have not been discussed.

6.2 Spectral Clustering and Weighted Kernel PCA

6.2.1 Multiway Formulation

The multiclass problem in classification consists of obtaining n_c classes by using a set of n_y binary classification problems. The number of additional classifiers is an important issue and typically corresponds to different encoding/decoding schemes. A unique codeword $y_l \in \{-1,1\}^{n_y}, l = 1,\ldots,n_c$ is assigned to each class. The *one-vs-all* coding scheme consists of setting n_e equal to the number of classes n_c and making binary decisions between each class and all the other classes. In *one-vs-one*, the number of binary classifiers is set to $n_c(n_c-1)/2$ and the discrimination is made for each possible pair of classes. The number of one-vs-one classifiers is usually higher than the number of classifiers in the one-vs-all case, but the resulting decision boundaries are typically simpler. Another encoding scheme is to use n_y problems to encode up to 2^{n_y} classes. The error-correcting output code (ECOC) [39] approach usually introduces redundancy ($n_y > n_c$) and a careful design of codewords (e.g. good Hamming distance between codewords, codewords motivated by known features). In order to classify a new point, the output variables are then compared with the set of codewords and the new point is assigned to the class of the closest codeword in terms of Hamming distance. This scheme has shown good results in terms of robustness against some errors induced by noise.

Given training data $\mathcal{D} = \{x_i\}_{i=1}^N$, consider the following weighted kernel PCA model as an extension of (6.1) with $V = D^{-1}$, n_e additional regularization constants, score variables and bias terms:

$$\begin{aligned} \min_{w^{(l)},e^{(l)},b_l} \quad & \frac{1}{2N}\sum_{l=1}^{n_e}\gamma_l e^{(l)^T} D^{-1} e^{(l)} - \frac{1}{2}\sum_{l=1}^{n_e} w^{(l)^T} w^{(l)} \\ \text{such that} \quad & \begin{cases} e^{(1)} = \Phi w^{(1)} + b_1 1_N \\ e^{(2)} = \Phi w^{(2)} + b_2 1_N \\ \vdots \\ e^{(n_e)} = \Phi w^{(n_e)} + b_{n_e} 1_N. \end{cases} \end{aligned} \tag{6.2}$$

The additional n_e score variables can be interpreted as encoding k clusters in a similar way as the multiclass LS-SVM models discussed in [102]. Each score variables $e^{(l)}$ provides a binary clustering with cluster indicator $q^{(l)} = \text{sign}(e^{(l)}), l = 1,\ldots,n_e$. The dual solutions to (6.2) are formalized in

the following Lemma.

Lemma 6.1. ***Alzate & Suykens [11]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$, regularization constants $\gamma_l \in \mathbb{R}^+, l = 1, \ldots, n_e$ and the inverse of the graph degree matrix D^{-1} which is diagonal with positive elements, then the Karush-Kuhn-Tucker (KKT) optimality conditions of the Lagrangian of (6.2) are satisfied by the following eigenvalue problem:*

$$D^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)} \tag{6.3}$$

where $\lambda_l = N/\gamma_l, l = 1, \ldots, n_e$ and

$$M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N} 1_N 1_N^T D^{-1}. \tag{6.4}$$

Proof: Consider the Lagrangian of the problem (6.2):

$$\begin{aligned}\mathcal{L}(w^{(l)}, e^{(l)}, b_l; \alpha^{(l)}) = \frac{1}{2} \sum_{l=1}^{n_e} \gamma_l {e^{(l)}}^T D^{-1} e^{(l)} - \frac{1}{2} \sum_{l=1}^{n_e} {w^{(l)}}^T w^{(l)} \\ - \sum_{l=1}^{n_e} {\alpha^{(l)}}^T \left(e^{(l)} - \Phi^{(l)} w^{(l)} - b_l 1_N\right)\end{aligned}$$

with KKT optimality conditions:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w^{(l)}} = 0 \to w^{(l)} = \Phi^T \alpha^{(l)} \\ \dfrac{\partial \mathcal{L}}{\partial e^{(l)}} = 0 \to \alpha^{(l)} = \gamma_i D^{-1} e^{(l)} \\ \dfrac{\partial \mathcal{L}}{\partial b_l} = 0 \to 1_N^T \alpha^{(l)} = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \alpha^{(l)}} = 0 \to e^{(l)} = \Phi w^{(l)} + b_l 1_N, \end{cases} \tag{6.5}$$

for $l = 1, \ldots, n_e$. The bias terms become:

$$b_l = -\frac{1}{1_N^T D^{-1} 1_N} 1_N^T D^{-1} \Omega \alpha^{(l)}, l = 1, \ldots, n_e. \tag{6.6}$$

Eliminating the primal variables $w^{(l)}, e^{(l)}, b_l$ leads to the eigenvalue problem (6.3). ■

The cluster indicators for the training points become

$$q^{(l)} = \text{sign}(e^{(l)}) = \text{sign}(\Omega\alpha^{(l)} + b_l 1_N), l = 1, \ldots, n_e.$$

Remark 6.1 (Relation to random walks and centering)**.** *Note that, the eigenvalue problem* (6.3) *is related to the random walks model but considering a different centering induced by the bias terms. The matrix M_D can be interpreted as a weighted centering matrix removing the weighted mean from each column of Ω. The weights are determined by $D^{-1}1_N$. One of the effects of this centering is that the eigenvector $\alpha^{(1)}$ corresponding to the largest eigenvalue λ_1 already contains information about the clustering. This is not the case in the random walks algorithm where the top eigenvector $r^{(1)}$ corresponds to 1_N, thus providing no discriminatory information. Consider the score variables of a test point x with the following centering in the feature space:*

$$z^{(l)}(x) = w^{(l)^T}\big(\varphi(x) - \hat{\mu}_{\varphi_D}\big),$$

where $\hat{\mu}_{\varphi_D} = \sum_{j=1}^{N} D_{jj}^{-1}\varphi(x_j)/(1_N^T D^{-1} 1_N)$ is the weighted mean and $l = 1, \ldots, n_e$. Applying the first optimality condition (6.5) *leads to:*

$$z^{(l)}(x) = \sum_{i=1}^{N} \alpha_i^{(l)} K(x_i, x) + b_l, l = 1, \ldots, n_e.$$

This shows that the score variables for test points are also centered with respect to the weighted mean in the feature space.

6.2.2 Encoding / Decoding Scheme

In this formulation, the encoding problem consists of defining the number of additional score variables n_e needed to encode k clusters. If the eigenvectors are piecewise constant then every cluster $\mathcal{A}_p, p = 1, \ldots, k$ is represented with a unique codeword $c_p \in \{-1, 1\}^{n_e}$. The codebook $\mathcal{C} = \{c_p\}_{p=1}^{k}$ can then be obtained from the rows of the binarized eigenvector matrix $[\text{sign}(\alpha^{(1)}), \ldots, \text{sign}(\alpha^{(n_e)})]$. Another effect of the centering induced by the bias terms is the fact that the eigenvectors have zero mean. This is important for encoding since the optimal threshold for binarizing the eigenvectors is automatically determined. Taking into account that the first eigenvector $\alpha^{(1)}$ already provides a binary clustering then number of score variables needed to encode k clusters is $n_e = k - 1$. The decoding scheme

consists of comparing the obtained cluster indicators with the codebook and selecting the nearest codeword in terms of Hamming distance. This scheme corresponds to the ECOC decoding procedure.

6.3 Out-of-Sample Extension

Spectral clustering methods provide a clustering only for the given training data without a clear extension to out-of-sample (test) points. This issue was first mentioned in [21] and applies also to several spectral algorithms for manifold learning such as local linear embedding (LLE) [91], isomap [106] and Laplacian eigenmaps [20]. A link between kernel PCA and manifold learning was presented in [52], where LLE, isomap and Laplacian eigenmaps and locality preserving projections (LPP) [53] are interpreted as kernel PCA with different kernel matrices. However, these techniques are more suited to data visualization than to clustering since the problem of obtaining clusters from the out-of-sample extensions is not clear. The out-of-sample method proposed in [21] consists of applying the Nyström method [18] in order to provide an embedding for the test points by approximating the underlying eigenfunction. Our proposed extension is based on the score variables which correspond to the projections of the mapped out-of-sample points onto the eigenvectors found in the training stage. The cluster indicators can be obtained by binarizing the score variables for out-of-sample points as follows

$$\text{sign}(z^{(l)}) = \text{sign}(\Omega_{\text{test}}\alpha^{(l)} + b_l 1_{N_{\text{test}}}), l = 1, \ldots, k-1. \tag{6.7}$$

where Ω_{test} is the $N_{\text{test}} \times N$ kernel matrix evaluated using the test points with entries $\Omega_{\text{test}}^{mi} = K(x_m^{\text{test}}, x_i), m = 1, \ldots, N_{\text{test}}, i = 1, \ldots, N$. This natural extension to out-of-sample points corresponds to the main advantage of the weighted kernel PCA framework. In this way, the clustering model can be trained, validated and tested in an unsupervised learning scheme. The proposed algorithm for multiway spectral clustering with out-of-sample extensions is described in Algorithm 5.

Proposition 6.1. ***Alzate & Suykens [11]*** *Consider a training dataset $\mathcal{D} = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$ sampled i.i.d. from an underlying distribution $\mathcal{P}$, a validation dataset $\mathcal{D}^v = \{x_m^v\}_{m=1}^{N_v}, x_m \in \mathbb{R}^d$ also sampled i.i.d. from $\mathcal{P}$ and an isotropic kernel function $K(x_i, x_j), i, j = 1, \ldots, N$. Let Ω be the kernel matrix with ij-entry $\Omega_{ij} = K(x_i, x_j)$ and let $\alpha^{(l)}, l = 1, \ldots, k-1$ be the eigenvectors of* (6.3)*. Consider the following assumptions:*

Algorithm 5 Multiway Spectral Clustering Algorithm

Input: Training set $\{x_i\}_{i=1}^N$, test set $\{x_m^{\text{test}}\}_{m=1}^{N_{\text{test}}}$, positive definite kernel function $K(x_i, x_j)$, number of clusters k.

Output: Clusters $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$, cluster codeset $\mathcal{C} = \{c_p\}_{p=1}^k$, $c_p \in \{-1, 1\}^{k-1}$.

1: Compute the eigenvectors $\alpha^{(l)}, l = 1, \ldots, k-1$ corresponding to the largest $k-1$ eigenvalues of $D^{-1}M_D\Omega$ where M_D is defined as in (6.4) and Ω is the training kernel matrix $\Omega_{ij} = K(x_i, x_j)$.

2: Binarize the eigenvectors: $\text{sign}(\alpha_i^{(l)}), i = 1, \ldots, N, l = 1, \ldots, k-1$, and let $\text{sign}(\alpha_i) \in \{-1, 1\}^{k-1}$ be the encoding vector for the training data point x_i, $i = 1, \ldots, N$.

3: Count the occurrences of the different encodings and find the k encodings with most occurrences. Let the codeset be formed by these k encodings: $\mathcal{C} = \{c_p\}_{p=1}^k, c_p \in \{-1, 1\}^{k-1}$

4: $\forall i$, assign x_i to $\mathcal{A}_{p^\star}$ where $p^\star = \text{argmin}_p d_H(\text{sign}(\alpha_i), c_p)$ and $d_H(\cdot, \cdot)$ is the Hamming distance.

5: Binarize the test data projections $\text{sign}(z_m^{(l)}), m = 1, \ldots, N_{\text{test}}, l = 1, \ldots, k-1$ using (6.7) and let $\text{sign}(z_m) \in \{-1, 1\}^{k-1}$ be the encoding vector of x_m^{test}, $m = 1, \ldots, N_{\text{test}}$.

6: $\forall m$, assign x_m^{test} to $\mathcal{A}_{p^\star}$ where $p^\star = \text{argmin}_p d_H(\text{sign}(z_m), c_p)$.

- *$\mathcal{P}$ contains k clusters denoted by $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$.*
- *The eigenvectors $\alpha^{(l)}, l = 1, \ldots, k-1$ are piecewise constant.*
- *$K(x_i, x_j) \to 0$ when x_i and x_j are in different clusters.*

Then, validation points in the same cluster are collinear in the $k-1$ dimensional subspace spanned by the columns of Z^v, where $Z^v \in \mathbb{R}^{N_v \times (k-1)}$ is the validation score variables matrix with ml-entry $Z_{ml}^v = z_m^{(l)} = \sum_{i=1}^N \alpha_i^{(l)} K(x_i, x_m^v)$ and $m = 1, \ldots, N_v, l = 1, \ldots, k-1$.

Proof: The eigenvectors $\alpha_i^{(l)}$ are piecewise constant, therefore $\alpha_j^{(l)} = c_p^{(l)}$, for all $x_j \in \mathcal{A}_p$, $l = 1, \ldots, k-1$, $p = 1, \ldots, k$ where $c_p^{(l)}$ is the constant value corresponding to the p-th cluster in the l-th eigenvector. The projections become:

$$z_m^{(l)} = c_p^{(l)} \sum_{j \in \mathcal{A}_p} K(x_j, x_m^v) + \sum_{u \notin \mathcal{A}_p} \alpha_u^{(l)} K(x_u, x_m^v), \tag{6.8}$$

$l = 1, \ldots, k-1$, $p = 1, \ldots, k$, $m = 1, \ldots, N_v$. Note that $K(x_u, x_m^v)$ tends to zero when x_u and x_m^v are not in the same cluster, therefore the term $\sum_{u \notin \mathcal{A}_p} \alpha_u^{(l)} K(x_u, x_m^v)$ is very small compared to the term $c_p^{(l)} \sum_{j \in \mathcal{A}_p} K(x_j, x_m^v)$. In this case, validation points belonging to the same cluster are represented as collinear points in the score variables space. ■

The proposed encoding scheme is based on an arbitrary set of orthogonal hyperplanes induced by the eigenproblem. Due to the fact that the eigenvectors can be arbitrarily rotated, the hyperplanes can bisect data points in the same cluster if the data are not well represented. However, this is improbable for well represented data because in that case, data points belonging to the same cluster will be located in the same orthant.

6.4 Model Selection

The results obtained using spectral clustering depend greatly on the choice of the affinity function and its parameters. The most used affinity function is the radial basis function (RBF) kernel $K(x_i, x_j) = \exp(-0.5||x_i - x_j||_2^2/\sigma^2)$ with tuning parameter σ^2. The selection of this parameter is not considered as part of the learning process and is commonly done in a manual way using grid search and minimizing some cluster distortion function [84]. However, counterexamples can be easily found in which clusters with small distortion do not correspond to the natural grouping of the data. The tuning method proposed in [118] introduces local scaling by selecting a σ_i parameter per data point x_i. The selection of this scaling parameter is done in a heuristic way using the distance of the point x_i to some nearest neighbor. Again, the results of this approach depend on the choice of the nearest neighbor. A different method that can be used to learn the parameters given a particular partition was proposed in [16]. In this case, the method corresponds to a supervised setting and the clustering cost function characterizes the ability of the induced affinity matrix to produce clusters that fit the given partition. The selection of the number of clusters k is also a general problem for all clustering methods and several approaches have been proposed. Typical algorithms are the eigengap heuristic, the elbow criterion, the gap statistic [107] and the silhouette index [89]. The simplest approach is the eigengap heuristic which consists of choosing a value for k such that the eigengap $|\lambda_{k+1} - \lambda_k|$ is greater than some threshold value θ. The justification of this procedure relies on the fact that in the ideal case when the graph has k

disconnected components, the multiplicity of the eigenvalue 1 is k. Thus, the $(k+1)$-th eigenvalue will be less than 1 resulting in some eigengap. In this ideal case, the threshold can be set to zero, but in practical situations this simple heuristic absolutely fails. The elbow criterion is a simple measure of the compactness of the clusters and the gap statistic together with the silhouette index are advanced cluster distortion indicators.

The proposed model selection criterion is called the *balanced line fit* (BLF) and can be used for estimating suitable spectral clustering parameters, namely the RBF kernel parameter σ^2 (or the χ^2 kernel parameter σ_χ^2) and the number of clusters k. This criterion exploits the eigenstructure of piecewise constant eigenvectors and the corresponding projections. Consider the following average measure of collinearity of a validation dataset $\mathcal{D}^v$ with respect to a clustering $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}, k > 2$

$$\text{linefit}(\mathcal{D}^v, k) = \frac{1}{k}\sum_{p=1}^{k} \frac{k-1}{k-2}\left(\frac{\zeta_1^{(p)}}{\sum_l \zeta_l^{(p)}} - \frac{1}{k-1}\right) \tag{6.9}$$

where $\zeta_1^{(p)} \geq \ldots \geq \zeta_{k-1}^{(p)}$ are the ordered eigenvalues of the sample covariance matrix:

$$C_{\tilde{Z}}^{(p)} = \frac{1}{|\mathcal{A}_p|}\tilde{Z}^{(p)^T}\tilde{Z}^{(p)}, p = 1, \ldots, k, \tag{6.10}$$

$\tilde{Z}^{(p)} \in \mathbb{R}^{|\mathcal{A}_p|\times(k-1)}$ is the matrix representing the zero mean score variables for validation data assigned to the p-th cluster. The term $\zeta_1^{(p)}/\sum_l \zeta_l^{(p)}$ indicates how much of the total variance is contained on the eigenvector corresponding to the largest eigenvalue of $C_{\tilde{Z}}^{(p)}$. In this way, if the validation score variables for the p-th cluster are collinear, then all variance is contained on the first eigenvector and $\zeta_1^{(p)}/\sum_l \zeta_l^{(p)}$ equals 1. On the other hand, if the total variance is evenly distributed on the eigenvectors then $\zeta_1^{(p)}/\sum_l \zeta_l^{(p)}$ equals $1/(k-1)$. The additional terms in (6.9) fix the criterion such that the linefit equals 0 when the score variables are distributed spherically (i.e. the eigenvalues are identical) and equals 1 when the score variables are collinear.

Note that the linefit is defined only for $k > 2$. When $k = 2$ only one eigenvector is needed to obtain a binary clustering. Consider the matrix $\breve{Z}^v \in \mathbb{R}^{N_v\times 2}$ where the m-th entry in the first column corresponds to the score variables $z_m = \sum_{i=1}^{N} \alpha_i K(x_i, x_m^v)$ and the m-th entry in the second column to $\sum_{i=1}^{N} K(x_i, x_m^v)$. This modification allows the application of the

linefit criterion when $k = 2$:

$$\text{linefit}(\mathcal{D}^v, k) = \begin{cases} \sum_{p=1}^{2} \left(\frac{\breve{\zeta}_1^{(p)}}{\breve{\zeta}_1^{(p)} + \breve{\zeta}_2^{(p)}} - \frac{1}{2} \right) & \text{if } k = 2 \\ \frac{1}{k} \sum_{p=1}^{k} \frac{k-1}{k-2} \left(\frac{\zeta_1^{(p)}}{\sum_l \zeta_l^{(p)}} - \frac{1}{k-1} \right) & \text{if } k > 2 \end{cases} \tag{6.11}$$

where $\breve{\zeta}_l^{(p)}$ are the ordered eigenvalues of the matrix $\breve{Z}^{(p)} \in \mathbb{R}^{|\mathcal{A}_p| \times 2}$ representing the zero mean score variables for validation data assigned to the p-th cluster when $k = 2$.

In real-life problems, the balance of the obtained clusters also becomes important. Consider the following clustering balance measure:

$$\text{balance}(\mathcal{D}^v, k) = \frac{\min\{|\mathcal{A}_1|, \ldots, |\mathcal{A}_k|\}}{\max\{|\mathcal{A}_1|, \ldots, |\mathcal{A}_k|\}}. \tag{6.12}$$

The balance index equals 1 when the clusters have the same number of elements and tends to 0 in extremely unbalanced cases. Combining the linefit with the balance index leads to the BLF:

$$\text{BLF}(\mathcal{D}^v, k) = \eta \text{linefit}(\mathcal{D}^v, k) + (1 - \eta)\text{balance}(\mathcal{D}^v, k), \tag{6.13}$$

where η is a parameter controlling the importance given to the linefit with respect to the balance index and $0 \leq \eta \leq 1$.

6.5 Examples

In this section, some experimental examples are presented to illustrate the proposed approach. All experiments reported are carried out in MATLAB on an Intel Dual Core, 3.0 GHz, 2GB RAM. All data have been separated into training, validation and test sets. The BLF criterion is used on validation data to find the number of clusters k and the kernel parameters and the η parameter is fixed to $\eta = 0.75$. Hence, more emphasis is given to the linefit. Simulations on toy data are presented. The adjusted Rand index (ARI) [59] is used as an external validation in order to compare the clustering results with some external cluster indicators. This clustering performance index ranges from 0 to 1 and takes the unitary value when the clustering results matches perfectly the external cluster indicators.

We compare our proposed method in terms of performance and computation times with the spectral clustering scheme discussed in [46]. This method uses

random subsampling to solve an eigenvalue problem of a small subset of training points and then extrapolates the eigenvectors to the complete set of data points using the Nyström approximation [18, 115]. The resulting approximated eigenvectors are not mutually orthogonal and additional orthogonalization steps are needed. The cluster indicators are obtained by applying k-means on the approximated orthogonalized eigenvectors. We used the one-shot implementation described in [46].

The first toy problem consists of 3 Gaussian clouds in a 2D space. The total number of data points is 800. The RBF kernel parameter σ^2 was set to 0.08. The training set consisted of 200 randomly selected points. Figure 6.1 shows the binary clustering boundaries induced by the encoding scheme. The first eigenvector $\alpha^{(1)}$ of (6.3) groups the left and bottom Gaussian clouds into one cluster. The second eigenvector $\alpha^{(2)}$ groups the bottom and right clouds into one cluster. The cluster indicators for out-of-sample points were inferred using the out-of-sample extension and the ECOC decoding scheme.

The second toy problem shows the influence of increasing number of clusters k with small training set sizes. Figure 6.2 shows the results of the proposed approach and the Nyström algorithm [46]. This dataset consists of 10 Gaussian clouds for a total of $2,000$ data points. We performed 10 randomizations of the training set and the total number of training points was set to one-fifth of the total number of points. We used the average ARI over the randomizations to assess the performance with respect to the optimal clustering. The proposed method performs better than the Nyström algorithm in terms of ARI.

