Kernel Choice for Unsupervised Kernel Methods

A dissertation

submitted in partial fulfillment of the requirements for the degree

Doctor of Philosophy

by

Md. Ashad Alam

in

Department of Statistical Science

The Institute of Statistical Mathematics

The Graduate University of Advanced Studies

Tokyo 190-8562, Japan.

September 2014

The dissertation of Md. Ashad Alam is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

Prof. Satoshi Kuriki

Dr. Shotaro Akaho

Prof. Daichi Mochihashi

Prof. Kenji Fukumizu

Adviser

Prof. Koji Kanefuji

Department Head/ Director

September 2014

DEDICATION

Our motherland

Gônôprôjatôntri ***Bangladesh***

(The People's Republic of ***Bangladesh****)*

EPIGRAPH

Without ***Modern Mathematics***

Statistics, Robust Statistics and Statistical Machine Learning are like

Gardening without any Flowers.

Contents

1 Introduction **1**

1.1 Designing kernel for unsupervised kernel methods 4

1.2 Robustness of unsupervised kernel methods 5

1.3 Scope of the research 5

1.4 Outline and summary of the contributions 6

2 Preliminary **7**

2.1 Inner product space 8

2.2 Hilbert space 10

2.3 Feature space and its drawback 12

2.3.1 Feature space 14

2.3.2 Computational problems of feature space 14

2.4 Kernel and positive definite kernel 15

2.4.1 Kernel 15

2.4.2 Advantages of kernel 18

2.4.3 Positive definite kernel (PDK) 19

2.4.4 Well-known PDKs and its parameters 21

2.4.5 Valid PDKs and its matrices 24

2.4.6 Properties of PDKs 26

2.4.7 Interplay between PDK, reproducing kernel and Mercer kernel . . . 31

2.5 Reproducing kernel Hilbert space (RKHS) 33

2.5.1 Properties of RKHS 34

2.5.2 Representer theorem 37

2.6 Methods based on RKHS 38

2.6.1 Supervised learning . . . 39
2.6.2 Unsupervised learning . . . 39
2.6.3 Nonparametric inference . . . 39

3 Automatic Way of Finding Hyperparameters in Kernel Principal Component Analysis 40

3.1 Motivation . . . 40
3.1.1 Kernel principal component analysis (kernel PCA) . . . 42
3.1.2 Choice of kernel . . . 43
3.2 Proposed methods . . . 45
3.2.1 Pre-Image of kernel PCA . . . 45
3.2.2 Hyperparameters choice . . . 47
3.2.3 Computational cost . . . 48
3.3 Experimental results . . . 49
3.3.1 Synthesized data . . . 49
3.3.2 Real world problems . . . 51
3.3.3 Polynomial kernel . . . 52
3.4 Discussion . . . 53

4 Classical, Robust and Kernel Canonical Correlation Analysis 58

4.1 Motivation . . . 58
4.2 Classical and robust canonical correlation analysis . . . 59
4.3 Measure of robustness . . . 61
4.3.1 Qualitative robustness . . . 61
4.3.2 Influence function . . . 61
4.3.3 Breakdown point . . . 62
4.4 Kernel canonical correlation analysis . . . 62
4.5 Experimental results . . . 64
4.5.1 Simulation results . . . 65
4.5.2 Sensitivity analysis . . . 70
4.5.3 Breakdown analysis . . . 70

5 Higher-order Regularized Kernel Canonical Correlation Analysis 74

5.1 Motivation 74

5.2 Cross-validation for the standard kernel canonical correlation analysis . . . 75

5.3 Distribution of a low dimensional projection 77

5.4 Higher-order regularized kernel CCA (hrKCCA) 80

5.4.1 Method 80

5.4.2 Kernel choice for hrKCCA 82

5.4.3 Computational issues 83

5.5 Experiments 84

5.5.1 Synthesized examples 84

5.5.2 Real world datasets 87

6 Conclusion and Future Research 105

6.1 Conclusion 105

6.2 Future research 107

List of Tables

2.1 Examples of well-known positive definite kernels and its parameters. 23

3.1 Computational cost (in second) of the proposed method for synthesized data-2 with different data sizes (n) and the numbers of components (ℓ) . . . 48

3.2 The configuration of datasets for hyperparmeters choice in kernel principal component analysis (kernel PCA). 50

3.3 Five real-world data sets: leave-one-out cross validation (LOOCV) reconstruction errors and LOOCV classification errors for inverse bandwidths (s) and the number of components (ℓ). The minimum values are written in bold fonts, and the classification errors with the hyperparameters chosen by the proposed method are underlined. 55