The third toy problem consists of 3 concentric rings in a 2D space. The training, validation and test sets contain 600, $1,200$ and 800 data points respectively. This problem is known to be difficult for spectral clustering because of its nonlinearity, the fact that every datapoint has few neighbors resulting in a very sparse affinity matrix and the multiscale nature of the rings. Figure 6.3 shows the model selection curves. In this particular case, the BLF and the silhouette index find the correct number of clusters, but for the selection of σ^2 the silhouette index selects a non reliable value. The BLF is very close to the adjusted Rand index yielding maxima in the same range of perfect performance. The eigengap is very small in this experiment, making the eigendecomposition problematic. Despite this fact, the BLF succeeds in selecting optimal parameters.

Figure 6.4 shows the score variables and the clustering results for an optimal

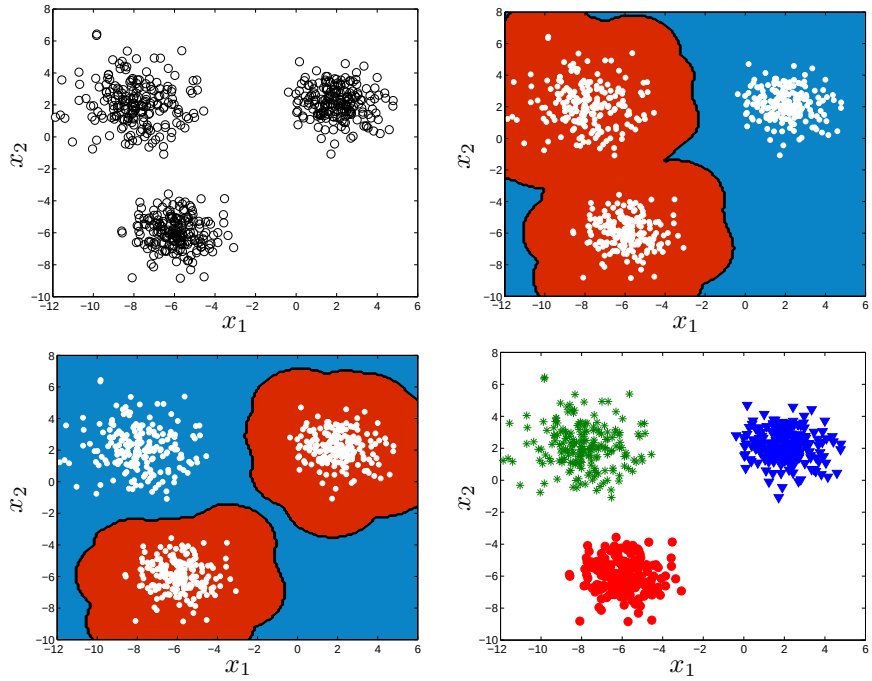


Figure 6.1: Effect of the encoding scheme. **Top left:** Full dataset with 3 clusters and 800 data points. **Top right:** Bipartitioning induced by binarizing the first eigenvector $\alpha^{(1)}$ of (6.3). **Bottom left:** Bipartitioning induced by binarizing the second eigenvector $\alpha^{(2)}$ of (6.3). **Bottom right:** Cluster results after encoding the two binary partitions The RBF kernel parameter was set to $\sigma^2 = 0.08$. The training set consisted of 200 randomly selected data points. The cluster indicators of the remaining data points were inferred using the out-of-sample extension.

value of σ^2 and for a value far away from the optimal range found by the BLF. The score variables in the optimal case are perfect lines resulting in an optimal clustering. In the non-optimal case, the score variables do not show the line structure leading to a wrong clustering. Figure 6.5 shows comparisons of the proposed method with the Nyström method in [46]. We compute the ARI between cluster indicators obtained through different random subsamplings. This can be seen as a measure of pairwise clustering agreement between different randomizations of the training data. Comparisons with respect to the optimal clustering are also reported. The proposed method shows less variability with respect to random subsamplings compared to the Nyström method. This issue can be due to convergence problems of the reclustering step using k-means. In the proposed method,

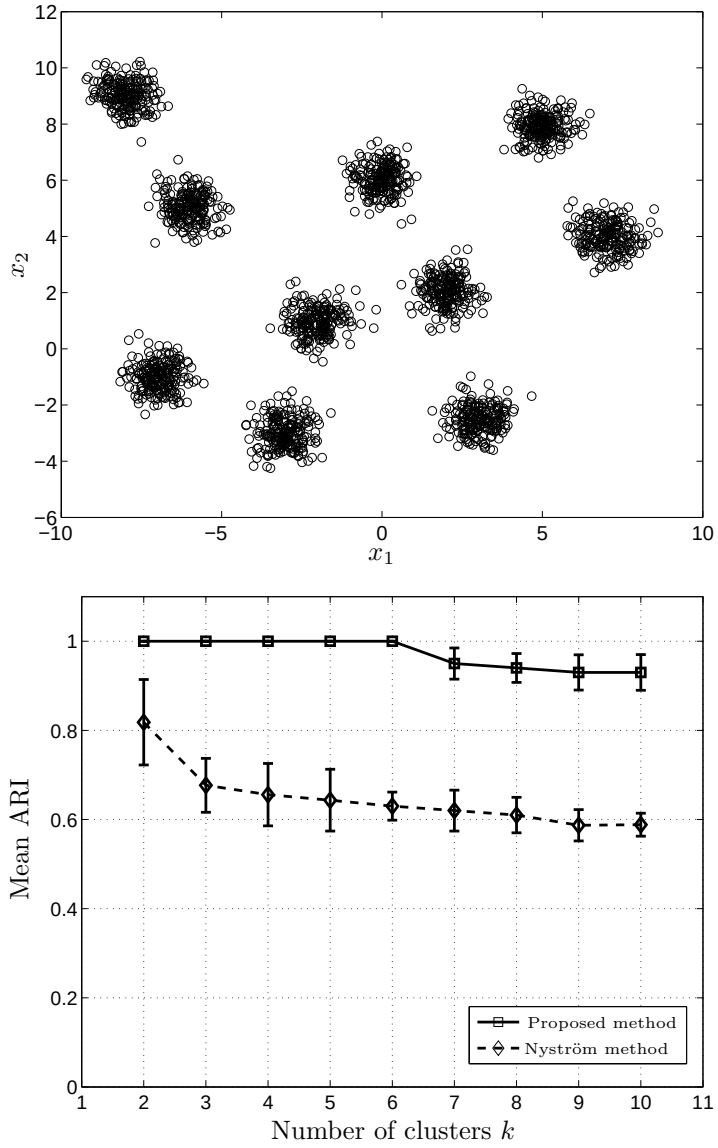


Figure 6.2: ARI with increasing number of clusters. **Top:** Full dataset with 10 clusters and 2,000 data points. **Bottom:** Mean ARI compared to the optimal clustering after 10 randomizations of the training set. The number of training data points was set to one-fifth of the total number of points. **Solid line:** Proposed approach. **Dashed line** Nyström approach [46]. The proposed method performs better than the Nyström method in terms of ARI with respect to an increasing number of clusters and small training set sizes.

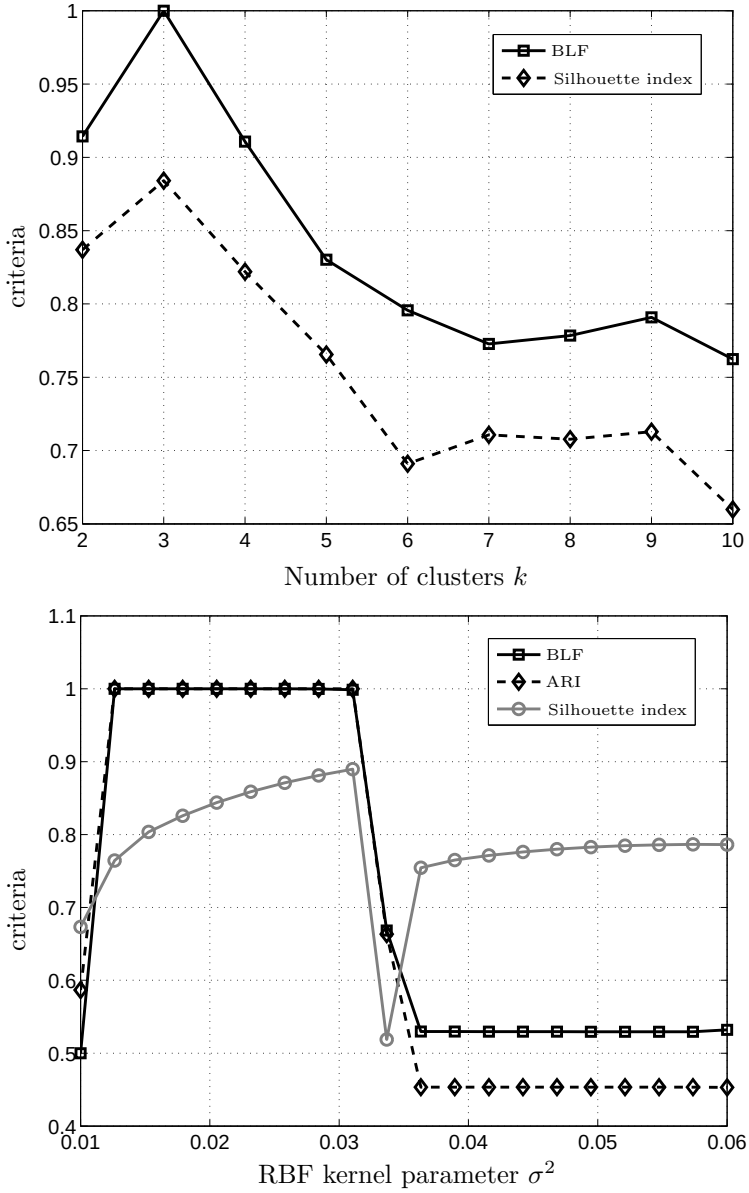


Figure 6.3: Toy problem - Model selection. **Top:** Maximum of the BLF (solid), maximum of the mean silhouette index applied to the score variables (dashed). Both criteria are maximal for $k = 3$. **Bottom:** Selection of σ^2. BLF for $k = 3$ (solid), adjusted Rand index (dashed), mean silhouette index for $k = 3$ (dotted). Note that the BLF is very similar to the adjusted Rand index.

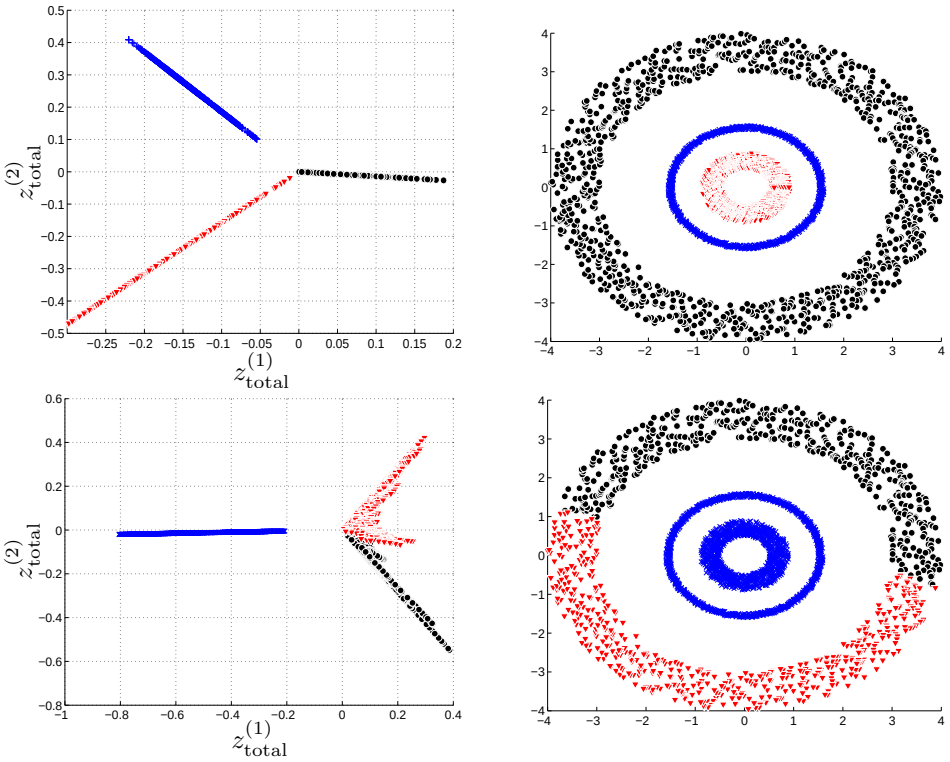


Figure 6.4: Toy problem - **Top left:** Score variables for the full dataset with an optimal $\sigma^2 = 0.02$ according to the BLF on validation data. **Top right:** Clustering results with optimal σ^2. **Bottom left:** Score variables for the full dataset with a non-optimal σ^2. **Bottom right:** Clustering results with a non-optimal σ^2. The triangles (red), crosses (blue) and circles (black) represent the clusters. Note that the score variables form lines in the case of optimal clustering.

the agreement between cluster indicators tends to stabilize as the number of training points increases. Computation times are reported in Figure 6.6.

6.6 Application: Image Segmentation

This application shows the use of the proposed clustering approach for image segmentation. Color images from the Berkeley image dataset[2] [75] were used. We computed a local color histogram with a 5×5 pixels window around

[2]http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

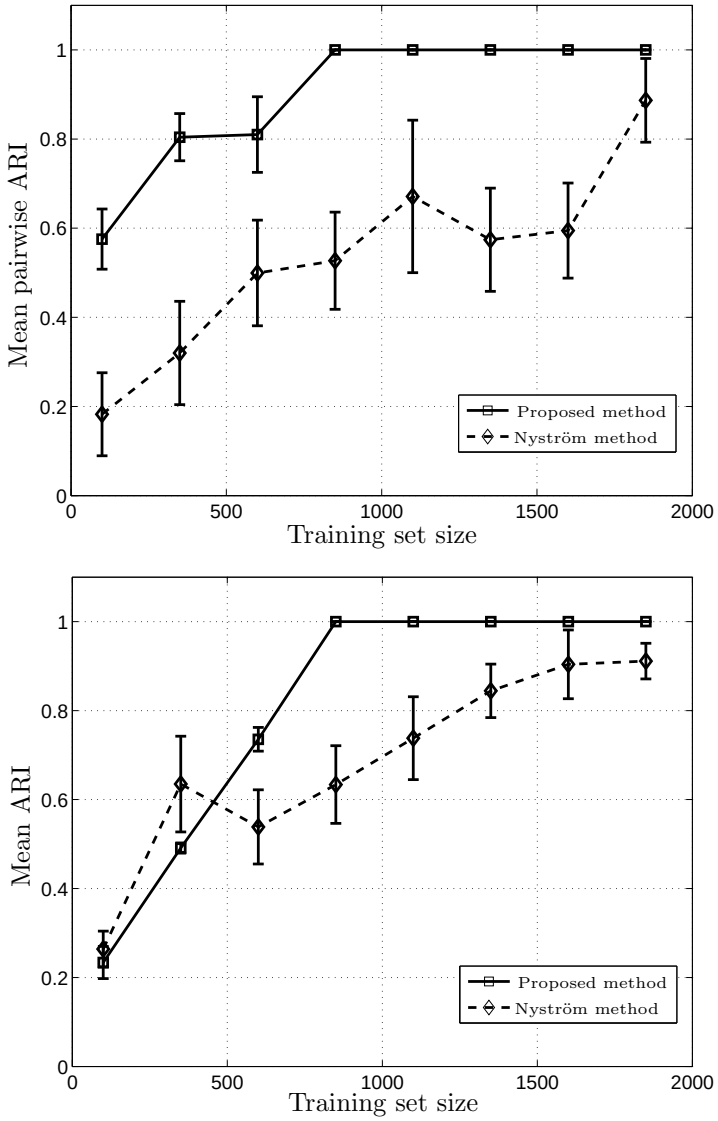


Figure 6.5: Toy problem - Agreement between cluster indicators. **Top:** Mean ARI between pairwise cluster indicators obtained through different random subsamplings. **Bottom:** Mean ARI between cluster indicators and the optimal clustering. The σ^2 parameter was set to 0.02 and 20 random subsamplings were performed. The Nyström method fails to stabilize even with increasing number of samples. The proposed method converges to the optimal clustering as the number of training samples increases.

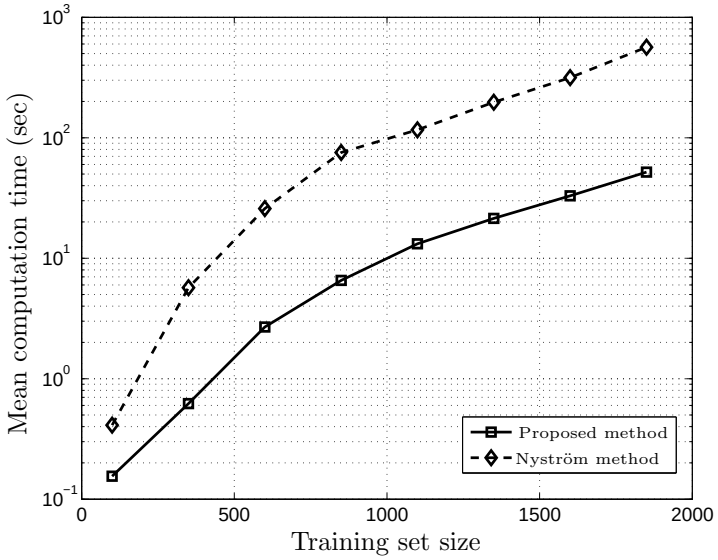


Figure 6.6: Toy problem - Mean computation times. **Solid line:** Proposed method. **Dashed line:** Spectral clustering using Nyström [46]. The main bottleneck in the Nyström algorithm is the orthogonalization of the approximated eigenvectors.

each pixel using minimum variance color quantization of 8 levels. We used the χ^2 test to compute the distance between two local color histograms $h^{(i)}$ and $h^{(j)}$ [88] $\chi^2_{ij} = 0.5 \sum_{b=1}^{B} (h_b^{(i)} - h_b^{(j)})^2 / (h_b^{(i)} + h_b^{(j)})$, where B is total number of quantization levels. The histograms are assumed to be normalized $\sum_{b=1}^{B} h_b^{(i)} = 1, i = 1, \ldots, N$. The χ^2 kernel $K(h^{(i)}, h^{(j)}) = \exp(-\chi^2_{ij}/\sigma_\chi)$ with parameter $\sigma_\chi \in \mathbb{R}^+$ is positive definite and has shown to be a robust and efficient way of comparing the similarity between two histograms for color discrimination and image segmentation [46]. We performed model selection using the BLF for determining the optimal number of clusters k and the χ^2 kernel parameter σ_χ. The training scenario consisted of $1,000$ pixel histograms for training and $20,000$ pixel histograms for validation. The training and validation sets were randomized 20 times. Figure 6.7 shows the model selection results for image with ID 167062. The obtained tuned parameters are $k = 2, \sigma_\chi = 0.09$. Similar tuning plots were used to determine the optimal parameters of each of the images used in this application.

Figure 6.8 shows the obtained clustering and the validation score variables

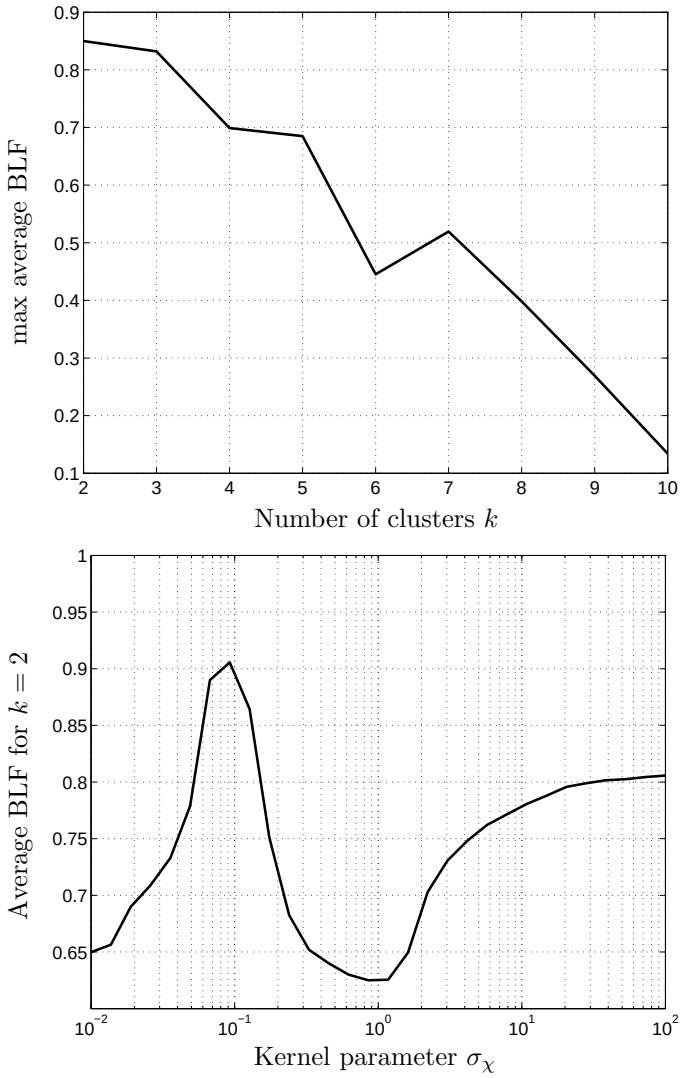


Figure 6.7: Model selection using the BLF for image with ID 167062. **Top:** Tuning the number of clusters k. **Bottom:** Tuning the kernel parameter σ_χ. The model selection scenario consists of $1,000$ pixel histograms for training and $20,000$ pixel histograms for validation. The training and validation sets were randomized 20 times. Tuned parameters are $k = 2, \sigma_\chi = 0.09$.

for image with ID 145086 using tuned $k = 4, \sigma_\chi = 0.084$. Each cluster in the image is represented as a line in the score variables space. Mean computation times can be seen in Figure 6.9. The bottleneck of the Nyström method corresponds to the orthogonalization steps. For training set sizes $N > 2,000$ the proposed method runs faster than [46] by at least one order of magnitude. Figures 6.10 and 6.11 show the image segmentation results for 20 images. After tuning, the methods are trained with $1,000$ pixel histograms randomly chosen. The cluster indicators of the remaining pixel histograms are inferred using the out-of-sample extension in the proposed method and approximated using Nyström in the case of [46]. The clustering indicators represent a segment-label image. The edges are then computed by applying a Canny edge detector over the segment-label image. Table 6.1 shows the comparison between the two methods using a performance measure with respect to human segmentation. The performance criterion used is the F-measure [74,109]. This measure is known to be meaningful for evaluating image segmentation boundaries. The F-measure is closely related to the area under Receiver Operating Characteristic (ROC) curves and is defined as the harmonic mean between precision and recall. The proposed method outperforms the Nyström algorithm in terms of agreement with human segmentation.