3.4 USPSG-500: LOOCV reconstruction errors and LOOCV classification errors (bold numbers indicate the minimum value). 56

3.5 LOOCV reconstruction errors for *food* data. 56

3.6 Polynomial kernel for *wine* data: LOOCV reconstruction errors and the LOOCV classification errors (bold numbers indicate the minimum value). . 57

4.1 Bias and mean square error of simulated data (bold numbers indicate the minimum value). 66

4.2 The value of qualitative robustness index. 66

5.1 Time complexity (in second) for example 3: the proposed kernel CCA with three numbers of iterations (I) and the standard kernel CCA (KCCA). The Gaussian kernel is used for the both methods, and n is the sample size. . . . 84

5.2 The configuration of datasets along with purposes (estimate of dependence feature (EDF), measure of association (MA) and estimate of low dimensional space (ELDS)) of all experimental datasets. 85
5.3 Cross-validation errors of the three examples ($E_1 - E_3$) using different inverse bandwidths s and regularization coefficients λ for the proposed method and standard kernel CCA (KCCA). 87
5.4 Cross-validation errors for *nutrimouse* dataset. 88
5.5 Classification errors (%) for *wine*, *BUPA liver disorders* and *diabetes*. One or two dimensional features are used with the proposed method (hrKCCA+kNN and hrKCCA+SVM) and the kernel CCA. 91
5.6 Classification errors (%) using one dimensional estimated subspace of *DB-World subject and bodies* datasets by the proposed method (hrKCCA+kNN and hrKCCA+SVM_L) other exiting methods. 93
5.7 Recognition rate (%) for *KTH* dataset (all scenarios) by the proposed method (hrKCCA+kNN and hrKCCA+SVM_L) and other methods. 94
5.8 Recognition rate (%) for *UMD* dataset by the proposed method (hrKCCA+kNN and hrKCCA+SVM_L) and some of the best stat-of-the-art methods of this dataset. 95

List of Figures

2.1 Background of the kernel in the 20th century. 16

3.1 Individual reproducing kernel Hilbert space for each kernel of the kernel principal component analysis (KPCA). 44

3.2 Scatter plots of the first two kernel principal components for *wine* data: Gaussian RBF kernel is used in the top panel (a) $s = 0.05$ (b) $s = 0.75$ (c) $s = 1$ (d) $s = 10$, and polynomial kernel in the bottom (e) $c = 0.001$, $d = 2$ (f) $c = 10$, $d = 2$ (g) $c = 1$, $d = 3$ (h) $c = 1$, $d = 4$. 45

3.3 Architecture of kernel choice in kernel PCA. 48

3.4 Algorithm of kernel choice in kernel PCA with Gaussian RBF kernel. . . . 49

3.5 Kernel PCA for synthesized data-1 (top) and synthesized data-2 (bottom). (a) Scatter plot for the two variables of a sample. (b) Box plots of the leave-one-out cross validation (LOOCV) reconstruction errors for 100 samples. (c, d) scatter plots of the first two kernel principal components using (c) the best inverse kernel widths ($s = 5, 10$) and (d) larger bandwidths $s = 50, 200$. 51

3.6 Visualization of the first two kernel principal components of *food* data (a) $s = 0.001$, (b) $s = 0.5$ and (c) $s = 200$ 53

4.1 The system of CCA. 60

4.2 The system of kernel CCA. 63

4.3 Box plots of canonical correlation coefficient of five estimators using population canonical correlation, PCC = 0.79 and PCC = 0.63. 67

4.4 Scatter plots of (X_1, Y_1) (top left) and first canonical variates 68

4.5 Scatter plots of (X_2, Y_2) (top left) and 2nd canonical variates 69

4.6 Box plots of sensitivity value over 200 samples a) multivariate normal data and b) transform multivariate normal data. 71
4.7 Breakdown plots for first canonical correlation coefficient. 73