6.7 Application: Clustering Load Time Series

Time series data pose a challenge to standard clustering methods due mainly to the high-dimensionality. The particular dataset corresponds to readings of the power load measure at 245 different substations in the Belgian grid for a period of 5 years. The objective is to identify types of costumers underlying the data. This task is particularly important for short, mid and long-term planning in the electricity sector. An approach to cluster this type of data was proposed in [43]. This approach uses a pre-modeling scheme based on periodic autoregressions (PAR) in order to reduce the dimensionality together with representing the time series independent of seasonal and calendar variations. In a later stage, k-means was used to find the clusters. Pre-modeling can be useful for interpretability and dimensionality reduction but there might be a loss of information while trying to pre-model the complete set of load time series with the same template. This clustering application aims at obtaining interpretable customer profiles of power load consumption by clustering the data directly in a learning framework with

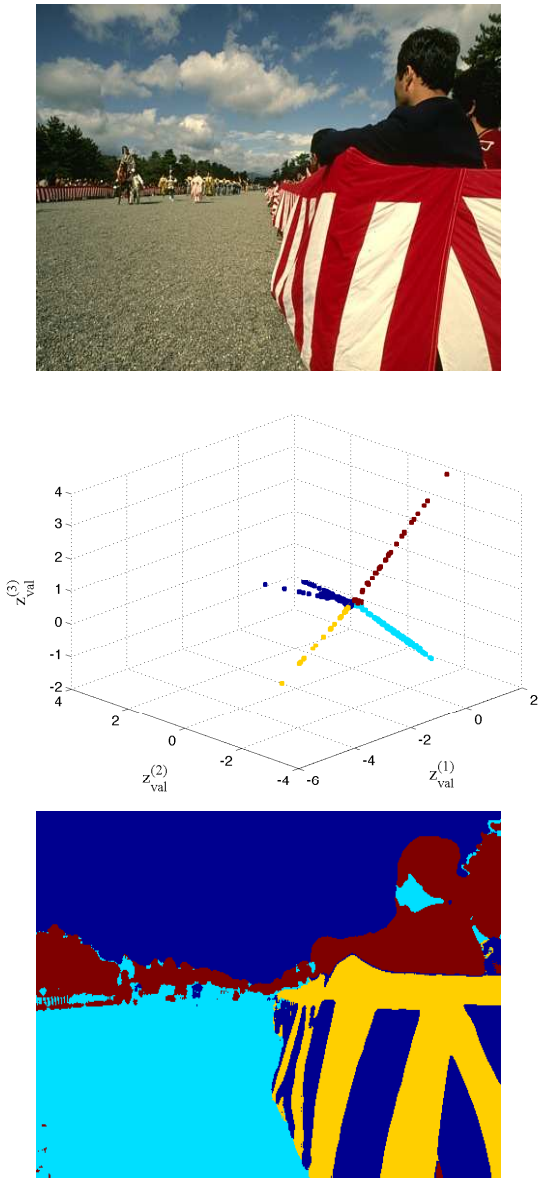


Figure 6.8: Model selection using the BLF. **Top:** Original image. **Center:** Score variables for the validation set. The tuned parameters are $k = 4, \sigma = 0.084$. Note the strong line structures present on the score variables. **Bottom:** Segment-label image.

Image ID	$F-$measure		Human
	(a)	(b)	
145086	**0.88**	0.78	0.85
42049	**0.88**	0.87	0.96
167062	**0.85**	0.46	0.95
147091	**0.8**	0.68	0.87
196073	**0.79**	0.74	0.85
62096	**0.78**	0.76	0.90
101085	0.68	**0.77**	0.93
69015	**0.77**	0.61	0.74
119082	**0.73**	0.62	0.80
3096	**0.72**	0.27	0.74
295087	**0.72**	0.60	0.82
37073	**0.72**	0.64	0.78
182053	0.65	**0.71**	0.75
101087	0.60	**0.71**	0.86
361010	**0.70**	0.58	0.91
299086	**0.70**	0.52	0.82
45096	**0.69**	0.35	0.77
241048	**0.69**	0.56	0.80
216081	0.66	**0.69**	0.86
253027	**0.68**	0.52	0.77
302008	**0.66**	0.59	0.86
385039	0.62	**0.66**	0.83
296007	**0.66**	0.47	0.83
285079	**0.66**	0.47	0.69
54082	**0.65**	0.47	0.68
189080	**0.65**	0.57	0.87
134035	**0.64**	0.61	0.54
126007	**0.64**	0.62	0.87
157055	**0.64**	0.56	0.78
21077	0.64	0.64	0.82
296059	**0.64**	0.53	0.88
41033	**0.64**	0.60	0.88
38092	0.62	**0.63**	0.87
219090	**0.63**	0.58	0.90
24077	0.63	0.63	0.79
306005	**0.63**	0.62	0.67
85048	**0.62**	0.52	0.85
271035	0.58	**0.62**	0.82
241004	**0.62**	0.60	0.94
66053	**0.61**	0.53	0.81
175032	**0.60**	0.55	0.73
42012	**0.60**	0.53	0.79
260058	**0.60**	0.37	0.73
143090	**0.60**	0.59	0.84
197017	**0.60**	0.56	0.89

Image ID	$F-$measure		Human
	(a)	(b)	
208001	**0.60**	0.41	0.78
69020	**0.59**	0.49	0.86
148089	**0.59**	0.54	0.70
223061	**0.59**	0.44	0.60
76053	**0.59**	0.55	0.58
376043	**0.59**	0.50	0.81
175043	**0.59**	0.56	0.50
58060	**0.59**	0.53	0.52
19021	**0.58**	0.52	0.80
159008	**0.58**	0.56	0.64
167083	**0.58**	0.44	0.75
14037	**0.58**	0.41	0.77
86000	**0.58**	0.52	0.73
351093	0.53	**0.57**	0.81
78004	**0.57**	0.54	0.81
300091	0.47	**0.57**	0.84
38082	0.56	0.56	0.74
103070	**0.56**	0.47	0.79
227092	**0.56**	0.49	0.65
106024	0.37	**0.55**	0.84
229036	**0.55**	0.47	0.86
148026	**0.55**	0.38	0.70
87046	**0.55**	0.47	0.82
65033	**0.54**	0.41	0.81
170057	**0.54**	0.32	0.76
291000	**0.53**	0.50	0.80
210088	**0.52**	0.45	0.54
102061	**0.52**	0.46	0.85
220075	**0.52**	0.42	0.78
97033	**0.51**	0.42	0.84
160068	**0.5**	0.42	0.91
41069	**0.5**	0.44	0.79
16077	**0.51**	0.43	0.87
236037	0.48	**0.49**	0.64
156065	**0.49**	0.43	0.69
89072	0.47	**0.49**	0.84
33039	**0.48**	0.42	0.75
304074	**0.48**	0.40	0.93
105025	**0.48**	0.36	0.77
109053	**0.48**	0.42	0.84
163085	**0.47**	0.40	0.73
12084	0.26	**0.47**	0.62
108005	**0.47**	0.41	0.81
253055	**0.47**	0.33	0.95
86016	**0.45**	0.41	0.77

Table 6.1: Performance of the test images from the Berkeley image dataset. **(a)** Proposed method. **(b)** Nyström method [46]. The proposed method performs better than [46] on 77 out of 90 images.

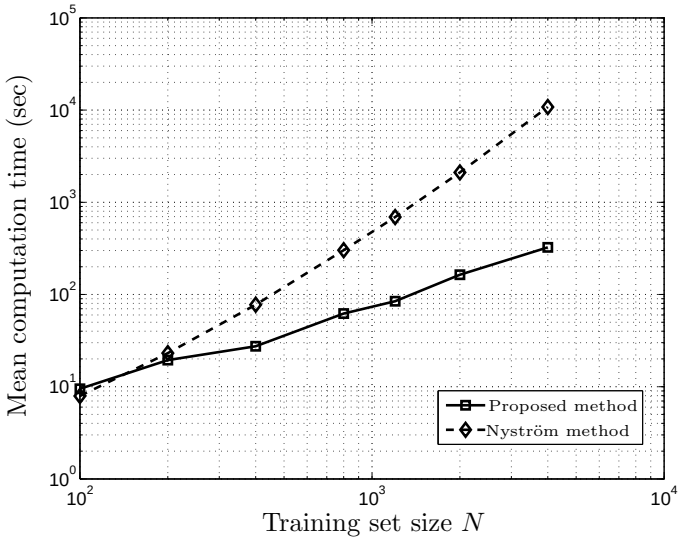


Figure 6.9: Mean computation times for the Segbench test dataset over 10 randomizations of the training set. **Solid line:** Proposed method. **Dashed line:** Nyström method [46] using the one-shot implementation. The main bottleneck in [46] is the orthogonalization of the approximated eigenvectors.

model selection and without the use of pre-modeling steps.

6.7.1 Data Description

This particular dataset consists of time series containing values of power load in 245 different high voltage - low voltage substations within the Belgian national grid operator ELIA for a period of 5 years. The sampling rate is 1 hour. The data characterize different profiles of load consumption such as business, residential and industrial. Each time series was normalized to have zero mean and unitary standard deviation. Figure 6.12 shows one particular time series of the dataset. The yearly cycles are visible together with the daily cycles. From the daily cycle it is also possible to visualize morning, noon and evening peaks.

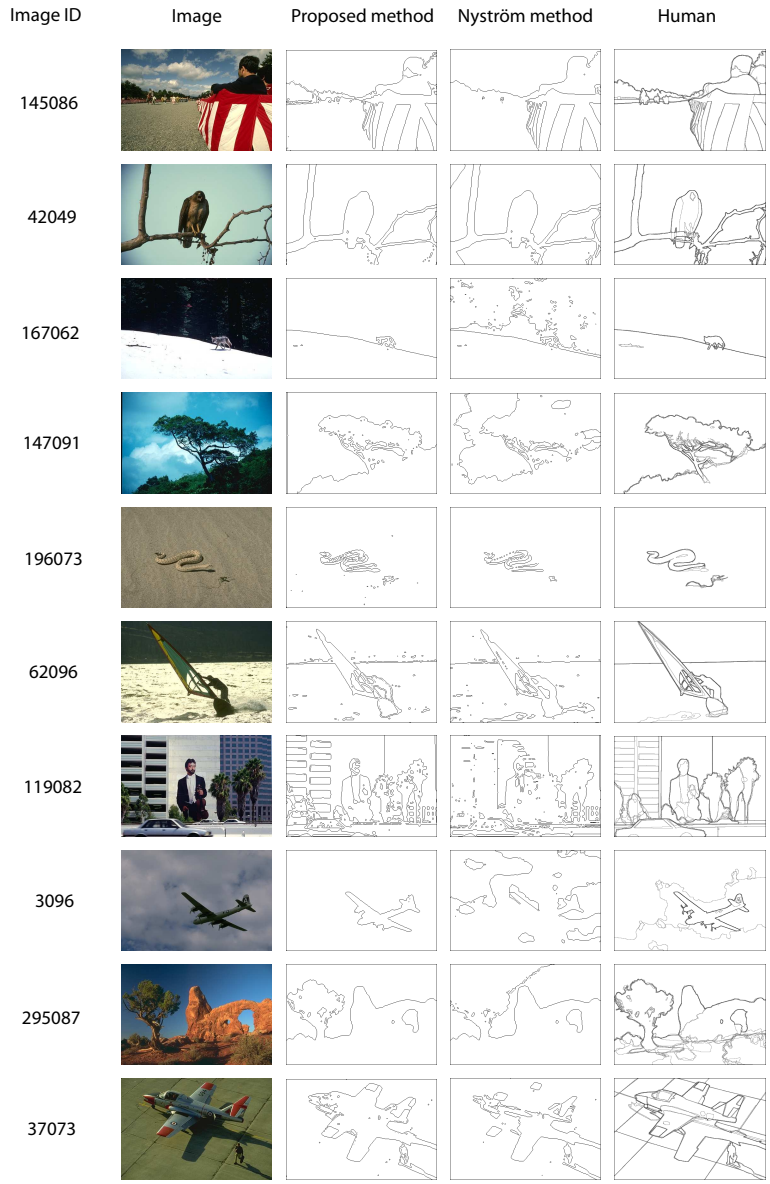


Figure 6.10: Image segmentation results using the proposed method and the Nyström algorithm [46]. The training set consisted of $1,000$ randomly chosen pixel histograms. The proposed approach performs better than [46] with respect to human segmentation. The performance in terms of F-measure of the full test set can be seen in Table 6.1.

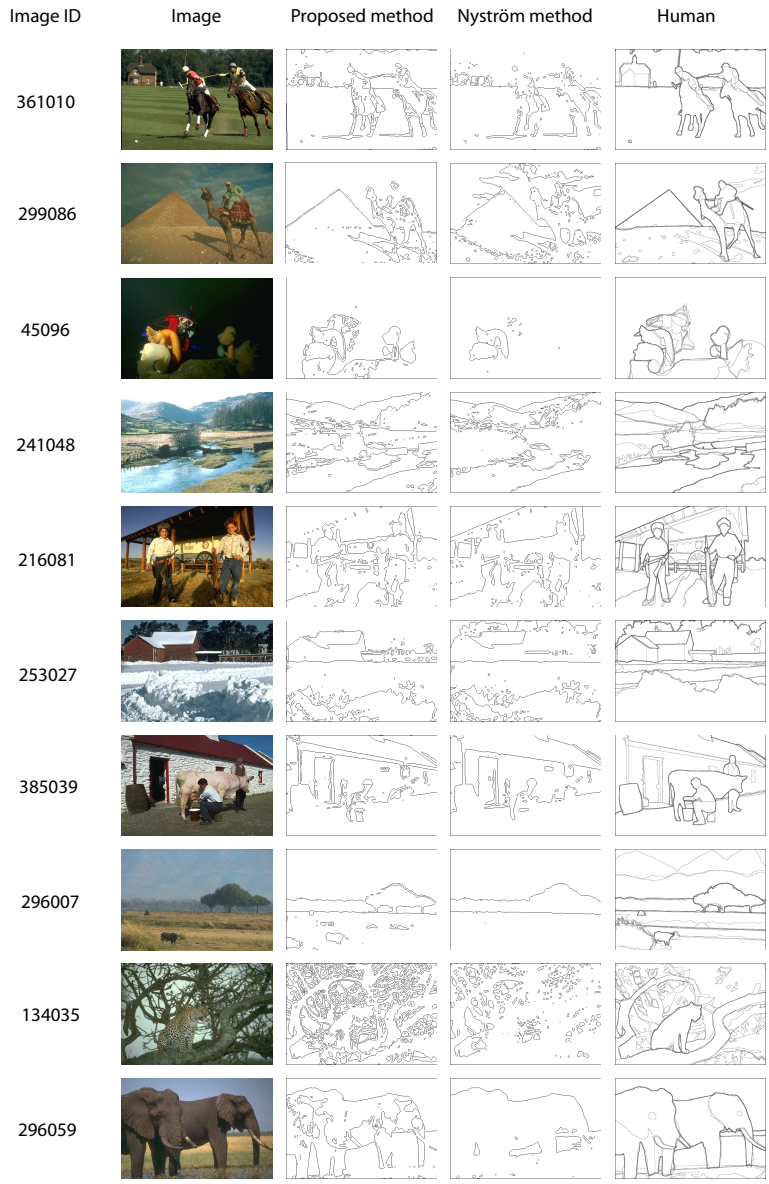


Figure 6.11: Image segmentation results using the proposed method and the Nyström algorithm [46]. The training set consisted of $1,000$ randomly chosen pixel histograms. The proposed approach performs better than [46] with respect to human segmentation. The performance in terms of F-measure of the full test set can be seen in Table 6.1.

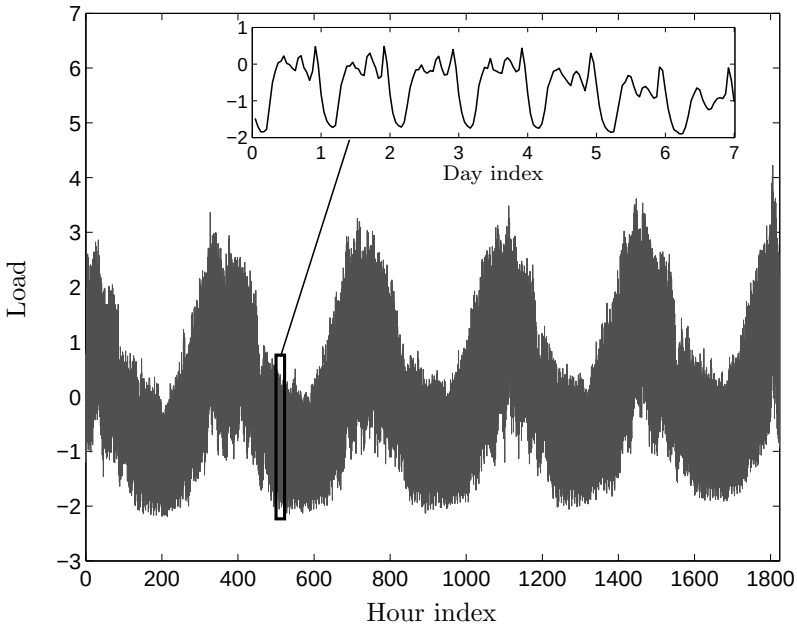


Figure 6.12: Example of time series from the electricity load dataset. Yearly and daily cycles are visible. It is also possible to visualize morning, noon and evening peaks.

6.7.2 Existing Methodology

An approach to cluster the load time series was proposed in [43]. This method computes a Typical Daily Profile (TDP) from the identified parameters in a Periodic Autoregression (PAR) system estimated for each times series, later performing k-means on the TDPs. PAR models allow the autoregression parameters to vary according to cyclic patterns (such as hours, days, years, seasons). These models have been used for forecasting in economical modeling and electricity price modeling. The method proposed in [43] aims at representing the time series as an hourly PAR model of order 48 with additional exogenous variables accounting for monthly and weekly cyclic variations together with temperature effects. The PAR parameters can be obtained by solving an Ordinary Least Squares (OLS) system. A TDP was defined in [43] as a convergence vector of the PAR(48) model after extracting all exogenous information. The main advantage of a TDP

is that it represents a daily load profile independent of the seasonal and calendar variations. In summary, [43] creates a TDP for each time series which implies a dimensionality reduction from $43,824$ to 24. PCA is then applied over the TDP to capture 99% of the information explained by the variance. This corresponds to a further reduction to 9 dimensions. The last step consists of applying k-means over the 9 principal components of the typical daily profile. Figure 6.13 shows the clustering results. Several types of customers can be identified from the results, particularly clusters 1 and 2 which corresponds to *residential* profiles, clusters 6 and 7 which display a *commercial* profile, clusters 4, 5, 8 show a profile typical of street lighting and certain industrial activities taking place during the night.

6.7.3 Proposed Clustering Scheme

The proposed scheme consists of clustering directing the time series in a learning framework with model selection without any premodeling steps. The training scenario consists of 123 time series for training and 122 times series for validation. Each time series is a $43,824$-dimensional vector with zero mean. The training / validation sets were randomized 10 times and the model selection results report average values. We used the BLF for tuning the number of clusters and the RBF kernel parameter σ. The η parameter was set to 0.5 hence, giving equal weight to the linefit and to the balance. The tuning range for the RBF kernel parameter was varied from 50 to $2,000$ in 100 steps. The number of clusters k ranged from 2 to 12. The choice for 50 as the minimum value of the range is motivated by the fact that with lower values the RBF kernel matrix was tending to the identity matrix which has trivial eigenspace with no discriminatory information about the clusters. On the other hand, values greater than $2,000$ led to a almost constant-valued kernel matrix independent of the data. The maximum of the BLF gives the optimal number of clusters and the optimal value for σ. Figure 6.14 shows the model selection results. The BLF detected 7 clusters and the optimal σ value was 200. For visualization purposes, each time series is transformed into a mean daily pattern by averaging the corresponding $1,824$ 24-hour non-overlapping windows. This mean daily pattern is similar to the TDP but differs in the sense that we did not model any periodicity or exogenous variables. Moreover, the mean daily pattern is computed *after* the clustering has been performed and serves only for visualization and clustering interpretation purposes. Figure 6.15 shows the mean daily patterns for the 245 time series using the multiway spectral

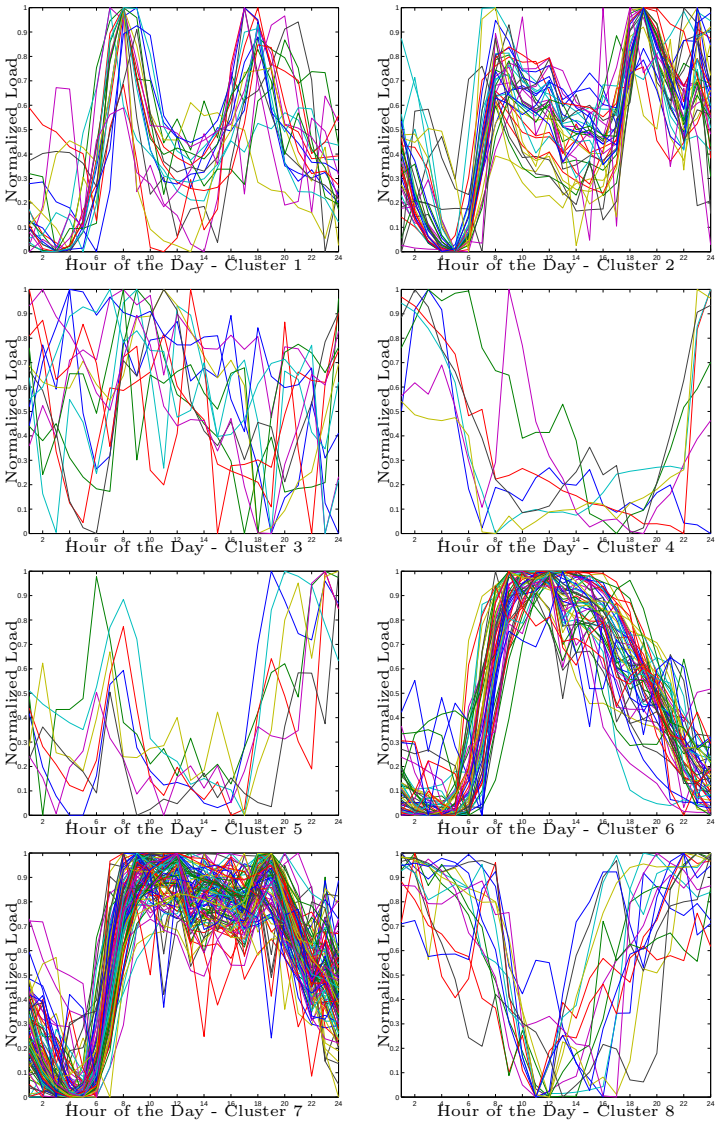


Figure 6.13: Typical daily profile (TDP) clustering results using the method proposed in [43] which corresponds to preprocessing the TDP with PCA to reduce the dimensionality to 9 and then applying k-means.