5.1 Standard kernel CCA: the scatter plots of the 1st canonical variates for the nutrimouse dataset (liver cells and hepatic fatty acids) using the Gaussian RBF kernel with eight inverse bandwidths s and fixed regularization coefficient $\kappa = 10^{-4}$. The 10-fold cross-validation errors are also embedded. . . 78
5.2 Algorithm of the higher-order regularized kernel CCA. 83
5.3 Scatter plots of 1st kernel canonical variates for the examples ($E_1 - E_3$). The first column for the standard kernel CCA. The final three columns for the hrKCCA, using different trade-off c for the regularization parameters: $\nu = c\lambda$. The inverse bandwidth s and the 4th moment regularization coefficient λ are chosen by the CV. 96
5.4 Box plots and line plots (inset) using mean values of cross-validation errors of 100 samples for example 3 (bandwidths, $s_1 = 225, s_2 = 250, s_3 = 275, s_4 = 300, s_5 = 325, s_6 = 350, s_7 = 375, s_8 = 400$). 97
5.5 Scatter plots of the 1st kernel canonical variates given by the proposed method for the nutrimouse dataset (liver cells and hepatic fatty acids) using the Gaussian RBF kernel with eight inverse bandwidths s and fixed regularization coefficient $\lambda = 0.75$. The 10-fold cross-validation error is also embedded (see also Table 5.4). 98
5.6 Scatter plots of the first canonical variates of real datasets (Email (D_2), Psychological (D_3) and Carbig (D_4)) using the parameters chosen by CV for the kernel CCA (a) and the proposed method (b). 99
5.7 Scatter plots of the first canonical variates ($a(i)$) -($d(i)$) and the first two canonical variates of the exploratory variables ($a(ii)$ -$d(ii)$) for the *wine* dataset. The proposed method ($s = 0.05, \lambda = 0.1$) in (a) and kennel CCA using three heuristic bandwidths ($s_1 = 0.02, s_2 = 0.073, s_3 = 0.05$) in (b - d) are shown. 100

5.8 Scatter plots of the first canonical variates (upper row) and one dimensional index plots (lower row) given by the proposed method for *DBWorld subject, bodies, BUPA liver disorders*, and *diabetes.* 101
5.9 Scatter plots of the first canonical variates (upper row) and the first two canonical variates of **X** (lower row) using *KTH* dataset (outdoor scenario only) for the kernel CCA. 102
5.10 Scatter plots of the first canonical variates (upper row) and first two canonical variates of **X** (lower row) using *KTH* dataset (outdoor scenario only) for the proposed hrKCCA. 103
5.11 Scatter plots of the first canonical variates (upper row), first two canonical variates of **X** (middle row) and confusion matrices (lower row) using *KTH* dataset for all scenarios (boxing (B), hand clapping (HC), hand waving (HW), jogging (J), running (R), and walking (W)) for the for the proposed hrKCCA. 104

ACKNOWLEDGMENTS

First and foremost, I would like to pay my thanks to my adviser, Professor ***Kenji Fukumizu***, The Institute of Statistical Mathematics (ISM) and The Graduate University of Advanced Studies (SOKENDAI). I owe a great deal of gratitude to him for his guidance, financial support and help throughout the exciting journey full of learning and growth at my doctoral courses and research in Japan.

I would like to extend my thanks to the faculty members, especially Professor *Shinto Eguchi*, Professor *Satoshi Ito*, Dr. *Yoichi Nishiyama* who taught me during the doctoral courses. It's my pleasure to thank to Dr. *Osamu Komori*. He served me as a family member in the last five years in Japan. I will keep them my mind always who had assisted me cordially in getting relevant literatures and other necessary advice.

I acknowledge to the staffs of ISM and research Cooperation Unit (Kenkyo) for their cordial help and co-operation during my research work. I am very grateful to Dr. *Takao Kumazawa* for his excellent guidance and sightseeing in Tokyo during first two years.

I would like to express my heartfelt gratitude to all Fukumizu's lab members and friends who have supported me through the years and have made my time at ISM so enjoyable. In particular, I would like to thank Dr. *Akifumi Notsu, Yuta Tanoue, Hisaki Ikebata and Katsuhiro Omae* with whom I share many unforgettable memories.

One thing I must confess, without the encouragement and care all though my life, of my parents ***Md. Shamsul Alam Akando*** and ***Mst. Helena Alam*** would be quite impossible to continue the doctoral course and research. I would like to thank all of my family members and relatives. My special thanks go my M.Sc. adviser, Professor *Mohammed Nasser* University of Rajshahi, who encourages a lot to begin my research.

At last, my deepest gratitude and love, of course, belongs to my wife, *Mst. Shajia Afrin*, my daughter, *Asfia Alsaba* and my son, *Saffat Ashad* for their unconditional love and help. They are one of my main sources of inspiration.

ABSTRACT OF THE DISSERTATION

by

Md. Ashad Alam

in

Department of Statistical Science

The Institute of Statistical Mathematics

The Graduate University of Advanced Studies

Tokyo 190-8562, Japan.

In kernel methods, choosing a suitable kernel is indispensable for favorable results. While cross-validation is a useful method of the kernel and parameter choice for supervised learning such as the support vector machines, there are no well-founded methods, have been established in general for unsupervised learning. We focus on kernel principal component analysis (kernel PCA) and kernel canonical correlation analysis (kernel CCA), which are the nonlinear extension of principal component analysis (PCA) and canonical correlation analysis (CCA), respectively. Both of these methods have been used effectively for extracting nonlinear features and reducing dimensionality.

As a kernel method, kernel PCA and kernel CCA also suffer from the problem of kernel choice. Although cross-validation is a popular method of choosing hyperparameters, it is not applicable straightforwardly to choose a kernel and the number of components in kernel PCA and kernel CCA. It is important, thus, to develop a well-founded method for choosing hyperparameters of the unsupervised methods.

In kernel PCA, it is not possible to use cross-validation for choosing hyperparameters because of the incomparable norms given by different kernels. The first goal of the dissertation is to propose a method for choosing hyperparameters in kernel PCA (the kernel and the number of components) based on cross-validation for the comparable reconstruction errors of pre-images in the original space. The experimental results of synthesized and real-world datasets demonstrate that the proposed method successfully selects an appropriate kernel and the number of components in kernel PCA in terms of visualization and classification errors on the principal components. The results imply that the proposed method enables the automatic design of hyperparameters in kernel PCA.

In recent years, the influence function of kernel PCA and a robust kernel PCA has been theoretically derived. One observation of their analysis is that kernel PCA with a bounded kernel such as Gaussian is robust in that sense the influence function does not diverged, while for kernel PCA with unbounded kernels for example polynomial the influence function goes to infinity. This can be understood by the boundedness of the transformed data onto the feature space by a bounded kernel. While this is not a result of kernel CCA but for kernel PCA, it is reasonable to expect that kernel CCA with a bounded kernel is also robust. This consideration motivates us to do some empirical studies on the robustness of kernel CCA. It is essential to know how kernel CCA is effected by outliers and to develop measures of accuracy. Therefore, we do intend to study a number of conventional robust estimates and kernel CCA with different functions but fixed parameter of kernel.

The second goal of the dissertation is to discuss five canonical correlation coefficients and investigate their performances (robustness) by influence function, sensitivity curve, qualitative robustness index and breakdown point using different type of simulated datasets.

The final goal of the dissertation is to extract the limitations of cross-validation for the kernel CCA, and to propose a new regularization approach to overcome the limitations of kernel CCA. As we demonstrate for Gaussian kernels, the cross-validation errors for kernel CCA tend to decrease as the bandwidth parameter of the kernel decreases, which provides inappropriate features with all the data concentrated in a few points. This is caused by the ill-posedness of the kernel CCA with the cross-validation. To solve this problem, we propose to use constraints on the 4th order moments of canonical variables in addition to the variances. Experiments on synthesized and real world datasets including human action recognition for a robot demonstrate that the proposed higher-order regularized kernel CCA can be applied effectively with the cross-validation to find appropriate kernel and regularization parameters.

Chapter 1

Introduction

Methods using positive definite kernel (PDK), *kernel methods* play an increasingly prominent role to solve various problems in statistical machining learning such as, web design, pattern recognition, human action recognition for a robot, computational protein function perdition, remote sensing data analysis and in many other research fields. Due to the kernel trick and reproducing property, we can use linear techniques in feature spaces without knowing explicit forms of either the feature map or feature spaces. It offers versatile tools to process, analyze, and compare many types of data and offers state-of-the-art performance.

Nowadays, PDK has become a popular tool for the most branches of statistical machine learning e.g., supervised learning, unsupervised learning, reinforcement learning, non-parametric inference and so on. Many methods have been proposed to kernel methods, which include support vector machine (SVM, Boser et al., 1992), kernel ridge regression (KRR, Saunders et al., 1998), kernel principal component analysis (kernel PCA, Schölkopf et al., 1998), kernel canonical correlation analysis (kernel CCA, Akaho, 2001, Bach and Jordan, 2002), Bayesian inference with positive definite kernels (kernel Bayes' rule, Fukumizu et al., 2013), gradient-based kernel dimension reduction for regression (gKDR, Fukumizu and Leng, 2014), kernel two-sample test (Gretton, 2012) and so on.