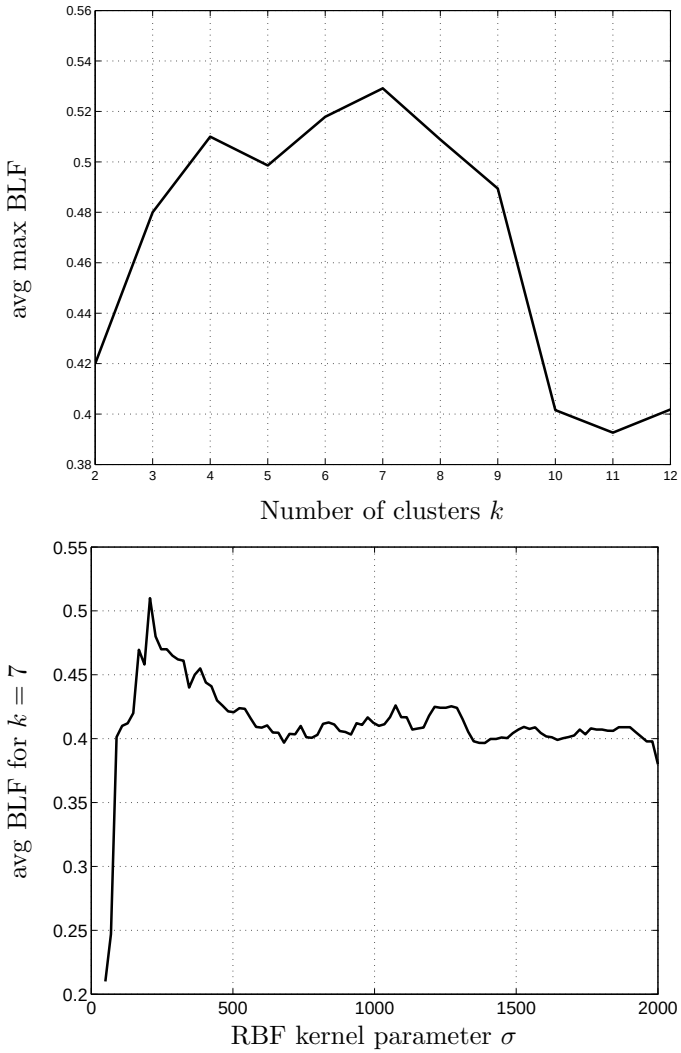


Figure 6.14: Model selection using the BLF. **Top:** Average maximal BLF to determine the number of clusters k. **Bottom:** Average BLF for $k = 7$ to determine the optimal value of the RBF kernel parameter σ. Tuned parameters are $k = 7$ and $\sigma = 200$. The plots show average BLF values after 10 randomizations of the training and validation sets.

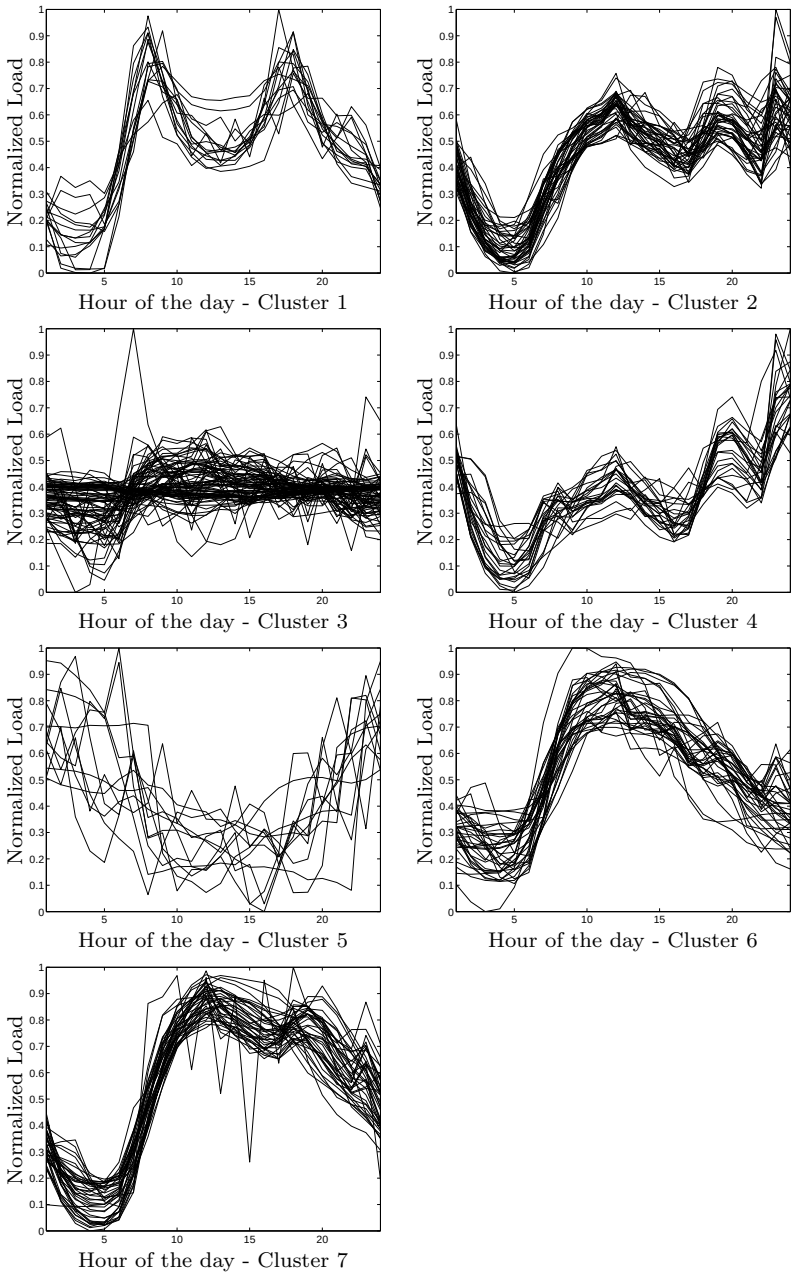


Figure 6.15: Clustering results using the proposed methodology. The results show typical customer profiles while clustering the data directly without pre-processing.

clustering with parameters $k = 7, \sigma = 200$ obtained through model selection with the BLF. The obtained clusters also display typical customer profile information. Namely, cluster 1 suggests *residential* behavior with morning and evening peaks. Cluster 5 characterizes low consumption during the day increasing during the night. Clusters 2, 6 and 7 suggest *commercial* or *business* profiles while cluster 3 and 4 display load behavior without particular associated profile.

6.8 Application: Scientific Journal Clustering

The main motivation in this application is to find groups of scientific journals that are similar with respect to textual and cross-citation information. The 22 Essential Science Indicators (ESI) of Thompson Scientific becomes the main external grouping for comparison. Several internal criteria to measure the quality of the clusters such as the silhouette index [89], the modularity [82, 83] and the normalized mutual information (NMI) [100] were used in this study. For comparison related to external validation, the adjusted Rand index (ARI) [59] was used. The data used in this application correspond to more than six million scientific papers indexed by the Web of Science (WoS) in the period $2002 - 2006$. The type of manuscripts considered is article, letter, note and review. Textual information has been extracted from titles, abstracts and keywords of each paper together with citation information. After several pruning operations in order to discard irrelevant information, the number of journals considered in the dataset is $8,305$. The objective is thus to cluster scientific journals that are similar with respect to textual and cross-citation information. A key aspect consists of integrating text and cross-citations at the kernel level in order to improve existing clustering approaches. The kernel used in this application is the normalized linear kernel which corresponds to the cosine similarity commonly used in information retrieval [92].

6.8.1 Journal By Term/Concept dataset

The textual information was processed using the term frequency - inverse document frequency (TF-IDF) weighting procedure [17]. Terms which occur only in one document and stop words were not considered into the analysis. The Porter stemmer was applied to the remaining terms in the abstract, title

and keyword fields. This processing leads to a term-by-document matrix of around six million papers and $669,860$ term dimensions. The final journal-by-term dataset is a $8,305 \times 669,860$ matrix. Additionally, latent semantic index (LSI) [36] was performed on this dataset to reduce the dimensionality to 200.

6.8.2 Journal cross-citation dataset

A different form of analyzing cluster information at the journal level is through a cross-citation graph. This graph contains aggregated citations between papers forming a journal-by-journal cross-citation matrix. The direction of the citations is not taken into account which leads to an undirected graph and a symmetric cross-citation matrix. The raw cross-citation between journal i and journal j was further normalized by dividing the value by the square root of the product of the total number of citations in journal i and the total number of citations in journal j. Intra-cluster self citations were counted only once. From the normalized journal cross-citation a *second-order* similarity was created by considering each row as a data point and computing the cosine similarity (normalized linear kernel) between each pair of data points. This operation resulted in a square $8,305 \times 8,305$ second-order cross-citation matrix where the ij-th entry represents the similarity between journal i and journal j based on their respective similarities with all other journals.

6.8.3 Data Integration

We consider two types of data integration schemes. The first one consists of concatenating the variables of the textual datasets with the cross-citation dataset leading to a composite concatenated matrix of size $8,305 \times (669,860 + 8,305)$ in the case of the journal-by-term matrix and $8,305 \times (200 + 8,305)$ in case of the journal-by-concept matrix. The second integration scheme consists of integration at the kernel by considering a weighted sum of the textual and cross-citation kernel matrices:

$$\Omega^{\text{integr}} = \tilde{\rho}\Omega^{\text{text}} + (1 - \tilde{\rho})\Omega^{\text{crosscit}}$$

where $0 \leq \tilde{\rho} \leq 1$ is a user-defined integration weight that can be tuned using internal validation measures for cluster distortion.

6.8.4 Experimental Results

Figure 6.16 shows the silhouette curves for the 22 ESI fields in terms of textual and cross-citation information. According to the silhouette index, clusters $2, 4, 7, 19$ and 21 are inconsistent with the data. Thus, it is important to find a consistent grouping of the scientific journals with a better agreement to the available data. Table 6.2 shows the 22 broad science ESI fields. For internal validation of the clusters, we used the following criteria:

- **Silhouette index** [89]. This index ranges from -1 to 1 and measures how similar data points in the same clusters are to each other compared to data points in other clusters. The silhouette plot shows the value of the silhouette index for each data point as a horizontal line. The more the line is to the positive side of the plot, the more coherent this data point is to points in the same cluster. Negative values in the silhouette plot indicates that this data point should be rather assigned to a different cluster. The mean silhouette value (MSV) gives an average measure of the coherence of the groups.
- **Modularity** [82, 83]. The modularity of a corresponding partition quantifies the clustering quality in terms of the number of intra-cluster connections minus the average number of connections in a graph with random connections. Well-formed clusters will have a high modularity value tending to 1.

For external validation, we consider the adjusted Rand index (ARI), the Jaccard similarity coefficient which is an extension of the Jaccard index defined as the cardinality of the intersection divided by the cardinality of the union and the normalized mutual information (NMI) which measures the statistical information shared between a pair of clusterings [100].

In this particular scenario, the model selection consists of determining the optimal integration weight $\tilde{\rho}$. Since we use a normalized linear kernel, there are no kernel parameters to tune. The number of clusters is considered fixed to 22 to be able to compare the proposed approach to other existing methodologies. Figure 6.17 shows the mean silhouette value computed on the eigenvectors of the integrated LSI+cross-citation results for $k = 22$ for varying the integration weight $\tilde{\rho}$. The optimal integration weight according to the silhouette value is $\tilde{\rho} = 0.5$. The spectral clustering results are shown in Table 6.3 for several integration methods together with a hybrid hierarchical clustering algorithm using a linear combination of textual and

Field	Name	Field	Name
1	Agricultural sciences	12	Mathematics
2	Biology and biochemistry	13	Microbiology
3	Chemistry	14	Molecular biology & genetics
4	Clinical medicine	15	Multidisciplinary
5	Computer science	16	Neuroscience & behavior
6	Economics and business	17	Pharmacology & toxicology
7	Engineering	18	Physics
8	Environment/Ecology	19	Plant & animal science
9	Geosciences	20	Psychology / Psychiatry
10	Immunology	21	Social sciences
11	Materials sciences	22	Space science

Table 6.2: The 22 science fields according to the essential science indicators (ESI)

citation information [64]. Figure 6.18 shows Jaccard indices measuring similarities between the integrated spectral clustering method (LSI+cross-citation) with respect to the hybrid algorithm and to the 22 ESI fields. Table 6.4 shows the best 15 terms per cluster according to the proposed LSI+cross-citation algorithm displaying a coherent structure. Figure 6.19 shows a multidimensional scaling (MDS) plot of the cluster centroids with respect to the 22 ESI fields. Some similarities between the two clusterings are clearly visible from the plot, e.g. cluster 6 with the ESI field psychology/psychiatry, cluster 20 with the ESI field chemistry and clusters $2, 8$ with the ESI field economics/business. Silhouette plots based on integrated data are shown in Figure 6.20. This plot shows that the proposed method outperforms the 22 ESI fields (see Figure 6.16) in terms of the silhouette index with respect to the combined textual and cross-citation information.

6.9 Conclusions

A new formulation for multiway spectral clustering with out-of-sample extensions is proposed. This formulation is based on a weighted kernel PCA framework in which the clustering model can be extended to new points. The proposed approach is cast in a constrained optimization framework providing primal-dual insights in a learning scheme. In the case of well formed clusters, the eigenvectors of a modified similarity matrix derived from the data display a special structure. Training data points in the same cluster

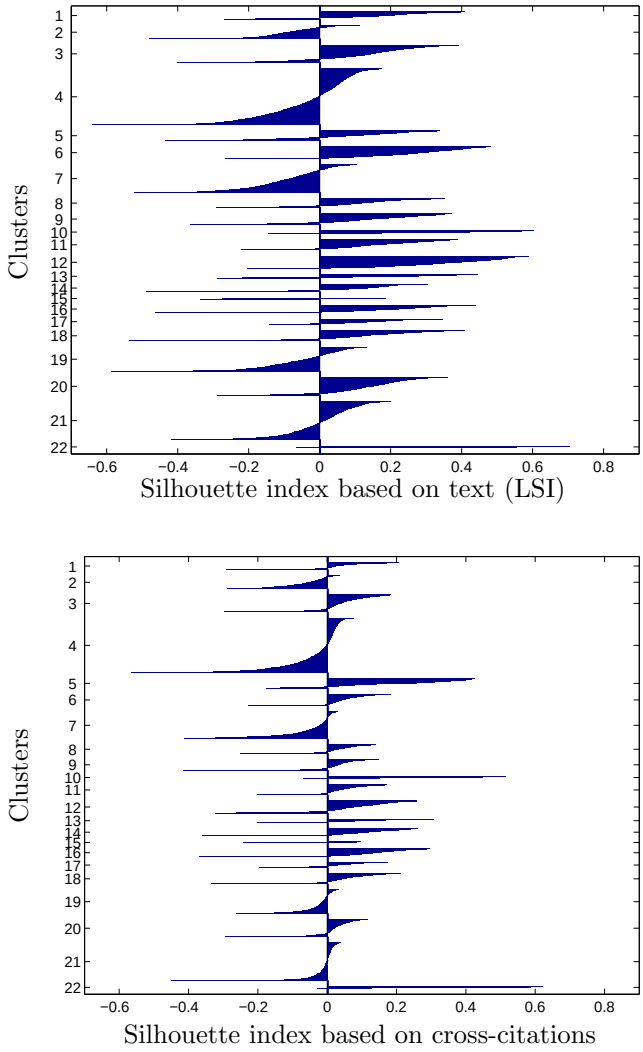


Figure 6.16: Silhouette plots of textual and cross-citation data with respect to the 22 ESI fields clustering. Note that, journals in clusters $2, 4, 7, 19$ and 21 have negative values which means they are inconsistent with the data according to the silhouette index.

	Internal validation			Extern
	MSV textual	MSV cross-cit.	MSV integrated	Modularity ISI 254
22 ESI fields	0.057	0.016	0.063	(**0.526**)[a]
Cross-citations	0.093	0.057	0.189	0.442
Textual (LSI)	0.118	0.035	0.130	0.451
Hierarch. Ward's method $\tilde{\rho} = 0.5$	0.121	0.055	0.190	**0.488**
Concat. Terms+Cross-citations	**0.153**	0.055	0.191	0.457
Concat. LSI+Cross-citations	**0.153**	0.049	0.185	0.446
Integr. Terms+Cross-citations $\tilde{\rho} = 0.5$	0.138	**0.064**	**0.201**	*0.465*
Integr. LSI+Cross-citations $\tilde{\rho} = 0.5$	*0.145*	*0.062*	*0.197*	*0.465*

Table 6.3: Spectral clustering results of several integration methods in terms of mean modularity, adjusted Rand index (ARI) and normalized mutual information (NMI). Fo clustering results are compared with respect to the 22 ESI fields and the ISI 254 subjec value per column is indicated in bold while the second highest value appears in italic. A external validation, the winning algorithms are the proposed integrated spectral clusteri

[a]There is a direct correspondence between the 22 ESI fields and the ISI 254 subject categories which value.

	Best 15 **terms**
Cluster 1	diabet therapi hospit arteri coronari physician renal hypertens mortal syndrom cardiac nurs chronic infect pain
Cluster 2	polit war court reform parti legal gender urban democraci democrat civil capit feder discours economi
Cluster 3	diet milk fat intak cow dietari fed meat nutrit fatti chees vitamin ferment fish dry
Cluster 4	alloi steel crack coat corros fiber concret microstructur thermal weld film deform ceram fatigu shear
Cluster 5	infect hiv vaccin viru immun dog antibodi antigen pathogen il pcr parasit viral bacteri dna
Cluster 6	psycholog cognit mental adolesc emot symptom child anxieti student sexual interview school abus psychiatr gender
Cluster 7	text music polit literari philosophi narr english moral book essai write discours philosoph fiction ethic
Cluster 8	firm price busi trade economi invest capit tax wage financi compani incom custom sector bank
Cluster 9	nonlinear finit asymptot veloc motion stochast elast nois turbul ltd vibrat iter crack vehicl infin
Cluster 10	soil seed forest crop leaf cultivar seedl ha shoot fruit wheat fertil veget germin rice
Cluster 11	soil sediment river sea climat land lake pollut wast fuel wind ocean atmospher ic emiss
Cluster 12	algebra theorem manifold let finit infin polynomi invari omega singular inequ compact lambda graph conjectur
Cluster 13	pain surgeri injuri lesion muscl bone brain ey surgic nerv mri ct syndrom fractur motor
Cluster 14	rock basin fault sediment miner ma tecton isotop mantl volcan metamorph seismic sea magma faci
Cluster 15	web graph fuzzi logic queri schedul semant robot machin video wireless neural node internet traffic
Cluster 16	student school teacher teach classroom instruct skill academ curriculum literaci learner colleg write profession disabl
Cluster 17	habitat genu fish sp forest predat egg nest larva reproduct taxa bird season prei nov
Cluster 18	star galaxi solar quantum neutrino orbit quark gravit cosmolog decai nucleon emiss radio nuclei relativist
Cluster 19	film laser crystal quantum atom ion beam si nm dope thermal spin silicon glass scatter
Cluster 20	polym catalyst ion bond crystal solvent ligand hydrogen nmr molecul atom polymer poli aqueou adsorpt
Cluster 21	receptor rat dna neuron mice enzym genom transcript brain mutat peptid kinas inhibitor metabol cancer
Cluster 22	cancer tumor carcinoma breast therapi prostat malign chemotherapi biopsi surgeri lesion lymphoma pancreat recurr resect

Table 6.4: Best 15 terms per cluster according to the integrated results (LSI+cross-citations) with $\tilde{\rho} = 0.5$ displaying a coherent structure.

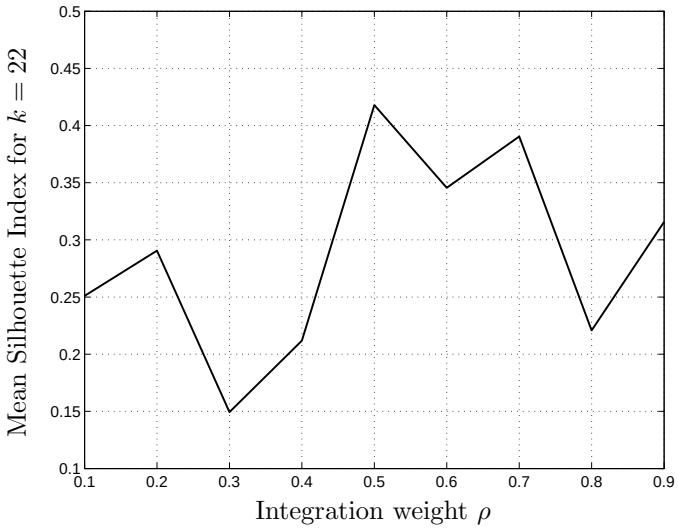


Figure 6.17: Parameter selection - Tuning ρ for the integrated results (LSI+cross-citation). The optimal integration weight ρ is 0.5 according to the mean silhouette value measured on the eigenvectors.

are mapped into a single point in the space spanned by the eigenvectors. In the case of validation data, points in the same cluster are collinear in the space spanned by the projected variables. This structure is exploited by the proposed model selection criterion to obtain useful model parameters leading to visually appealing and interpretable clusters. Toy examples illustrate the proposed clustering model and model selection criterion. Three application studies are also discussed showing the applicability of the method to image segmentation, power load time series clustering and scientific journal clustering.