During the last decade, unsupervised learning has become an important application area of the kernel methods. There are two most powerful tools of unsupervised kernel methods, namely kernel principal component analysis (kernel PCA) and kernel canonical correlation analysis (kernel CCA) (Schölkopf et al., 1998, Akaho, 2001). Using these two methods, we are able to extract effective nonlinear features by high dimensional embedding of data

based on the reproducing kernel Hilbert space (RKHS). They have also closed connection with many unsupervised dimensional reductions and manifold learning techniques (Izenman, 2008, Chapter 16). Kernel PCA and kernel CCA have been applied in different areas of statistical machine learning such as reduction of dimensionality, image processing, feature extraction, de-noising, statistical shape analysis, novelty detection, pre-processing of regression and classification (Mika et al., 1999, Kwok and Tsang, 2003, Arias et al., 2007, Zheng et al., 2010, Hardoon et al., 2004, Huang et al., 2009a, Alzate and Suykens, 2008). In all of the above areas, kernel PCA and kernel CCA have been used with an arbitrary choice of the kernel and the number of features. The results are in fact very sensitive to both the choice of the kernel and the number of features of RKHS. As a kernel method, kernel PCA and kernel CCA also suffer from the problem of kernel choice. To the best of our knowledge, however, a well-founded technique for choosing the parameters has not yet been established.

The cross-validation (CV) approach is popularly used for choosing parameters of kernel methods, such as the bandwidth parameter in Gaussian kernel, especially in supervised learning. For SVM, the cross-validation is one of the most popular and useful ways of choosing the kernel and parameters (Arlot, 2010, Woen and Perry, 2009, Stone, 1974). In the case of standard linear PCA, the algorithm can be formulated as minimization for self-regression with reduced rank, and cross-validation approaches have been proposed for choosing the number of components (Krzanowski, 1987, Wold, 1978). The k-fold CV has been used for classical CCA (Liang, 1995). Note that it is not possible to apply the leave-one-out CV (LOOCV) with canonical correlation value, since the correlation is not computable with one data. While the CV with the canonical correlation value has been used for choosing the bandwidth in kernel CCA (Suetani et al., 2006), where very dense data from a chaotic dynamics are discussed, it is not easy in general to obtain reliable canonical correlation values by the k-fold CV for small data points.

The result of kernel PCA obviously depends on the choice of the kernel. It is often the case that the kernel has some parameters like the popular examples shown in Table 2.1. In such a case, these parameters may have a strong influence on the results. To depict the influence, using *wine* dataset we show the plots of the first two kernel principal components with different values of inverse-bandwidth parameters in the Gaussian RBF kernel, and

degree and constant in the polynomial kernel (see Section 3.3 and Figure 3.2). From the figures, we see that in both the kernels the results of kernel PCA depend strongly on the parameters, and an appropriate choice is indispensable for the method to give reasonable low-dimensional representation of data.

In kernel CCA with Gaussian RBF kernel, we need to select a proper inverse bandwidth and a regularization parameter. It is also well known each hyperparameter has the influence on the result of kernel CCA. A guideline to select the regularization parameter has been proposed by Hardoon et al. (2004) and a heuristic technique has been also used for choosing the bandwidth (Hardoon and Shawe-Taylor, 2009).

It is known that CCA and kernel CCA can be regarded as an alternating regression problem (Breiman and Friedman, 1985, Shawe-Taylor and Cristianini, 2004). CV using correlations or prediction error depends on the data concentration. When the data are concentrated in only a few extreme points with perfect or nearly perfect correlations, on the one hand, the CV error will be very small which satisfies the objective of kernel CCA (maximum correlation of canonical variate) but on the other hand, the canonical variates do not follow any well-posed distribution. In classification problems, we can use CV based on classification rates, but the smallest classification error does not correspond to the high correlated features of kernel CCA, in general.

For kernel CCA, however, the cross-validation approach based on the prediction error does not necessarily choose a good parameter in general. We demonstrate this problem using an example of the nutrimouse dataset for the Gaussian RBF kernel (see the Chapter 5). In Figures 5.1 we show eight scatter plots of the first canonical variates using eight inverse bandwidths together with the cross-validation errors. As we see, the larger value of inverse bandwidth provides smaller error (≈ 0), but the solution are ill-posed: high correlation is achieved by the features with most data concentrating on a few points. This example illustrates that a straightforward application of cross-validation for choosing a kernel is not appropriate. It is expected that the variance constraints on the kernel CCA do not regulate sufficient for a large variety of nonlinearity given by different kernels.