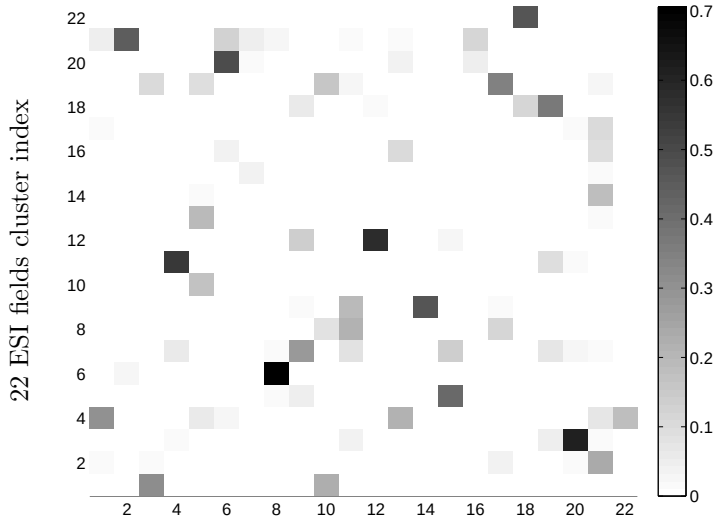


Figure 6.18: Visual Jaccard Index. Agreement of the proposed integrated scheme with respect to the 22 ESI clustering and the Ward's method with linearly combined distances. **Top:** Clusters $2, 4, 8, 12, 20$ of the proposed approach are very similar to clusters $21, 11, 6, 12, 3$ in the 22 ESI indicators respectively. **Bottom:** Clusters $6, 15, 16, 17, 19, 20$ of the proposed clustering agree with clusters $21, 18, 11, 19, 20, 4$ of Ward's method respectively.

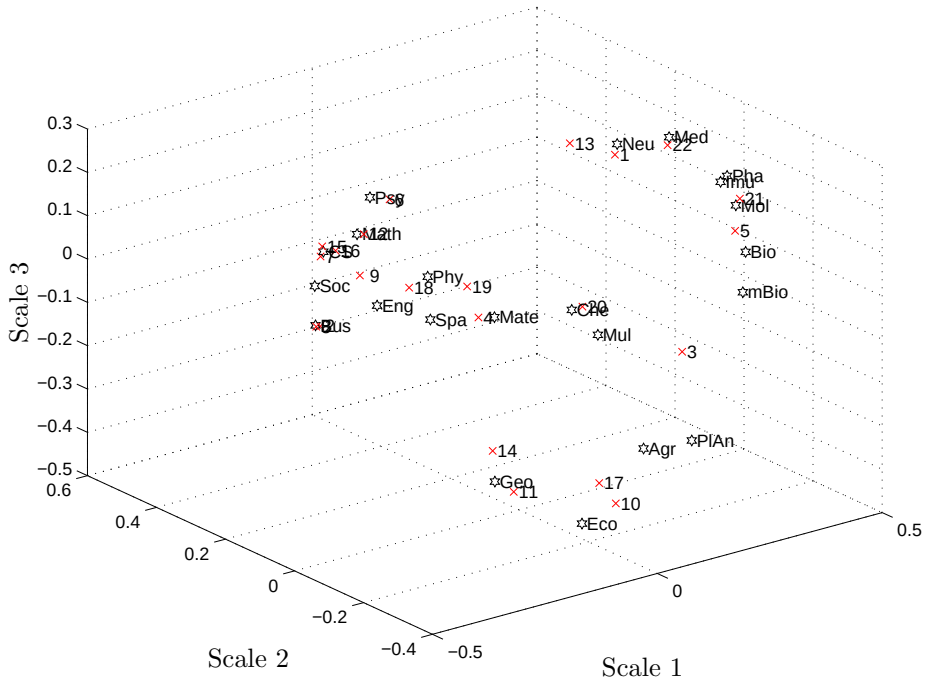


Figure 6.19: Multidimensional scaling (MDS) map of the cluster centroids of the 22 ESI fields with respect to the proposed clustering scheme (LSI+cross-citations). A cluster centroid is defined as the linear combination of all documents in the cluster. The crosses indicate the cluster centroids for the proposed method. Similarities between the different clusterings are e.g. cluster 6 with the ESI field psychology/psychiatry, cluster 20 with the ESI field chemistry and clusters $2, 8$ with the ESI field economics/business.

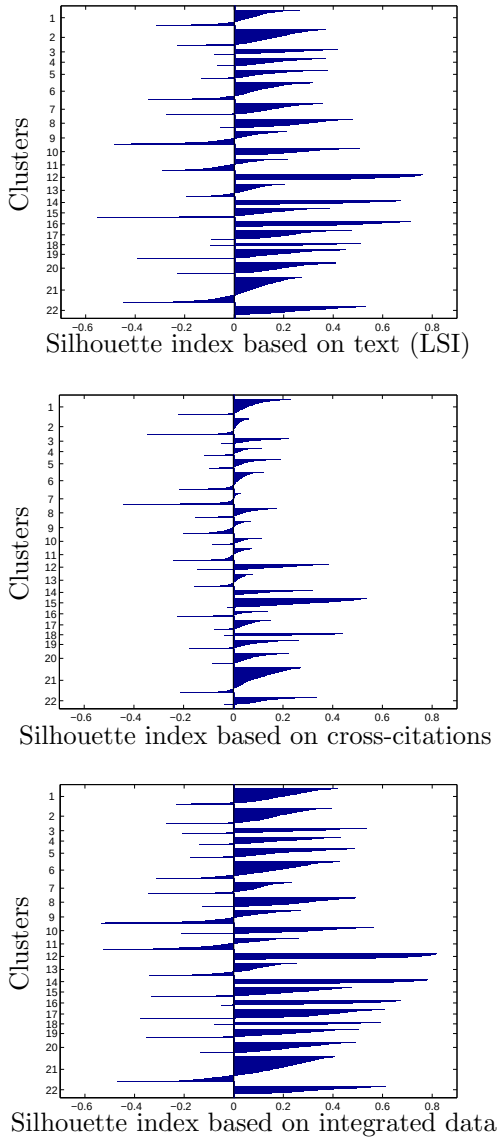


Figure 6.20: Spectral clustering results - Silhouette plots. **Top:** Silhouette index with respect to textual information (LSI). The mean silhouette value is 0.145. **Center:** Silhouette index with respect to cross-citations. The mean silhouette value is 0.062. **Bottom:** Silhouette index with respect to integrated information. Mean silhouette value is 0.197. Note the improvements compared to Figure 6.16 in terms of cluster quality according to the silhouette index.

Chapter 7

Sparse Spectral Clustering

This chapter presents an extension of the multiway spectral clustering method introduced in chapter 6. For large scale problems, the cost of storing and computing an eigendecomposition of the matrices involved in spectral clustering methods can be prohibitive. Another issue relies on the fact that the cluster indicators are obtained through dense kernel expansions in which every training point contributes. Thus, approximation techniques and low-rank decompositions are needed in order to deal with problems with a large number of data points. The incomplete Cholesky decomposition [48, 56] used previously in chapter 5 to reduce the computational burden of the kernel regularized correlation measure for ICA can also be applied in the context of spectral clustering. In this chapter, a sparse kernel model for spectral clustering based on the incomplete Cholesky decomposition is proposed. This sparse kernel model aims at approximating the eigenvectors of the modified affinity matrix by solving a smaller eigenvalue problem. The cluster indicators for out-of-sample points are also approximated via a reduced set of points obtained automatically from the incomplete Cholesky decomposition. This chapter is organized as follows. Section 7.1 contains the approximation of the eigenvectors together with the reduced set method. In Section 7.2, an algorithm for sparse spectral clustering is proposed and Section 7.3 shows experimental results with large-scale toy data.

7.1 Sparse model via the incomplete Cholesky decomposition

7.1.1 Approximated Eigenvectors

Consider the eigenvalue problem of the multiway spectral clustering method described in chapter 6:

$$D^{-1}M_D\Omega\alpha^{(l)} = \lambda_l\alpha^{(l)} \tag{7.1}$$

where the cluster indicators of a given data point x can be obtained from the projections:

$$q^{(l)}(x) = \operatorname{sign}\big({w^{(l)}}^T\varphi(x) + b_l\big) = \operatorname{sign}\bigg(\sum_{i=1}^{N}\alpha_i^{(l)}K(x,x_i) + b_l\bigg), \tag{7.2}$$

$l = 1,\ldots,k-1$. The incomplete Cholesky decomposition of the kernel matrix Ω gives:

$$\Omega \approx GG^T, \tag{7.3}$$

where $G \in \mathbb{R}^{N\times R}$ and $R \leq N$. Using the singular value decomposition of G, it is possible to write:

$$G = U\Lambda V^T \tag{7.4}$$

where $U \in \mathbb{R}^{N\times R}$ is the matrix of left singular vectors, $\Lambda \in \mathbb{R}^{R\times R}$ is the diagonal matrix of singular values and $V \in \mathbb{R}^{N\times R}$ is the matrix of right singular vectors. Substituting (7.4) into (7.3) gives $\Omega \approx U\Lambda^2U^T$. Hence, the approximation of (7.1) can be written as:

$$D^{-1}M_DU\Lambda^2U^T\alpha^{(l)} = \lambda_l\alpha^{(l)}, l = 1,\ldots,k-1,$$

premultiplying both sides of the eigenequation by U^T gives:

$$\begin{aligned} U^TD^{-1}M_DU\Lambda^2U^T\alpha^{(l)} &= \lambda_lU^T\alpha^{(l)} \\ U^TD^{-1}M_DU\Lambda^2\tilde{\alpha}^{(l)} &= \lambda_l\tilde{\alpha}^{(l)} \end{aligned} \tag{7.5}$$

where $\tilde{\alpha}^{(l)} = U^T\alpha^{(l)}$ and $l = 1,\ldots,k-1$. Note that the matrix $U^TD^{-1}M_DU\Lambda^2$ in (7.5) is $R \times R$ which can be much smaller than the full $N \times N$ matrix in (7.1) provided that the eigenspectrum of the kernel matrix decays fast. This approximation scheme can also be applied to kernel PCA, other spectral clustering techniques and in general to any eigenvalue-based kernel method.

7.1.2 Reduced Set Method

So far, the size of the matrix involved in the eigendecomposition of (7.1) has been reduced from $N \times N$ to $R \times R$. However, note also that the score variables used to determine the cluster indicators (7.2) are expressed in terms of dense kernel expansions. This comes from the KKT conditions for optimality (6.5) which expresses the primal projection vectors $w^{(l)}$ as linear combinations of the whole set of mapped training points:

$$w^{(l)} = \sum_{i=1}^{N} \alpha_i^{(l)} \varphi(x_i) = \Phi^T \alpha^{(l)}.$$

The objective of a reduced set method is to approximate $w^{(l)}$ by

$$\tilde{w}^{(l)} = \sum_{r=1}^{N_s} \beta_m^{(l)} \varphi(\tilde{x}_r) = \Psi^T \beta^{(l)}$$

where $l = 1, \ldots, k-1$, $\Psi = [\varphi(\tilde{x}_1^T); \ldots; \varphi(\tilde{x}_{N_s}^T)]$, $\Psi \in \mathbb{R}^{N_s \times n_h}$, $\beta = [\beta_1, \ldots, \beta_{N_s}]$ is the vector of reduced set coefficients, $\{\tilde{x}_r\}_{r=1}^{N_s}$ is the reduced set and $N_s < N$. A similar scheme was used in [22] to approximate support vector decision rules for classification and in [95] to sparsify kernel PCA expansions. The reduced set problem consists of two subproblems: first, finding the reduced set coefficients β and second finding the reduced set $\{\tilde{x}_r\}_{r=1}^{N_s}$. One approach to solve the first subproblem consists of minimizing the squared two-norm of the residuals with respect to $\beta^{(l)}$:

$$\min_{\beta^{(l)}} ||w^{(l)} - \tilde{w}^{(l)}||_2^2 = \min_{\beta^{(l)}} w^{(l)^T} w^{(l)} - 2\tilde{w}^{(l)^T} w + \tilde{w}^{(l)^T} \tilde{w}^{(l)}.$$

Using $w^{(l)} = \Phi^T \alpha^{(l)}$ and $\tilde{w}^{(l)} = \Psi^T \beta^{(l)}$ gives:

$$\min_{\beta^{(l)}} \alpha^{(l)^T} \Phi \Phi^T \alpha^{(l)} - 2\beta^{(l)^T} \Psi \Phi^T \alpha^{(l)} + \beta^{(l)^T} \Psi \Psi^T \beta^{(l)}$$

and applying the kernel trick leads to:

$$\min_{\beta^{(l)}} \alpha^{(l)^T} \Omega \alpha^{(l)} - 2\beta^{(l)^T} \Omega^{\Psi\Phi} \alpha^{(l)} + \beta^{(l)^T} \Omega^{\Psi\Psi} \beta^{(l)},$$

where $l = 1, \ldots, k-1$, $\Omega^{\Psi\Phi}$ is a $N_s \times N$ matrix with ri-th entry $\Omega_{ri}^{\Psi\Phi} = K(\tilde{x}_r, x_i), r = 1, \ldots, N_s, i = 1, \ldots, N$ and $\Omega^{\Psi\Psi}$ is a $N_s \times N_s$ matrix with rn-th entry $\Omega_{rn}^{\Psi\Psi} = K(\tilde{x}_r, \tilde{x}_n), r, n = 1, \ldots, N_s$.

The first-order optimality condition

$$\frac{\partial ||w^{(l)} - \tilde{w}^{(l)}||_2^2}{\partial \beta^{(l)}} = 0$$

gives:

$$\Omega_{\Psi\Psi}\beta^{(l)} = \Omega_{\Psi\Phi}\alpha^{(l)} \tag{7.6}$$

which is a linear system to be solved in $\beta^{(l)}, l = 1, \ldots, k-1$. A different approach for sparse expansions was also mentioned in [95]. This approach uses L_1 shrinkage penalizers and the reduced set coefficients can be obtained by solving a QP problem.

The second subproblem can be solved by means of the incomplete Cholesky decomposition. This factorization also gives information about the data points used in the approximation of Ω (also called pivots). The pivots represent the subset $\{\tilde{x}_r\}_{r=1}^{N_s} \subset \mathcal{D}$ for which the mapped points $\varphi(\tilde{x}_r)$ are linearly independent. In this case, the matrix $\Omega_{\Psi\Psi}$ is full rank and $N_s = R$.

The approximated score variables for training data become:

$$e_i^{(l)} \approx \sum_{r=1}^{R} \beta_r^{(l)} K(x_i, \tilde{x}_r), i = 1, \ldots, N \tag{7.7}$$

and for test data:

$$z_m^{(l)} \approx \sum_{r=1}^{R} \beta_r^{(l)} K(x_m^{\text{test}}, \tilde{x}_r), m = 1, \ldots, N_{\text{test}}. \tag{7.8}$$

7.2 Algorithm

An implementation of the proposed sparse approach is shown in Algorithm 6. Note that the matrix $D^{-1}M_D \in \mathbb{R}^{N \times N}$ in (7.1) does not need to be stored. The ij-th entry of $D^{-1}M_D$ can be expressed in terms of the graph degrees:

$$[D^{-1}M_D]_{ij} = \begin{cases} -\frac{1_N^T D^{-1} 1_N}{(\deg_i \deg_j)} & \text{if } j \neq i \\ \frac{1}{\deg_i} - \frac{1_N^T D^{-1} 1_N}{\deg_i^2} & \text{if } j = i \end{cases}$$

$i, j = 1, \ldots, N$. Thus, the rows of $D^{-1}M_D$ can be computed on demand to build the $R \times R$ matrix $U^T D^{-1} M_D U \Lambda^2$.

Algorithm 6 Sparse Kernel Model for Spectral Clustering Using the Incomplete Cholesky Decomposition

Input: Training set $\{x_i\}_{i=1}^N$, positive definite kernel function $K(x_i, x_j)$, number of clusters k, Cholesky error tolerance η, test set $\{x_m^{\text{test}}\}_{m=1}^{N_{\text{test}}}$.
Output: Partition $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$, approximated eigenvectors α, cluster codeset $\mathcal{C} = \{c_p\}_{p=1}^k, c_p \in \{-1, 1\}^{k-1}$, reduced set $\mathcal{R}$.

1: Compute the incomplete Cholesky factor $G \in \mathbb{R}^{N \times R}$ of the kernel matrix Ω such that $||\Omega - GG^T||_2^2 \leq \eta$ and obtain the reduced set $\mathcal{R} = \{\tilde{x}_m\}_{m=1}^R$ from the pivots.
2: Compute the left singular vectors U and the singular values Λ of G.
3: Compute the eigenvectors $\tilde{\alpha}^{(l)}$ corresponding to the $k-1$ largest eigenvalues of $U^T D^{-1} M_D U \Lambda^2$.
4: Obtain the approximated eigenvectors via:

$$\alpha^{(l)} = U\tilde{\alpha}^{(l)}, l = 1, \ldots, k-1.$$

5: Let $\text{sign}(\alpha_i) \in \mathbb{R}^{k-1}$ be the encoding vector for the training data point x_i, $i = 1, \ldots, N$.
6: Count the occurrences of the different encodings and find the k encodings with most occurrences. Let the codebook be formed by these k encodings: $\mathcal{C} = \{c_p\}_{p=1}^k, c_p \in \{-1, 1\}^{k-1}$
7: For all i, assign x_i to $\mathcal{A}_{p^\star}$ where $p^\star = \arg\min_p d_H(\text{sign}(\alpha_i), c_p)$ and $d_H(\cdot, \cdot)$ is the Hamming distance.
8: Let $\text{sign}(z_m) \in \mathbb{R}^{k-1}$ be the encoding vector for the test data point $x_m^{\text{test}}, m = 1, \ldots, N^{\text{test}}, l = 1, \ldots, k-1$.
9: For all m, assign x_m^{test} to $\mathcal{A}_{p^\star}$ where

$$p^\star = \arg\min_p d_H(\text{sign}(z_m), c_p)$$

7.3 Empirical Results

In this section we report empirical results. The assessment of the clustering for the toy examples is done using the adjusted Rand index [59]. This external clustering performance measures the degree of mismatch between some known cluster indicators and the results of a clustering algorithm. The adjusted Rand index ranges from 0 in the case of complete mismatch to 1 in the case of a perfect match. All experiments were performed on an Intel

Core Duo 2.0 GHz, 2 GB RAM in Matlab 7.5. We used the incomplete Cholesky routine described in [14] which computes efficiently the Cholesky factor.

7.3.1 Three Gaussian clouds in 3D

This experiment consists of three Gaussian clouds in a 3D space. The number of training points is $6,000$. The RBF kernel parameter σ^2 is fixed to 15. The Cholesky error tolerance η is fixed within the range $10^{-5} \leq \eta \leq 10^{-1}$. The adjusted Rand index between the approximated cluster indicators and the true labels is equal to 1 for the whole range. For $\eta > 0.5$, the adjusted Rand index starts to decrease (i.e. some points are assigned to the wrong Gaussian cloud). The computation times with respect to η are shown in Figure 7.1. The dashed line corresponds to the *full* approach which is computing the eigendecomposition of (7.1) directly without approximations. The solid line shows the time spent to solve the proposed approach. Note that for $\eta > 10^{-3}$ the reduction in time is more than one order of magnitude. Figure 7.2 shows the training points in gray and the reduced set points in black. The obtained reduced set size is $R = 78$ and the points are distributed on the outskirts of the clouds.

7.3.2 Large-scale intertwined spirals problem

This experiment consists of two intertwined spirals in 2D. The RBF kernel parameter was fixed to $\sigma^2 = 0.5$ and the Cholesky error tolerance to $\eta = 0.5$. The computation times of the proposed approach with respect to the number of data points are shown in Figure 7.3 and reduced set sizes in Table 7.1. The proposed sparse approach takes about 8 hours to compute the approximated eigenvectors of the dataset when $N = 100,000$ and the obtained reduced set size is 261 points (0.21%). Figure 7.4 shows the clustering results for the given σ^2 and $N = 40,000$. The two spiral clusters are correctly detected. We did not perform comparisons with the full approach because in that case the matrix $D^{-1}M_D\Omega$ does not fit into memory.

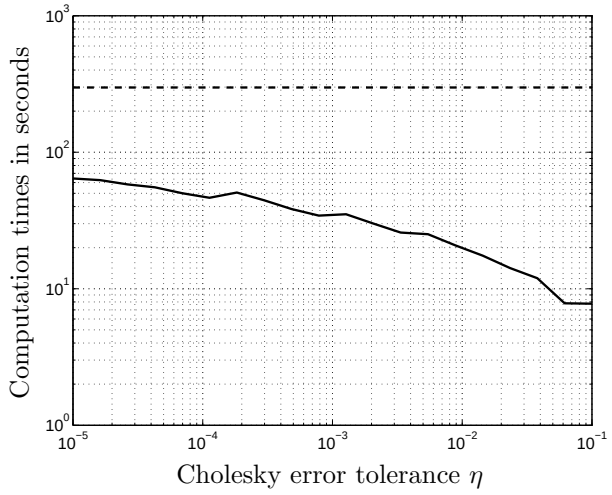


Figure 7.1: Three Gaussian clouds experiment. Computation times in seconds of the sparse approach (solid line) with respect to the Cholesky error tolerance η. The dashed line corresponds to the full approach. The reduction in time is more than one order of magnitude for $\eta > 10^{-3}$.

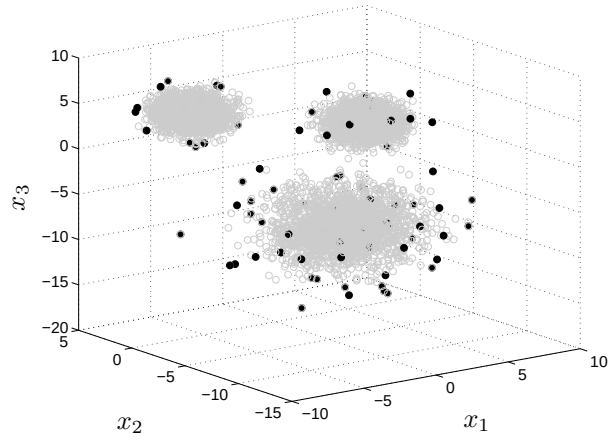


Figure 7.2: Three Gaussian clouds experiment. The reduced set ($R = 78$) is depicted in black and the training points ($N = 6,000$) in gray. The reduced set points are mainly distributed on the borders of the clouds.

7.3.3 Image Segmentation

For this experiment we use a 400×280 pixels image from the Berkeley image dataset [75]. The RBF kernel parameter $\sigma^2 = 0.15$ was tuned using

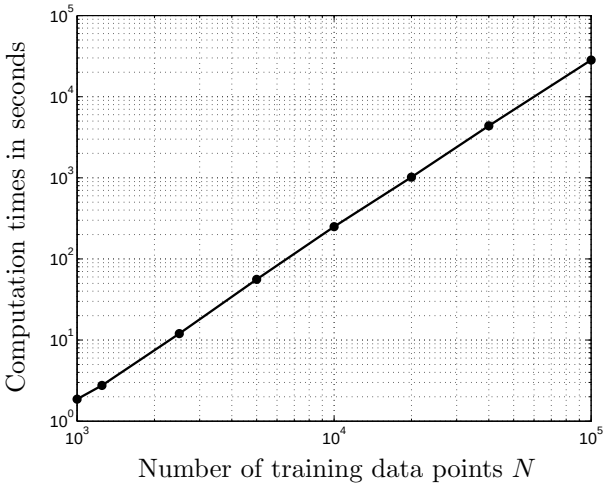


Figure 7.3: Large-scale intertwined spirals. Computation times of the proposed approach with respect to N with η fixed to 0.5. Note that for the given range of N, the full approach does not fit into memory.

Training set size N	Reduced set size R
$1,000$	168 (83.2%)
$5,000$	210 (95.8%)
$10,000$	223 (97.8%)
$50,000$	242 (99.5%)
$100,000$	261 (99.7%)

Table 7.1: Large-scale intertwined spirals. Reduced set size R with respect to N. The percentages indicate the degree of sparseness. The Cholesky error tolerance η is fixed to 0.5.

the BLF. The training set consists of a 200×140 pixels image therefore $N = 28,000$. The validation set is used to find the kernel parameters and the cluster indicators of the test sets are inferred using the out-of-sample extension. The Cholesky error tolerance η is fixed to 10^{-1} with an obtained reduced set of 124 pixels (99.6% sparseness). The total computation time was 120 seconds. Figure 7.5 shows the segmentation results. The clusters found are visually appealing.

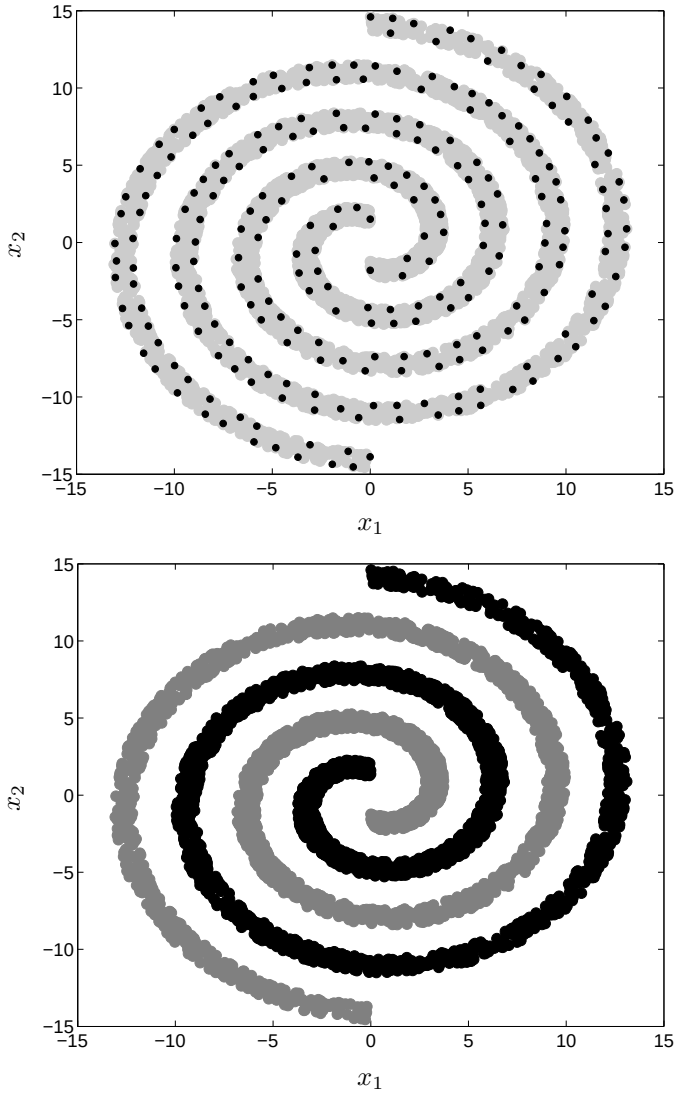


Figure 7.4: Large-scale intertwined spirals with $N = 40,000$. **Top:** Training set shown in gray. The reduced set points (shown in black) are distributed on the borders of the spirals. **Bottom:** Clustering results. Cluster 1 is shown in gray, cluster 2 in black. The proposed sparse approach correctly indicates the two spiral clusters.

7.4 Conclusions

A new sparse kernel model for computing efficiently and approximating the solutions of spectral clustering problems is proposed. This method is

based on the incomplete Cholesky decomposition and also allows the efficient computation of cluster indicators for out-of-sample points via a reduced set technique. The formulation is cast in an optimization framework where the eigenvectors of a modified affinity matrix are the dual solutions of a primal problem. The framework allows extending the clustering model to out-of-sample points. The proposed approximations can handle large-scale data with a reduced computational complexity and with similar performance compared to the original method without approximations. With the out-of-sample extension the clustering model can be trained, validated and tested in a learning frawework. Simulations with large-scale toy datasets show the applicability of the proposed method.

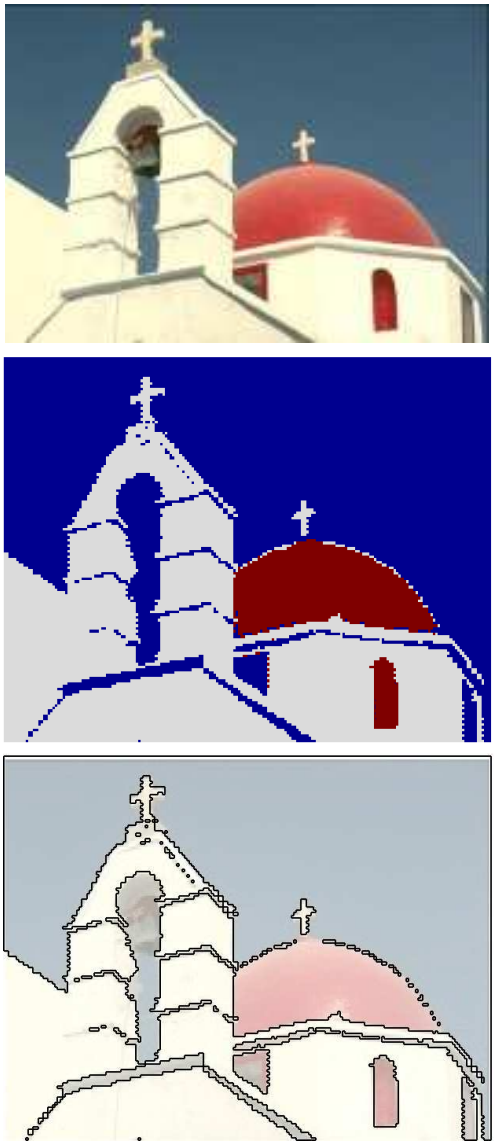


Figure 7.5: Image segmentation. *Top:* Original 400×280 image. Training, validation and test sets are obtained from this image using subsampling. *Center:* Obtained clusters shown in different colors. *Bottom:* Sobel edge detector on the obtained clusters. The original image is shown faded. The Cholesky error tolerance η is fixed to 10^{-1} with an obtained reduced set of 124 pixels (99.6% sparseness).

Chapter 8

Spectral Clustering with Pairwise Constraints

This chapter presents another extension of spectral clustering with out-of-sample extensions towards a form of semi-supervised learning. Incorporating prior knowledge into clustering algorithms has gained significant interest in the machine learning and data mining communities. The type of prior knowledge used is usually in the form of constraints on the cluster assignment. These constraints may come from experts, known or perceived similarity and common sense. Pairwise constraints are cluster assignments restrictions and can be must-link or cannot-link. Must-link constraints specify that two data points should belong to the same cluster while cannot-link constraints restrict that two data points should be in different clusters. Incorporating pairwise constraints into the clustering algorithm have been discussed for centroid-based algorithms such as k-means [19, 112]. However, few methods exist for constrained spectral clustering [66, 72]. The method described in [66] uses an RBF affinity function replacing directly the entries corresponding to the pairwise constraints. This ad-hoc modification does not work well in practice and may cause that the kernel matrix becomes negative-definite. The method in [72] proposes a modification of the kernel matrix arising from a Gaussian process interpretation but with no extensions to new points. In this chapter, additional constraints

characterizing pairwise constraints are added to the primal problem of the weighted kernel PCA framework for spectral clustering. The constraints lead to rank-1 downdates of the kernel matrix in the dual problem. This chapter is organized as follows. Section 8.1 discusses a hard constrained formulation. Section 8.2 contains a regularized soft constrained formulation together with an algorithm. In Section 8.3, a modification of the BLF criterion is introduced to measure the constraint fit and in Section 8.4, experimental results are reported.

8.1 Hard Pairwise Constraint

In general, a pairwise constraint of the type must-link should lead to mapping the pair of points close to each other in the score variables space. In this way, the data point will have the same codeword thus belonging to the same cluster. On the other hand, a *cannot-link* pairwise constraint should map the points far from each other in the score variables space such that they have different codewords.

Given training data $\mathcal{D} = \{x_i\}_{i=1}^N$ and a pair of data points $x_a, x_d \in \mathcal{D}, a \neq d$, a pairwise constraint between x_a and x_d in the $\mathbb{R}^{k-1}$ score variables space can be represented by the following set of $k-1$ equations:

$$w^{(l)^T}\varphi(x_a) = cw^{(l)^T}\varphi(x_d), l = 1, \ldots, k-1, \tag{8.1}$$

where $c \in \{-1, 1\}$ defines the relationship between x_a and x_d in the score variables space and takes the value of either -1 or 1 for cannot-link and must-link constraints respectively. Note that, (8.1) represents a *hard* constraint requiring that x_a and x_d should be mapped exactly to the same point.

Introducing (8.1) into the multiway spectral clustering method of chapter 6 leads to the following primal problem:

$$\min_{w^{(l)}, e^{(l)}, b_l} \frac{1}{2N}\sum_{l=1}^{k-1}\gamma_l e^{(l)^T} D^{-1} e^{(l)} - \frac{1}{2}\sum_{l=1}^{k-1} w^{(l)^T} w^{(l)} \tag{8.2}$$

$$\text{such that } \begin{cases} e^{(1)} = \Phi w^{(1)} + b_1 1_N \\ \vdots \\ e^{(k-1)} = \Phi w^{(k-1)} + b_{k-1} 1_N \\ {w^{(1)}}^T \varphi(x_a) = c {w^{(1)}}^T \varphi(x_d) \\ \vdots \\ {w^{(k-1)}}^T \varphi(x_a) = c {w^{(k-1)}}^T \varphi(x_d). \end{cases}$$

The dual solutions are formalized in the following Lemma.

Lemma 8.1. ***Alzate & Suykens [3]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$, regularization constants $\gamma_l \in \mathbb{R}^+, l = 1, \ldots, n_e$, the inverse of the graph degree matrix D^{-1} which is diagonal with positive elements and a constant $c \in \{-1, 1\}$ representing a pairwise constraint, then the KKT optimality conditions of the Lagrangian of* (8.2) *are satisfied by the following eigenvalue problem:*

$$\lambda_l \alpha^{(l)} = D^{-1} M_D \Omega^\star \alpha^{(l)} \tag{8.3}$$

where $\lambda_l = N/\gamma_l$, $\Omega^\star = \Omega - \frac{1}{k_1}(\Omega(:, a) - c\Omega(:, d))(\Omega(:, a) - c\Omega(:, d))^T$, Ω is the kernel matrix, $\Omega(:, i) \in \mathbb{R}^N$ is the i-th column of Ω, $k_1 = K(x_a, x_a) - 2cK(x_a, x_d) + c^2 K(x_d, x_d)$, $M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N} 1_N 1_N^T D^{-1}$, and $l = 1, \ldots, k-1$.

Proof: The Lagrangian of (8.2) is:

$$\mathcal{L}(w^{(l)}, e^{(l)}, b_l; \alpha^{(l)}, \beta_l) = \frac{1}{2N} \sum_{l=1}^{k-1} \gamma_l {e^{(l)}}^T D^{-1} e^{(l)} - \frac{1}{2} \sum_{l=1}^{k-1} {w^{(l)}}^T w^{(l)} - \sum_{l=1}^{k-1} {\alpha^{(l)}}^T (e^{(l)} - \Phi w^{(l)} - b_l 1_N) - \sum_{l=1}^{k-1} \beta_l {w^{(l)}}^T (\varphi(x_a) - c\varphi(x_d)),$$

where $\alpha^{(l)} \in \mathbb{R}^N, \beta_l \in \mathbb{R}$ are Lagrange multipliers and $l = 1, \ldots, k-1$. The

KKT conditions for optimality are:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w^{(l)}} = 0 \rightarrow w^{(l)} = \Phi^T \alpha^{(l)} - \beta_l(\varphi(x_a) - c\varphi(x_d)) \\ \dfrac{\partial \mathcal{L}}{\partial e^{(l)}} = 0 \rightarrow \alpha^{(l)} = (\gamma_l/N) D^{-1} e^{(l)} \\ \dfrac{\partial \mathcal{L}}{\partial b_l} = 0 \rightarrow 1_N^T \alpha^{(l)} = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \alpha^{(l)}} = 0 \rightarrow e^{(l)} = \Phi w^{(l)} + b_l 1_N \\ \dfrac{\partial \mathcal{L}}{\partial \beta_l} = 0 \rightarrow {w^{(l)}}^T (\varphi(x_a) - c\varphi(x_d)) = 0, \end{cases}$$

for $l = 1, \ldots, k-1$. Eliminating the primal variables $w^{(l)}, e^{(l)}, b_l$ and using Mercer's theorem leads to the eigenvalue problem in (8.3). ■

Remark 8.1 (Equivalent kernel - hard pairwise constraint)**.** *Note that the modified kernel matrix $\Omega^\star$ corresponds to a rank-1 downdate of the original kernel matrix Ω. This modification changes the metric according to the pairwise constraints and the equivalent kernel function can be written as:*

$$\begin{aligned} K^\star_{eq}(x_i, x_j) = & K(x_i, x_j) - \\ & \frac{1}{k_1}\big(K(x_i, x_a) - cK(x_i, x_d)\big)\big(K(x_j, x_a) - cK(x_j, x_d)\big). \end{aligned} \tag{8.4}$$

The bias terms become $b_l = -\frac{1}{1_N^T D^{-1} 1_N} 1_N D^{-1} \Omega^\star \alpha^{(l)}$ and the projected variables for an out-of-sample point x are:

$$z^{(l)}(x) = {w^{(l)}}^T \varphi(x) + b_l = \sum_{j=1}^{N} \alpha_j^{(l)} K^\star_{\text{eq}}(x, x_j) + b_l, l = 1, \ldots, k-1$$

Remark 8.2 (Related approach)**.** *A constrained spectral clustering algorithm from a completely different perspective was discussed in [72]. This method starts from a Gaussian process interpretation of the kernel matrix and leads to the same equivalent kernel* (8.4)*. However, [72] only aims at modifying the kernel matrix with no clear extensions to out-of-sample points. Moreover, interaction between* cannot-link *constraints create false similarities and negative entries in the equivalent kernel matrix. This fact may cause data points having negative graph degrees and therefore invalidating the computation of a graph Laplacian.*

8.2 Soft Pairwise Constraint via Regularization

Consider now introducing a *soft* pairwise constraint between x_a and x_d:

$$\min_{w^{(l)},e^{(l)},b_l,r_l} \frac{1}{2}\sum_{l=1}^{k-1}\frac{\gamma_l}{N}e^{(l)^T}D^{-1}e^{(l)} - \sum_{l=1}^{k-1}w^{(l)^T}w^{(l)} - \rho\sum_{l=1}^{k-1}r_l^2 \tag{8.5}$$

$$\text{such that } \begin{cases} e^{(1)} = \Phi w^{(1)} + b_1 1_N \\ \vdots \\ e^{(k-1)} = \Phi w^{(k-1)} + b_{k-1} 1_N \\ w^{(1)^T}\varphi(x_a) = cw^{(1)^T}\varphi(x_d) + r_1 \\ \vdots \\ w^{(k-1)^T}\varphi(x_a) = cw^{(k-1)^T}\varphi(x_d) + r_{k-1} \end{cases}$$

where $r_l, l = 1,\ldots,k-1$ are residual terms that allow some distance between the mapped x_a and x_b in the score variables space. The amount of discrepancy between the points is included in the objective function using a new regularization term $\rho \in \mathbb{R}^+$. The dual solutions are described in the following Lemma.

Lemma 8.2. ***Alzate & Suykens [3]*** *Given a positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$, regularization constants $\gamma_l \in \mathbb{R}^+, l = 1,\ldots,n_e$, $\rho \in \mathbb{R}^+$, the inverse of the graph degree matrix D^{-1} which is diagonal with positive elements and a constant $c \in \{-1, 1\}$ representing a pairwise constraint, then the KKT optimality conditions of the Lagrangian of* (8.2) *are satisfied by the following eigenvalue problem:*

$$\lambda_l\alpha^{(l)} = D^{-1}M_D\Omega^{\star\star}\alpha^{(l)} \tag{8.6}$$

where $\lambda_l = N/\gamma_l$, $\Omega^{\star\star} = \Omega - k_2(\Omega(:,a) - c\Omega(:,d))(\Omega(:,a) - c\Omega(:,d))^T$, $\Omega(:,i) \in \mathbb{R}^N$ is the i-th column of Ω, $k_2 = \rho/(1+\rho k_1)$, $k_1 = K(x_a,x_a) - 2cK(x_a,x_d) + c^2K(x_d,x_d)$, $M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N}1_N 1_N^T D^{-1}$, and $l = 1,\ldots,k-1$.

Proof: The Lagrangian of (8.5) becomes:

$$\mathcal{L}(w^{(l)}, e^{(l)}, b_l; \alpha^{(l)}, \beta_l) = \frac{1}{2}\sum_{l=1}^{k-1}\big(\frac{\gamma_l}{N}e^{(l)^T}D^{-1}e^{(l)} - w^{(l)^T}w^{(l)} - \rho r_l^2\big) -$$

$$\sum_{l=1}^{k-1}\alpha^{(l)^T}(e^{(l)} - \Phi w^{(l)} - b_l 1_N) - \sum_{l=1}^{k-1}\beta_l\big(w^{(l)^T}(\varphi(x_a) - c\varphi(x_d)) - r_l\big),$$

with KKT conditions for optimality:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w^{(l)}} = 0 \rightarrow w^{(l)} = \Phi^T \alpha^{(l)} - \beta_l(\varphi(x_a) - c\varphi(x_d)) \\ \dfrac{\partial \mathcal{L}}{\partial e^{(l)}} = 0 \rightarrow \alpha^{(l)} = (\gamma_l / N) D^{-1} e^{(l)} \\ \dfrac{\partial \mathcal{L}}{\partial b_l} = 0 \rightarrow 1_N^T \alpha^{(l)} = 0 \\ \dfrac{\partial \mathcal{L}}{\partial r_l} = 0 \rightarrow \beta_l = \rho r_l \\ \dfrac{\partial \mathcal{L}}{\partial \alpha^{(l)}} = 0 \rightarrow e^{(l)} = \Phi w^{(l)} + b_l 1_N \\ \dfrac{\partial \mathcal{L}}{\partial \beta_l} = 0 \rightarrow {w^{(l)}}^T (\varphi(x_a) - c\varphi(x_d)) = r_l. \end{cases}$$

Eliminating the primal variables $w^{(l)}, e^{(l)}, b_l, r_l$ and expressing the KKT conditions in terms of the dual variables $\alpha^{(l)}$ leads to the eigenvalue equation (8.6). ■

The bias terms become

$$b_l = -\frac{1}{1_N^T D^{-1} 1_N} 1_N D^{-1} \Omega^{\star\star} \alpha^{(l)} \tag{8.7}$$

and the projected variables for an out-of-sample point x:

$$z^{(l)}(x) = {w^{(l)}}^T \varphi(x) + b_l = \sum_{j=1}^{N} \alpha_j^{(l)} K_{\text{eq}}^{\star\star}(x, x_j) + b_l, \tag{8.8}$$

for $l = 1, \ldots, k-1$.

Remark 8.3 (Iterative approach)**.** *Note that incorporating a single pairwise constraint into the formulation leads to a modified metric induced by the equivalent kernel function. This means that multiple pairwise constraints can be incorporated by iteratively adding one constraint at a time and successively building the equivalent kernel matrix.*

Remark 8.4 (Soft constrained approach as general case)**.** *The equivalent kernel for the soft constrained problem becomes:*

$$\begin{aligned} K_{eq}^{\star\star}(x_i, x_j) =& K(x_i, x_j) - \\ & k_2\big(K(x_i, x_a) - cK(x_i, x_d)\big)\big(K(x_j, x_a) - cK(x_j, x_d)\big), \end{aligned}$$

where $k_2 = \rho/(1 + \rho k_1)$. *Note that, the equivalent kernel of the hard constrained problem* (8.4) *is a particular case of the soft constrained formulation when* $\rho \to \infty$. *Moreover, if* $\rho \to 0$ *then* $k_2 \to 0$ *and the soft constrained formulation becomes the standard multiway spectral clustering formulation* (6.3) *discussed in chapter 6. Therefore, the regularization term* ρ *controls the importance of the available prior knowledge in terms of pairwise constraints.*

An implementation of the proposed regularized approach with soft pairwise constraints is shown in Algorithm 7.

8.3 The Constrained BLF

In this section, we extend an existing model selection criterion for spectral clustering in order to incorporate pairwise constraints. Introducing an additional term measuring the accuracy of fulfilling the pairwise constraints into the BLF formula leads to the constrained BLF (cBLF):

$$\text{cBLF}(k) = \eta_l \text{linefit}(k) + \eta_b \text{balance}(k) + \eta_c \text{cfit}(k), \tag{8.9}$$

where $\text{cfit}(k)$ is the ratio of fulfilled constraints with respect to the total number of constraints, η_l, η_b, η_c are given parameters controlling the importance of the linefit, balance and constraint fit respectively and $\eta_l + \eta_b + \eta_c = 1$.