Although cross-validation is a popular method for choosing hyperparameters, it is not applicable straightforwardly to choose a kernel of kernel PCA and kernel CCA. It is important, thus, to develop a well-founded method of designing a kernel of unsupervised

kernel methods. The main goal of the dissertation is to design a kernel and robustness for unsupervised kernel methods.

1.1 Designing kernel for unsupervised kernel methods

Selection of hyperparameters (kernel and the number of features) of kernel PCA is not a straightforward task because each kernel provides an individual norm in corresponding feature space. So, we are not able to choose them based on a performance with the norms of the feature spaces. In Chapter 3, we propose a method for choosing an optimum kernel and the number of features through the LOOCV based on pre-image performance, which is an approximate inverse image of a point in the feature space. To this ends, we extract the pre-image of a test point projected onto the subspace in RKHS using a pre-image method of each feature space. We then evaluate the reconstruction error based on this pre-image. As in the leave-one-out cross-validation, each data set is regarded as a test point and the average error is computed. The kernel and the number of features corresponding to the minimum error is chosen as the optimum ones.

As the kernel PCA, selection of the kernel and the number of features for kernel CCA is also a challenging problem. We are not able to extract the kernel CCA fruitfully by LOOCV. Canonical correlation analysis is the generalization of regression analysis with multiple response variables. So, it is possible to use mean square error like as regression analysis to apply LOOCV. By this result, we have observed that for small bandwidth of Gaussian RBF kernel, the most of data points are accumulated but for a few extreme points. In this situations, the CV error is very small. Moreover, the CV values highly to depends on bandwidth in finite samples.

Kernel CCA is given by the second order statistics (e.g., variance) of the canonical variates; this would suffice for a complete statistical description of a Gaussian distribution, but not in general. With the rich function classes given by positive definite kernels, we need much stronger constraint to regulate the canonical variate sufficiently to make the cross-validation applicable. The kernel CCA subject to the higher-order constraints is proposed to select the tuning parameters using the cross-validation technique. We demonstrate the effectiveness of the proposed higher-order regularized kernel CCA, combined with the

cross-validation, in measuring the relationship and extracting effective features for classification using various synthesized and real world problems (see Chapter 5).

1.2 Robustness of unsupervised kernel methods

The influence function (IF) of kernel PCA has been theoretically well-known. Kernel PCA with a bounded kernel such as Gaussian is robust in the sense that the influence function does not diverge, while it is not robust with unbounded kernels for example polynomial (IF goes to infinity). This can be understood by the boundedness of the transformed data onto the feature space by a bounded kernel (Huang et al., 2009b, Suykens et al., 2010). While this is not a result of kernel CCA, it is reasonable to expect that kernel CCA with a bounded kernel is also robust. This consideration motivates us to do some empirical studies on the robustness of kernel CCA. It is essential to know how kernel CCA is affected by outliers and to develop measures of accuracy. Therefore, we intend to study a number of conventional robust estimates and kernel CCA with different functions, but fixed bandwidth of Gaussian RBF kernel and Laplacian kernel (see Chapter 4).

1.3 Scope of the research

Kernel PCA and kernel CCA are most useful and fundamental unsupervised statistical pattern recognition techniques as well as preprocessing techniques for supervised learning: regression analysis and classification analysis to predict from massive amounts of data. Both of these techniques are also used in unsupervised learning to extract significant and anomaly structure of high dimensional datasets. Kernel CCA is also used as a constrast function of independent component analysis, an important method of unsupervised learning. An additional important application area of kernel CCA is stochastic processes to learn the dynamic nature of the processes. The proposed two methods provide a way of automatic findings the hyperparemtes of the kernel PCA and kernel CCA. It is expected that our research would be a great impact on the area of statistical machine learning.

1.4 Outline and summary of the contributions

Chapter 1: The motivation of designing the kernel and robustness for unsupervised kernel methods are presented in the first chapter.

Chapter 2: A review of important and useful results of functional analysis and PDK is given in this chapter.

Chapter 3: This chapter discusses the limitations of kernel PCA and provides a new method for choosing hyperparameters in kernel PCA (kernel and the number of components) based on CV for the comparable reconstruction errors of pre-images in the original space. The proposed method has been applied to synthesized and a number of real world datasets.

Chapter 4: This chapter treats the performances of five canonical correlation coefficients in the different types of distribution using influence function, sensitivity curve, qualitative robust index and breakdown point.