8.4 Experimental Results

In this section we report experimental results. The assessment of the clustering is done using the adjusted Rand index [59]. This external clustering performance measures the degree of mismatch between some known cluster indicators and the results of a clustering algorithm. The adjusted Rand index ranges from 0 in the case of complete mismatch to 1 in the case of a perfect match. The regularization term ρ can be different for *must-link* and *cannot-link* constraints and can be also incorporated into a model selection procedure. However, in this work we fixed this parameter to $\rho_m = 10^{-1}$ for *must-link* constraints and $\rho_c = 10^{-3}$ for *cannot-link* constraints. We choose ρ_c much smaller than ρ_m to avoid negative entries in the equivalent kernel matrix due to rank-1 downdates. For the BLF, η

Algorithm 7 Regularized Spectral Clustering with Pairwise Constraints

Input: Training set $\mathcal{D} = \{x_i\}_{i=1}^{N}$, positive definite kernel function $K(x_i, x_j)$, number of clusters k, ρ_m, ρ_c regularization terms for *must-link* and *cannot-link* constraints respectively, pairwise constraint index set $\mathcal{J} = \{(a_n, b_n)\}_{n=1}^{N_{\text{const}}}, a_n, b_n \in \mathbb{N}$, where N_{const} is the total number of pairwise constraints, test set $\{x_t^{\text{test}}\}_{t=1}^{N_{\text{test}}}$.

Output: Partition $\{\mathcal{A}_p\}_{p=1}^{k}$, cluster codeset $\{c_p\}_{p=1}^{k}, c_p \in \{-1, 1\}^{k-1}$.

1: Compute the kernel matrix Ω, $\Omega_{ij} = K(x_i, x_j), i, j = 1, \ldots, N$.
2: **for** $n = 1$ to N_{const} **do**
3: $x_a \leftarrow x_{a_n}, x_b \leftarrow x_{b_n}$ and let $c = -1$ for *cannot-link* and $c = 1$ for *must-link*.
4: Compute the equivalent kernel $\Omega^{\star\star}$ using Ω, ρ_m, ρ_c.
5: $\Omega \leftarrow \Omega^{\star\star}$
6: **end for**
7: Compute the eigenvectors $\alpha^{(l)}, l = 1, \ldots, k-1$ corresponding to the largest $k-1$ eigenvalues of $D^{-1}M_D\Omega^{\star\star}$ where D^{-1} is the inverse graph degree matrix of $\Omega^{\star\star}$.
8: Binarize the eigenvectors: $\text{sign}(\alpha_i^{(l)}), i = 1, \ldots, N, l = 1, \ldots, k-1$, and let $\text{sign}(\alpha_i) \in \{-1, 1\}^{k-1}$ be the encoding vector for the training data point x_i, $i = 1, \ldots, N$.
9: Count the occurrences of the different encodings and find the k encodings with most occurrences. Let the codeset be formed by these k encodings: $\mathcal{C} = \{c_p\}_{p=1}^{k}, c_p \in \{-1, 1\}^{k-1}$
10: $\forall i$, assign x_i to $\mathcal{A}_{p^\star}$ where $p^\star = \text{argmin}_p d_H(\text{sign}(\alpha_i), c_p)$ and $d_H(\cdot, \cdot)$ is the Hamming distance.
11: Binarize the test data projections $\text{sign}(z_m^{(l)}), m = 1, \ldots, N_{\text{test}}, l = 1, \ldots, k-1$ using (8.8) and let $\text{sign}(z_m) \in \{-1, 1\}^{k-1}$ be the encoding vector of x_m^{test}, $m = 1, \ldots, N_{\text{test}}$.
12: $\forall m$, assign x_m^{test} to $\mathcal{A}_{p^\star}$ where $p^\star = \text{argmin}_p d_H(\text{sign}(z_m), c_p)$.

was fixed to 0.75 and for the cBLF, $\eta_l = 0.5, \eta_b = 0.25, \eta_c = 0.25$. All experiments were performed on a MacBook Intel Core 2 Duo 2.4 GHz, 4 GB RAM in Matlab 7.7.

8.4.1 Toy Problems

The first toy problem of two moons and a ring for a total of $1,500$ data points. The training scenario consists of 500 data points for training and 500 data points for validation. Figure 8.1 shows the training set and the pairwise constraints. *Must-link* constraints are shown in green with circles and *cannot-link* constraints are shown in red with crosses. Figure 8.2 shows the model selection plots. The number of clusters k and the RBF kernel parameter σ^2 are obtained through the BLF using the validation set. For the original unconstrained problem, we used the BLF and for the problem with constraints, we used the cBLF. Figure 8.3 shows the clustering results.

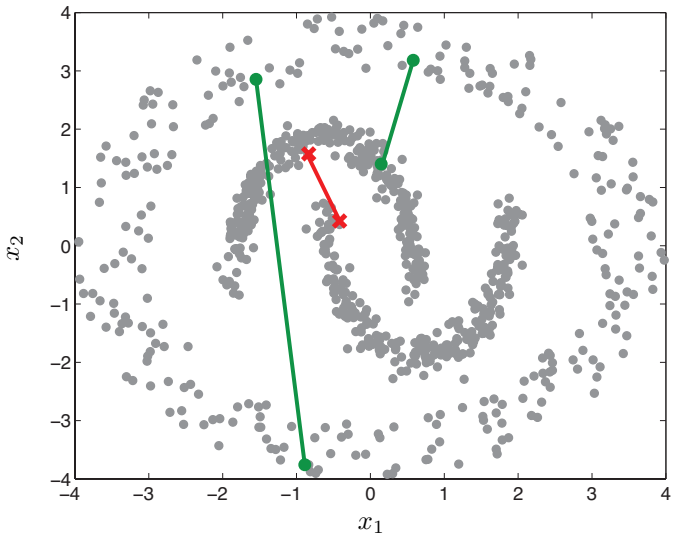


Figure 8.1: Toy problem 1 - Training set with constraints. *Must-link* constraints are shown in green with circles and *cannot-link* constraints are shown in red with crosses.

The second toy problem consists of four Gaussian clouds in $2D$ for a total of 900 data points. Model selection was performed in a similar way as in the toy problem 1 and is shown in Figure 8.4. Figure 8.5 shows the complete set with constraints. The constraints impose a "XOR"-like solution. Note that only two constraints were required to solve this problem.

The third toy problem consists of the Ripley dataset. The goal is to obtain a classification decision by iteratively increasing the number of pairwise constraints in a semi-supervised way. In this case, we compared with the

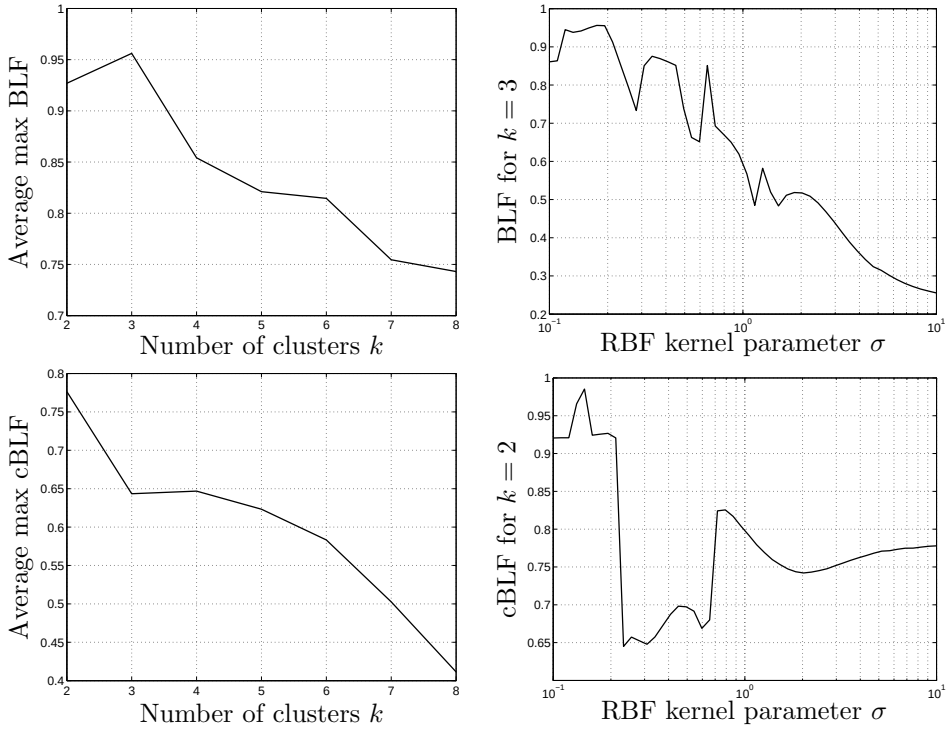


Figure 8.2: Toy problem 1 - Model selection on validation data using the BLF and the cBLF. **Top row:** Tuning without constraints (BLF). **Bottom row:** Tuning with 3 pairwise constraints (cBLF). Note that the constraints shift the cBLF to be maximal at 2 clusters instead of 3 clusters like in the unconstrained case.

constrained spectral clustering method described in [72]. However, since this method does not have out-of-sample extensions, we combine it with the Nyström method discussed in [46] to be able to use a smaller set of data points for training and then approximate the full eigenvectors. The dataset set was divided into 250 points for training, 500 for validation and 500 for testing. The RBF kernel parameter σ^2 was obtained through the cBLF each time a pairwise constraint was added. The pairwise constraint set was randomized 10 times. Figure 8.6 shows the average adjusted Rand index (ARI) with respect to the true labels in terms of increasing number of constraints. The proposed method achieves an ARI of 0.63 with 125 pairwise constraints while [72] achieves an ARI of 0.31. Figure 8.7 shows the results of the full Ripley dataset after model selection on validation data.

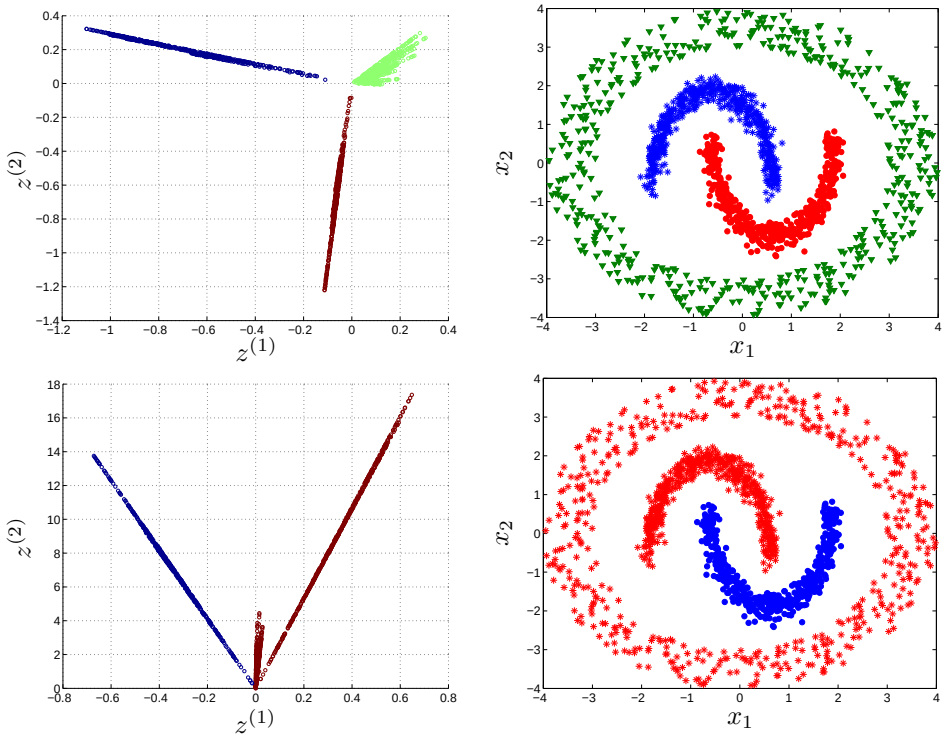


Figure 8.3: Toy problem 1 - Clustering results after tuning. **Top row:** Original problem without constraints. **Bottom row:** Clustering problem with pairwise constraints. The first column shows the score variables space. Note the strong line structures detected by the model selection criteria. The constraints cause that the line structure corresponding to the ring moves to the same quadrant as the upper moon and therefore belonging to the same cluster.

The proposed method outperforms [72] in terms of agreement with the true labels.

8.4.2 Image Segmentation

In this experiment, we used an image from the Berkeley image database [75]. We computed a local color histogram with a 5×5 pixels window around each pixel using minimum variance color quantization of 8 levels. We used the χ^2 test to compute the distance between two local color histograms

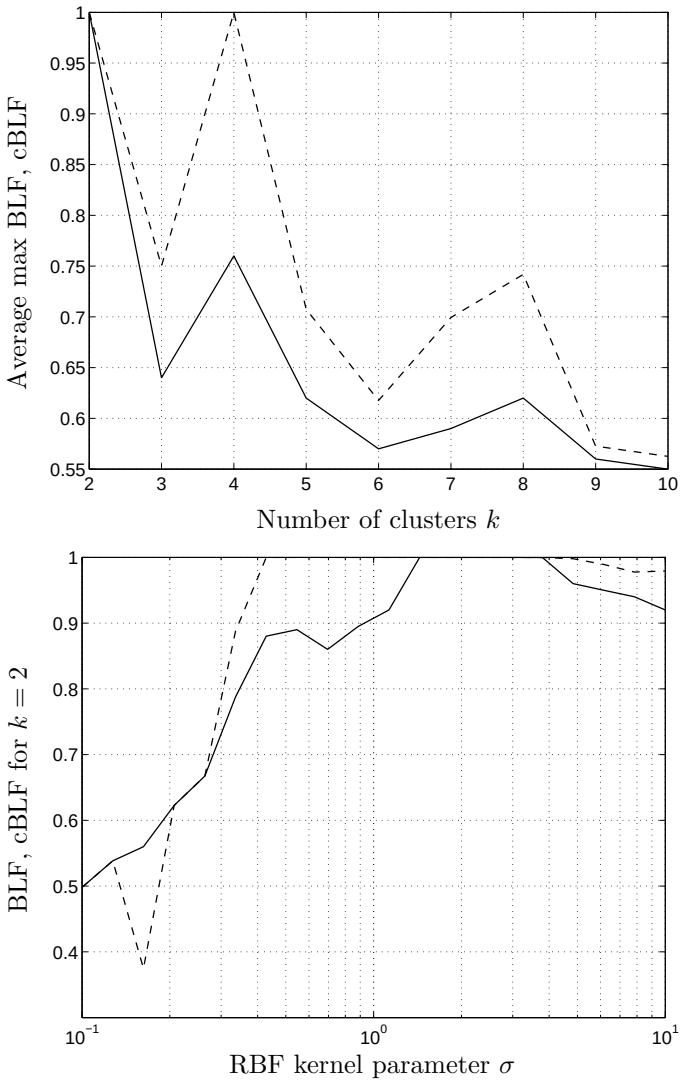


Figure 8.4: Toy problem 2 - Model selection on validation data using the BLF and the cBLF. The dashed line corresponds to the BLF and the solid line to the cBLF. **Top:** The optimal number of clusters according to the BLF (unconstrained) is 2 or 4. The cBLF gives preference to 2 clusters to fulfill the imposed pairwise constraints. **Bottom:** Determining the optimal RBF kernel parameter. The cBLF narrows down the optimal value of σ.

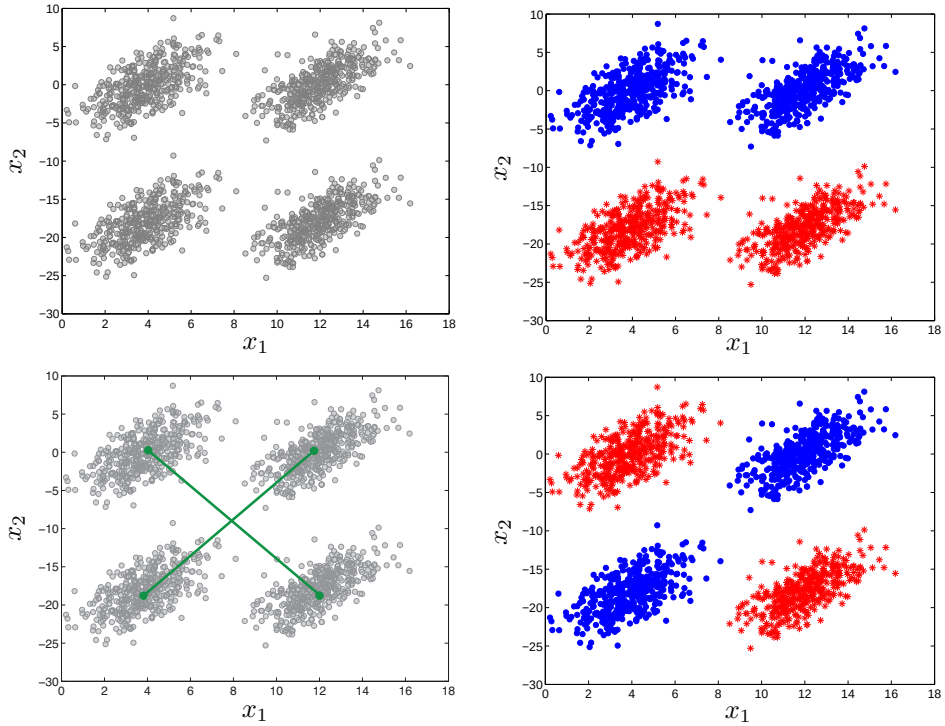


Figure 8.5: Toy problem 2 - Clustering results. **Top row:** Original problem without constraints. **Bottom row:** Clustering problem with pairwise constraints. The clustering parameters (k, σ^2) were obtained through the BLF and cBLF as shown in Figure 8.4. Note that only two constraints were required to solve this problem.

$h^{(i)}$ and $h^{(j)}$ [88] $\chi^2_{ij} = 0.5 \sum_{b=1}^{B} (h_b^{(i)} - h_b^{(j)})^2 / (h_b^{(i)} + h_b^{(j)})$, where B is total number of quantization levels. The histograms are assumed to be normalized $\sum_{b=1}^{B} h_b^{(i)} = 1, i = 1, \ldots, N$. The χ^2 kernel $K(h^{(i)}, h^{(j)}) = \exp(-\chi^2_{ij}/\sigma_\chi)$ with parameter $\sigma_\chi \in \mathbb{R}^+$ is positive definite and has shown to be a robust and efficient way of comparing the similarity between two histograms for color discrimination and image segmentation [46]. The image is 321×481 pixels wide giving a total number of data points of $154,401$. The training scenario is 600 pixel histograms for training, $3,000$ for model selection using the cBLF and the remaining pixel histograms are the test set. Note that [72] alone cannot cope with the full image since it requires storing and computing the eigendecomposition of a $154,401 \times 154,401$ kernel matrix. Hence, we use the Nyström method for spectral clustering described in [46]

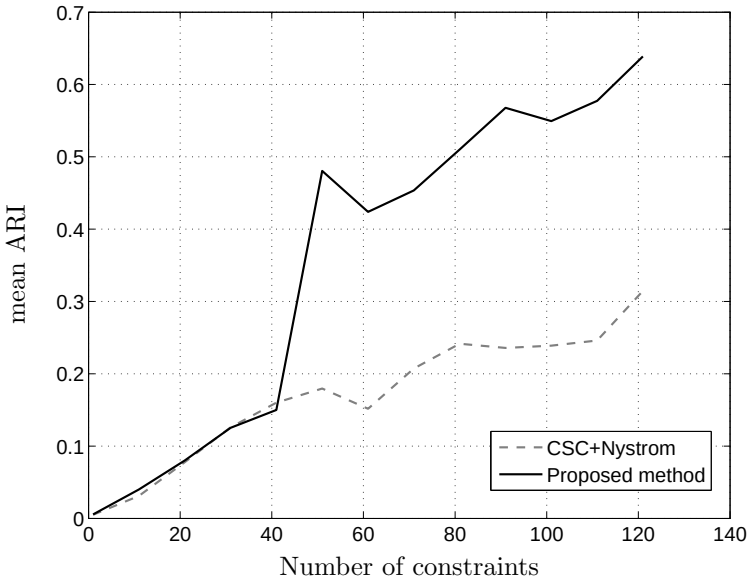


Figure 8.6: Toy problem 3 - Ripley dataset. Mean adjusted Rand index (ARI) of the full dataset ($1,250$ points) with respect of numbers of pairwise constraints over 10 randomizations of the constraints. The training set was fixed to 250 points. **Dashed line:** Constrained spectral clustering algorithm of [72] together with the Nyström approximation [46]. **Solid line:** Proposed method. The proposed method outperforms [72] in terms of agreement with the true labels.

to approximate the eigenvectors and be able to compare with the proposed method. The computation time of the proposed method was in average 15 seconds while [72] combined with [46] was in average 170 seconds. Figure 8.8 shows the original image with constraints and the results of [72] and the proposed method. The proposed method produces a visually appealing clustering with respect to the given constraints.

8.5 Conclusions

A regularized formulation to incorporate pairwise constraints into spectral clustering is proposed. This formulation is based on a weighted kernel PCA framework in which the clustering model can be extended to new points. The proposed approach in cast in a constrained optimization framework

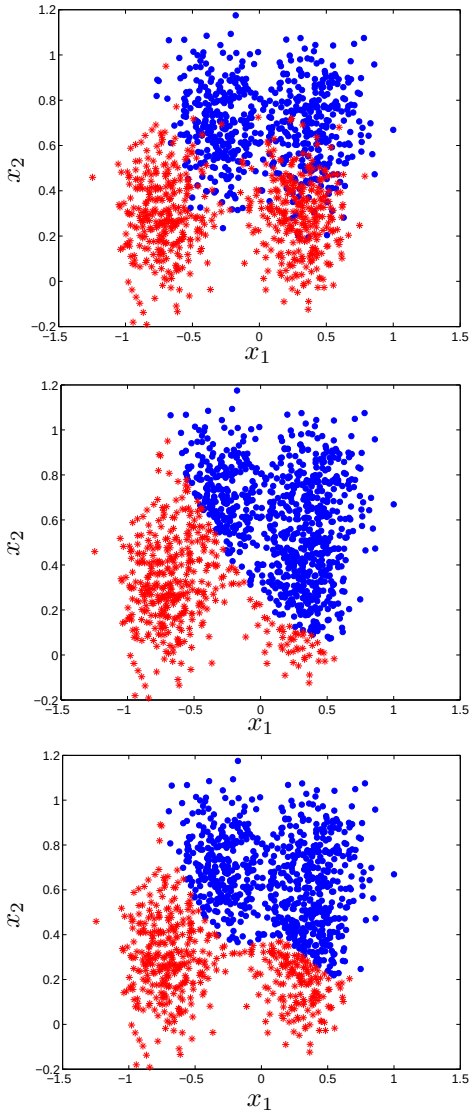


Figure 8.7: Toy problem 3 - Ripley dataset results after model selection using the cBLF on validation data. **Top:** True classification. **Center:** Results of [72] together with the Nyström approximation [46] using 125 constraints. The ARI with respect to the true labels is 0.31. **Bottom:** Proposed method using 125 constraints. The ARI is 0.63. The proposed method outperforms [72] in terms of agreement with the classification labels.