Chapter 5: This chapter discusses the CV for kernel CCA with its limitations, and provides a new 4th order moment regularization approach for kernel CCA. The proposed method demonstrates for synthesized and real world problems, including human action recognition for a robot.

Chapter 6: A conclusion and further aspects of research are discussed in the final chapter.

Chapter 2

Preliminary of Functional Analysis and Kernel

Functional analysis is an abstract branch of mathematics where the vector spaces are in general infinite dimensional and not all operators on them can be represented by matrices unlike the linear algebra. An abstract approach starts with a set of elements satisfying certain axioms. For example, in linear algebra we use it in connection with fields, rings and groups, but in functional analysis it in connection with *abstract spaces* (inner product space, Hilbert spaces, Banach spaces etc.,). An *abstract space* is a set (unspecified) of elements satisfying certain axioms. The different sets of axioms provide different abstract spaces.

Functional analysis has a history of more than one century and nowadays its results have been used in a number of areas: mathematics, statistics, robust statistics and statistical machine learning. Some basic definitions and results are summarized in this chapter, which are frequently used to reproducing kernel Hilbert space (RKHS) and its methods. The fundamental references of this chapter are: Hille (1972), Reed and Simon (1980), Kreyszig (1989), Cucker and Smale (2002), Schölkopf and Smola (2002), Berlinet and Thomas-Agnan (2004), Shawe-Taylor and Cristianini (2004), Bishop (2006), Steinwart and Christmann (2008), Hofmann et al. (2008), Gärtner (2008), Fukumizu et al. (2009), King (2009), Fasshauer (2011), Steinwart and Scovel (2012) and Fukumizu et al. (2013).

2.1 Inner product space

In calculus, the notions of convergence and continuity can be formulated in terms of distance between two numbers or between two vectors, where the objects are very simple ($\mathbb{R}$ or $\mathbb{R}^n$). In functional analysis, we consider more general spaces, which contain more complicated objects than numbers and vectors e.g., functions and so on. To find a distance between two complicated objects, we need a new notion of distance that lead a new notion of convergence and continuity. These again lead to new arguments surprisingly similar to those we have already seen in calculus. Now the question, is it possible to develop a general theory of distance where we can prove the results (we need once and for all)? We have a positive answer and the theory is called the theory of *metric spaces*. By inducing general conditions on the distance function, we are able to develop a general notion of distance, which can be applicabled even on complicated objects. Using metric space theory, we can formulate and prove results about convergence and continuity once and for all.

Metric space is a set with a metric on it. We can generalize it by a notion of nearness (open set is sufficient). Simply imagine small open balls around a point to measure nearness without mentioning distances, which leads to the idea of a *topological space*. The fundamental notion of *topological space* is based on the collection of all open sets (complement is closed set) instead of metric. The field of topology is an abstract study that evolved as an independent discipline in response to certain problems of classical analysis and geometry. It provides a unifying theory that can be used in many diverse branches of mathematics.

The metric and topology are defined on an abstract set, which may or may not be a vector space. To make a relation between algebraic and geometric properties of an abstract set, we need to define a metric in the special form, say *norm*. A norm is nothing but a metric on the vector space, which is combined the algebraic structure and metric concepts. It can be given more useful and important metric spaces (Kreyszig, 1989).

In a normed space we can do vector addition and scalar multiplication of a vector just like the elementary vector algebra, but still we cannot do two useful operations: vector dot product and orthogonality of two vectors. It is possible to fill this gap using the inner product. A vector space together with an inner product is called an *inner product space*.

Definition 2.1.1 (Inner product space) *Let $\mathcal{V}$ is a vector space (real or complex). The*

inner product of the $\mathcal{V}$ is a continues mapping $\langle , \rangle : \mathcal{V} \times \mathcal{V} \to \mathbb{F}$, that has the following four properties, where $\mathbb{F}$ is a scaler field of $\mathcal{V}$:

CI1 (Positive definite)

$$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0, \qquad \langle \mathbf{x}, \mathbf{x} \rangle = 0 \Leftrightarrow \mathbf{x} = 0.$$

CI2 (Anti-symmetric (Hermitian))

$$\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}, \qquad \forall\, \mathbf{x},\, \mathbf{y} \in \mathcal{V}.$$

For a real vector space (Symmetric).