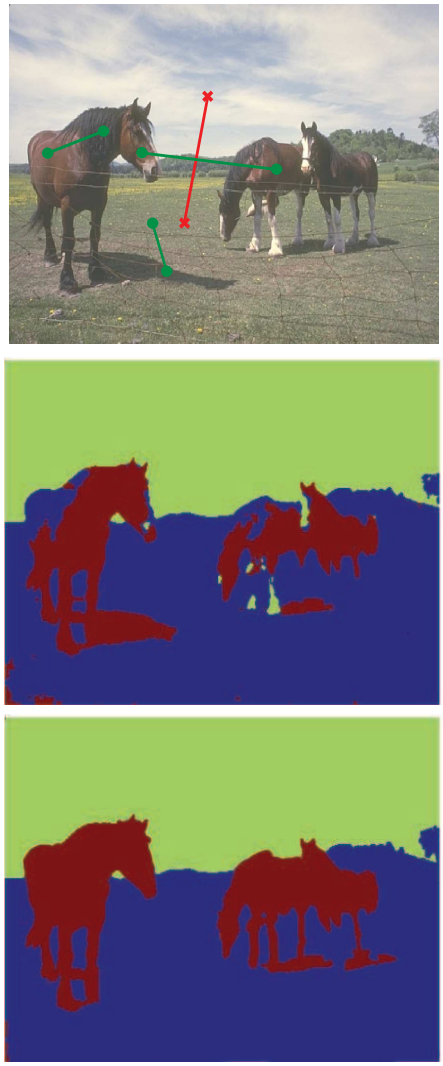


Figure 8.8: Image Segmentation - Clustering results. The image is 321×481 pixels wide, for a total of $154,401$ data points. The training set consists of 600 randomly selected pixel histograms and the cluster model was extended to the remaining data points using the out-of-sample extension. **Top:** Original image with constraints. **Center:** Clustering results of [72] together with the Nyström approximation [46]. Note that [72] alone cannot cope with the large number of data points of this image. **Bottom:** Proposed method. The proposed method produces a visually appealing clustering with respect to the given constraints.

allowing pairwise constraints to be naturally added in the primal problem. The corresponding dual problem is an eigenvalue problem of a modified kernel matrix where the constraints correspond to rank-1 downdates of the original affinity. The proposed method can be trained, validated and tested in a learning framework using a model selection criterion. Simulations results with toy examples and an image segmentation problem show the applicability of the proposed method.

Chapter 9

General Conclusions

9.1 Concluding Remarks

This thesis discusses a series of methodologies for unsupervised learning based on support vector machines and particularly on LS-SVMs. Core kernel models for unsupervised learning are extended towards the incorporation of advanced elements into the formulations. This modular approach based on the primal-dual nature typical of LS-SVMs allows the incorporation of key aspects such as regularization and prior information into the primal problem which is formulated in high dimensional feature spaces. The corresponding dual problem obtained through the Karush-Kuhn-Tucker optimality conditions of the Lagrangian appears now in terms of kernel functions after the application of Mercer's theorem. A benefit of using this constrained optimization framework for unsupervised learning lies in understanding the role of regularization, the importance of model selection and the extension to out-of-sample data. These three critical aspects are typically missing in classical unsupervised learning methods due to lack of an underlying model.

This thesis extends LS-SVM formulations for unsupervised learning to the following aspects:

- **Imposing a general loss function into kernel PCA**. Considering an epsilon-insensitive Huber loss function, robustness and sparseness are imposed into kernel PCA. This formulation leads to solving a set

of nonlinear equation via an SQP method. Chapter 4.

- **Considering a weighed formulation for kernel PCA.** An iterative approach is proposed to solve kernel PCA with general loss function in an fast and efficient way. This formulation considers weighted least squares and leads to solving a sequence of generalized eigenvalue problems. Chapter 4.
- **Extending the LS-SVM formulation to kernel CCA towards the multivariate case.** An extension of kernel CCA based on LS-SVM to more than two sets of data is proposed, where regularization is incorporated in a natural way. Chapter 5
- **Introducing the kernel regularized correlation (KRC).** The KRC can be used to characterize pairwise independence in the input space by measuring correlation in the feature space induced by universal kernels. Chapter 5.
- **Linking several algorithms for binary spectral clustering with weighted kernel PCA**. For a particular choice of the weights, links to well-known binary spectral clustering methods can be made. An additional insight is the out-of-sample extension. Chapter 6.
- **Introducing a multiway spectral clustering algorithm with out-of-sample extensions**. This formulation is also based on weighted kernel PCA with additional constraints and bias terms. This leads to a clustering model with out-of-sample extensions and a model selection criterion to exploit the structure of the solutions. Chapter 6.
- **Incorporating sparseness and prior information into multiway spectral clustering**. The clustering model is further extended towards sparse solutions which becomes particularly important for large-scale data. Prior information is also added in the form of pairwise restriction as additional constraints in the primal problem. Chapters 7 and 8.

9.2 Future Research

The thesis presents some contributions in unsupervised learning using support vector methods. This research work can be extended in several

directions.

- A natural extension of this work consists of bridging the gap between unsupervised and supervised learning. Imposing pairwise constraints on the cluster assignment for spectral clustering can be considered as a form of supervision, thus shifting the clustering task towards semi-supervised learning. However, a broader type of prior knowledge and constraints can be still incorporated into the analysis not only for clustering but also with respect to other unsupervised learning applications.

- Techniques based on the Nyström approximation and incomplete Cholesky decomposition have been developed specifically for large-scale spectral clustering. Sparse models applicable to a broader type of applications with large-scale data become an area of open research. Algorithms for extremely large datasets such as the PageRank approximation can also be useful in this context.

- The multiway spectral clustering method proposed in chapter 6 can be also further extended to several key aspects such as a hierarchical decomposition; relation to multiclass problems in semi-supervised learning; different centerings of the data in the feature space and structural properties of the eigenvectors and projections beyond collinearity.

- The successful application of a kernel method depends greatly on an appropriate kernel function for the data together with carefully selected model parameters. We have shown a model selection criterion for spectral clustering based on a structural property of the solutions. However, more generic types of model selection and kernel design criteria for unsupervised and semi-supervised learning become an important future research of this work.

- Another direction for future research is to extend the current kernel models to work with time-dependent data. Time series are found in several domains including data generated by biological processes, speech and energy consumption records. Incorporating information about the dynamical process underlying the data into the current kernel models becomes a key aspect.

Appendix A

The Nyström Method

The Nyström method [18] is a technique for finding numerical approximations of eigenfunctions. This technique aims at approximating the corresponding eigenfunction using a predetermined subset of $M \ll N$ training points and then extrapolating the approximation to the full set of points. This method has been used for speeding up SVMs and Gaussian processes in classification and regression [115], for estimation in the primal space in LS-SVM models leading to sparse versions [102] and for reducing the computational burden in spectral clustering eigenvalue problems [46]. This appendix summarizes the Nyström approximation.

Consider the following integral equation:

$$\int K(x_j, x)\phi_i(x)p(x)dx = \lambda_i \phi_i(x_j)$$

where x is a random variable with probability density $p(x)$. Given a finite sample $\{x_j\}_{j=1}^N$, the integral can be approximated by:

$$\frac{1}{N}\sum_{l=1}^{N} K(x_j, x_l)u_i(x_l) = \hat{\lambda}_i u_i(x_j)$$

which can be written as the following eigenvalue problem:

$$\Omega U = N\hat{U}\Lambda$$

where Ω is the kernel matrix with ij-th entry $\Omega_{ij} = K(x_i, x_j)$, U is the $N \times N$ matrix of eigenvectors $U = [u_1, u_2, \ldots, u_N]$, $u_i \in \mathbb{R}^N, i = 1, \ldots, N$ and $\hat{\Lambda}$ is a $N \times N$ diagonal matrix of ordered eigenvalues $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_N$. The original eigenvalues λ_i and eigenfunctions ϕ_i can be approximated directly for points in the sample with:

$$\begin{aligned} \lambda_i &\approx \frac{\hat{\lambda}_i}{N}, \\ \hat{\phi}_i(x_j) &\approx \sqrt{N} U_{ji}. \end{aligned}$$

One of the key points of the Nyström method is the possibility to obtain approximations of the eigenfunction evaluated at any point x:

$$\hat{\phi}_i(x) = \frac{\sqrt{N}}{\hat{\lambda}_i} \sum_{j=1}^{N} k(x, x_j) U_{ji}, \tag{A.1}$$

which can be used to build an M-dimensional approximation of a given point x in the feature space:

$$\hat{\varphi}(x) = \left[\sqrt{\lambda_1}\hat{\phi}_1(x), \sqrt{\lambda_2}\hat{\phi}_2(x), \ldots, \sqrt{\lambda_M}\hat{\phi}_M(x)\right]^T.$$

where $M \ll N$.

This finite representation of a point in the feature space can be used to solve LS-SVM models directly at the primal level [102]. Note that (A.1) is related to the projections of a given point x onto the kernel PCA eigenspace.

Bibliography

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

[2] S. Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*, Tokyo, 2001. Springer-Verlag.

[3] C. Alzate and J. A. K. Suykens. A regularized formulation for spectral clustering with pairwise constraints. Internal Report 09-07, ESAT-SISTA, K.U.Leuven (Leuven, Belgium). Accepted for publication in IJCNN 2009.

[4] C. Alzate and J. A. K. Suykens. Extending kernel principal component analysis to general underlying loss functions. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 214–219, 2005.

[5] C. Alzate and J. A. K. Suykens. A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN'06)*, pages 138–144, 2006.

[6] C. Alzate and J. A. K. Suykens. ICA through an LS-SVM based kernel CCA measure for independence. In *Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN'07)*, pages 2920–2925, 2007.

[7] C. Alzate and J. A. K. Suykens. Image segmentation using a weighted kernel PCA approach to spectral clustering. In *Proc. of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)*, pages 208–213, Honolulu, USA, April 2007.

[8] C. Alzate and J. A. K. Suykens. Kernel component analysis using an epsilon insensitive robust loss function. *IEEE Transactions on Neural Networks*, 19(9):1583–1598, 2008.

[9] C. Alzate and J. A. K. Suykens. A regularized kernel CCA contrast function for ICA. *Neural Networks*, 21(2-3):170–181, March 2008.

[10] C. Alzate and J. A. K. Suykens. Sparse kernel models for spectral clustering using the incomplete cholesky decomposition. In *Proc. of the 2008 International Joint Conference on Neural Networks (IJCNN 2008)*, pages 3555–3562, 2008.

[11] C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence. In Press*, 2009.

[12] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 757–763, Cambridge, MA, 1996. MIT Press.

[13] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[14] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[15] F. R. Bach and M. I. Jordan. Learning spectral clustering. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[16] F. R. Bach and M. I. Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7:1963–2001, 2006.

[17] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[18] C. Baker. *The numerical treatment of integral equations*. Clarendon Press, Oxford, 1977.

[19] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 59–68, Seattle, WA, 2004.

[20] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585–591, Cambridge, MA, 2002. MIT Press.

[21] Y. Bengio, J. -F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[22] C. J. C. Burges. Simplified support vector decision rules. In *Proceedings of the 13th Intl. Conf. on Machine Learning*, pages 71–77, 1996.

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

[24] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, 13(9):1088–1096, 1994.

[25] A. Christmann and I. Steinwart. On robustness properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, 5:1007–1034, August 2004.

[26] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[27] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[28] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–297, 1995.

[29] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[30] F. Critchley. Influence in principal component analysis. *Biometrika*, 72:627–636, 1985.

[31] C. Croux and G. Haesbroeck. Principal component analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *Biometrika*, 87(3):603–618, 2000.

[32] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[33] F. De la Torre and M. Black. Robust principal component analysis for computer vision. In *In Proc. International Conference on Computer Vision (ICCV'2001)*, pages 362–369, 2001.

[34] M. Debruyne, M. Hubert, and J. A. K. Suykens. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9:2377–2400, 2008.

[35] M. Debruyne, M. Hubert, and J. Van Horebeek. Detecting influential observations in kernel PCA. Technical Report TR-07-11, Department of Mathematics, Section of Statistics, K.U.Leuven (Leuven, Belgium). Submitted for publication, 2008.

[36] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[37] I.S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *10th ACM Knowledge Discovery and Data Mining Conf.*, pages 551–556, 2004.

[38] I.S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.

[39] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Machine Learning Research*, 2:263–286, 1995.

[40] C. Ding and X. He. K-means clustering via principal component analysis. In *ICML '04: Proceedings of the 21st international conference on Machine learning*, page 29, 2004.

[41] C. Ding and X. He. Linearized cluster assignment via spectral ordering. In *ICML '04: Proceedings of the twenty-first International Conference on Machine Learning*, page 30, New York, NY, USA, 2004. ACM Press.

[42] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1999.

[43] M. Espinoza, C. Joye, R. Belmans, and B. De Moor. Short-term load forecasting, profile identification and customer segmentation: A methodology based on periodic time series. *IEEE Transactions on Power System*, 20(3):1622–1630, August 2005.

[44] K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations. *Proc. Natl. Acad. Sci. USA*, 35(11):652–655, 1949.

[45] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czech. Math. J.*, 25(100):619–633, 1975.

[46] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, February 2004.

[47] R. Gittins. *Canonical Analysis - A Review with Applications in Ecology.* Berlin: Springer Verlag, 1985.

[48] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1996.

[49] A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.

[50] M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. Technical report, Penn. State Univ, Computer Science and Engineering., 2001.

[51] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning algorithms. *IEEE Trans. Computer-Aided Design*, 11(9):1074–1085, 1992.

[52] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21st international conference on Machine learning*, pages 369–376, 2004.

[53] X. He and P. Niyogi. Locality preserving projections. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[54] I. Higuchi and S. Eguchi. The influence function of principal component analysis by self-organizing rule. *Neural Computation*, 10:1435–1444, 1998.

[55] I. Higuchi and S. Eguchi. Robust principal component analysis with adaptive selection for tuning parameters. *Journal of Machine Learning Research*, 5:453–471, 2004.

[56] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[57] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[58] P. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

[59] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, pages 193–218, 1985.

[60] M. Hubert, P. J. Rousseeuw, and K. Vanden Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 47:64–79, 2005.

[61] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[62] A. Hyvärinen and E. Oja. A fast fixed point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[63] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

[64] F. Janssens. *Clustering of Scientific Fields by Integrating Text Mining and Bibliometrics*. PhD thesis, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium, 2007.

[65] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[66] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 561–566, 2003.

[67] J. R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

[68] D. D. Kosambi. Statistics in function space. *J. Indian Math. Soc.*, 7:76—88, 1943.

[69] J. T. Kwok and I. W. Tsang. The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks*, 15(6):1517–1525, November 2004.

[70] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(10):365–377, 2000.

[71] M. Loève. *Probability Theory*. D. Van Nostrand Company, Inc, New Jersey, second edition, 1960.

[72] Z. Lu and M. Carreira-Perpiñán. Constrained spectral clustering through affinity propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[73] J. MacQueen. Some methods for classification and analysis of multivariate observations. In University of California Press., editor, *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[74] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):1–20, 2004.

[75] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, 2001.

[76] M. Meila. Multiway cuts and spectral clustering. Technical report, University of Washington, 2003.

[77] M. Meila and J. Shi. Learning segmentation by random walks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[78] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Artificial Intelligence and Statistics AISTATS*, 2001.

[79] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 1909.

[80] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K. Müller. Fisher discriminant analysis with kernels. In *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, pages 41–48, 1999.

[81] S. Mika, B. Schölkopf, A. J. Smola, K. R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[82] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103(23):8577–8582, 2006.

[83] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 2004.

[84] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, 2002. MIT Press.

[85] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2000.

[86] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2(6):559–572, 1901.

[87] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, editor, *Proceedings of the 1977 Dundee Biennial Conference on Numerical Analysis*, pages 144–157. Springer Verlag, 1978.

[88] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Computer Vision and Pattern Recognition*, pages 267–272, 1997.

[89] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[90] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley New York (N. Y.), 1987.

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

[92] G. Salton and M.J. Mcgill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1986.

[93] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

[94] I. Schoenberg. Metric spaces and positive definite functions. *American Mathematical Society*, 44(3):522–536, 1938.

[95] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. R. Müller, M. Scholz, G. Rätsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, September 1999.

[96] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

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

[98] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):888–905, 2000.

[99] A. J. Smola, O. Mangasarian, and B. Schölkopf. Sparse kernel feature analysis. Technical report 99-03, University of Wisconsin, Data Mining Institute, Madison, 1999.

[100] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[101] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1-4):85–105, 2002.

[102] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[103] J. A. K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor. A support vector machine formulation to PCA analysis and its kernel version. *IEEE Transactions on Neural Networks*, 14(2):447–450, March 2003.

[104] J. A. K. Suykens and J. Vandewalle. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[105] T. Takahashi and T. Kurita. Robust de-noising by kernel pca. *International Conference on Artificial Neural Networks (ICANN2002). In Artificial Neural Networks - ICANN2002, Springer*, pages 739–744, 2002.

[106] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[107] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society B*, 63:411–423, 2001.

[108] W. S. Torgeson. Multidimensional scaling of similarity. *Psychometrika*, 30:379–393, 1965.

[109] C. Van Rijsbergen. *Information Retrieval*. Dept. of Computer Science, Univ. of Glasgow, 1979.

[110] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[111] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[112] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.

[113] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1978.

[114] C. K. I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning and Inference in Graphical Models*, pages 599–621. Kluwer, 1998.

[115] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[116] L. Xu and A. L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1):131–143, January 1995.

[117] Y. Yamanishi, J.-P. Vert, A. Nakaya, and M. Kanehisa. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, 19:323i–330i, 2003.

[118] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In L. K. Saul, Y. Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1601–1608, Cambridge, MA, 2005. MIT Press.

[119] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

Curriculum Vitae

Carlos A. Alzate P. was born in Tuluá, Colombia in April 1980. He received the degree in Electronic Engineering from the Universidad Nacional de Colombia, Manizales, Colombia, in 2002 and the Master degree in Artificial Intelligence from the Katholieke Universiteit Leuven, Leuven, Belgium in 2004. In 2005, he started the Ph.D program at the Department of Electrical Engineering (ESAT) of the K.U.Leuven under the supervision of Prof. Dr. Ir. Johan Suykens. Carlos' scientific interests include kernel methods, optimization, unsupervised learning and model selection.

Publications by the author

Journal Papers

- C. Alzate and J. A. K. Suykens. Kernel component analysis using an epsilon insensitive robust loss function. *IEEE Transactions on Neural Networks*, 19(9):1583–1598, 2008.
- C. Alzate and J. A. K. Suykens. A regularized kernel CCA contrast function for ICA. *Neural Networks*, 21(2-3):170–181, Mar. 2008.
- C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence. In Press*, 2009.
- C. Alzate, F. Janssens, W. Glänzel, J. A. K. Suykens and B. De Moor. Spectral clustering of scientific journals using a data integration scheme. *In preparation*, 2009.

International Conference Papers

- C. Alzate and J. A. K. Suykens. Extending kernel principal component analysis to general underlying loss functions. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 214–219, 2005.
- C. Alzate and J. A. K. Suykens. A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN'06)*, pages 138–144, 2006.
- C. Alzate and J. A. K. Suykens. ICA through an LS-SVM based kernel CCA measure for independence. In *Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN'07)*, pages 2920–2925, 2007.

- C. Alzate and J. A. K. Suykens. Image segmentation using a weighted kernel PCA approach to spectral clustering. In *Proc. of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)*, pages 208–213, Honolulu, USA, Apr. 2007.
- C. Alzate and J. A. K. Suykens. Sparse kernel models for spectral clustering using the incomplete cholesky decomposition. In *Proc. of the 2008 International Joint Conference on Neural Networks (IJCNN 2008)*, pages 3555–3562, 2008.
- C. Alzate and J. A. K. Suykens. A regularized formulation for spectral clustering with pairwise constraints. Internal Report 09-07, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2009. Accepted for Publication in IJCNN'09.
- C. Alzate, M. Espinoza, B. De Moor and J. A. K. Suykens. Identifying customer profiles in power load time series using spectral clustering. Internal Report 09-59, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2009. Submitted for publication.

Abstracts

- C. Alzate and J. A. K. Suykens, Robust Kernel Principal Component Analysis using Huber's Loss Function, Benelux Meeting on Systems and Control, Houffalize, Belgium, 2005.
- C. Alzate and J. A. K. Suykens, Extending Kernel PCA to General Underlying Loss Functions, The Analysis of Patterns Doctoral School, Erice, Italy, 2005.
- C. Alzate and J. A. K. Suykens, Generalized Kernel PCA, 21st European Conference on Operational Research (EURO 2006), Reykjavik, Iceland, 2006. (Invited talk).
- C. Alzate and J. A. K. Suykens, An Out-of-Sample Extension for Spectral Clustering based on Weighted Kernel PCA, CIL Research Day, Brussels, Belgium, 2006.
- C. Alzate and J. A. K. Suykens, A Unifying Framework for Spectral Clustering based on Weighted Kernel PCA, Benelearn 2006, Ghent, Belgium, 2006.
- C. Alzate and J. A. K. Suykens, An Out-of-Sample Extension to Spectral Clustering based on Weighted Kernel PCA and its Application to Image Segmentation, International Workshop on Current Challenges in Kernel Methods (CCKM06), Brussels, Belgium, 2006.

- C. Alzate and J. A. K. Suykens, A Weighted Kernel PCA Approach to Graph-Based Image Segmentation, Embeddings of Graphs and Groups into Hilbert and Banach Spaces, Lausanne, Switzerland, 2007.
- C. Alzate and J. A. K. Suykens, A Contrast Function for ICA Using LS-SVM Based Formulations to Kernel CCA, Benelearn 2007, Amsterdam, The Netherlands, 2007.
- M. Espinoza, C. Alzate, J. A. K. Suykens and B. De Moor, Clustering of Time Series: Looking for the Dynamics, IAP VI/4, DYSCO Workshop, Brussels, Belgium, 2007.
- C. Alzate and J. A. K. Suykens, Multiway Spectral Clustering through Weighted Kernel PCA, 22nd European Conference on Operational Research (EURO 2007), Prague, Czech Republic, 2007. (Invited talk).