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle, \qquad \forall\, \mathbf{x},\, \mathbf{y} \in \mathcal{V}.$$

CI3 (Homogeneity)

$$\langle a\mathbf{x}, \mathbf{y} \rangle = a\langle \mathbf{x}, \mathbf{y} \rangle, \qquad \forall a \in \mathbb{F}.$$

CI4 (Cumulative (addition))

$$\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle, \qquad \forall \mathbf{x},\, \mathbf{y},\, \mathbf{z} \in \mathcal{V}.$$

The pair $(\mathcal{V}, \langle , \rangle)$ is called inner product space.

A metric is a measure of how different the elements of a set are. A norm is a measure of how large the element is. An inner product is a measure of what degree the elements are linearly independent.

A useful property of inner products is the Cauchy-Schwarz inequality,

$$\langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}} \langle \mathbf{y}, \mathbf{y} \rangle^{\frac{1}{2}}. \tag{2.1}$$

This relationship is also sometimes called the Cauchy-Bunyakovskii-Schwarz inequality.

2.2 Hilbert space

Hilbert spaces are possibly-infinite-dimensional analogues of the finite-dimensional Euclidean spaces familiar to us. In particular, Hilbert spaces have inner products, so notions of perpendicularity (or orthogonality) and orthogonal projection are available. Reasonably enough, in the infinite-dimensional case we must be careful not to extrapolate too far based only on the finite-dimensional case. Perhaps strangely, few naturally-occurring spaces of functions are Hilbert spaces.

Definition 2.2.1 (Hilbert space) *A Hilbert space is a real or complex inner product space that is complete under the inner product.*

Example 2.2.1 (Sequence spaces) *The space ℓ^2 is a Hilbert space (real valued) with inner product defined as*

$$\langle x, y\rangle = \sum_{i=1}^{\infty} \xi_i \nu_i, \qquad x, y \in \ell^2.$$

Example 2.2.2 *Let $\mathcal{H}$ is a finite dimensional real vector space of functions with basis $(f_1, f_2, \cdots, f_n)$. Any vector of $\mathcal{H}$ can be defined as a linear combination of $(f_1, f_2, \cdots, f_n)$ in unique way. An inner product $\langle , \rangle_{\mathcal{H}}$ on $\mathcal{H}$ is define by the numbers*

$$k_{ij} = \langle f_i, f_j\rangle, \qquad i, j = 1, 2, 3, \cdots, n.$$

If $u_1 = \sum_{i=1}^{n} u_{1i} f_i$ and $u_2 = \sum_{j=1}^{n} u_{2j} f_j$, then

$$\langle u_1, u_2\rangle_{\mathcal{H}} = \langle \sum_{i=1}^{n} u_{1i} f_i, \sum_{j=1}^{n} u_{2j} f_j\rangle_{\mathcal{H}} = \sum_{i=1}^{n}\sum_{j=1}^{n} u_{1i} u_{2j} k_{ij}.$$

The matrix $K = (k_{ij})$ is called the Gram matrix of the basis. It is a positive definite matrix. A finite dimensional space endowed with any inner product is always complete and therefor it is a Hilbert space.

Example 2.2.3 (Lebesgue spaces) *Given μ a Lebesgue measure on the set $\mathbb{R}$ of real numbers and $\mathcal{L}(a, b)$ ($-\infty \le a \le b \le \infty$, abstract set) be the set of all complex measurable*

function over (a, b) *such that*

$$\int_a^b |f(t)|^2 \mathrm{d}\mu(\mathrm{t}) < \infty.$$

Identifying two functions f_1 *and* f_2 *of* $\mathcal{L}(a, b)$ *which are equal expect on a set of Lebesgue measure equal to zero, we get a Hilbert space,* $\mathrm{L(a, b)}$ *with inner product*

$$\langle f_1, f_2 \rangle_{\mathrm{L(a,b)}} = \int_a^b f_1(t)\overline{f_2(t)}\mathrm{d}\mu(\mathrm{t}).$$

Two (or more) Hilbert spaces can be combined to produce another Hilbert space by taking either their direct sum or their tensor product (new Hilbert spaces from old).

Definition 2.2.2 (Orthogonal complement) *Let* $\mathcal{H}$ *is a Hilbet space and* $\mathcal{V}$ *of be a closed subspace* $\mathcal{H}$ *then*

$$V^{\perp} : \left\{ \mathbf{x} \in \mathcal{H} \,\middle|\, \langle \mathbf{x}, \mathbf{y} \rangle = 0,\ \forall \mathbf{y} \in V \right\}$$

is closed subspace and called the orthogonal complement of $\mathcal{V}$*.*

Definition 2.2.3 (Orthogonal projection) *Let* $\mathcal{H}$ *is a Hilbert space and V be a closed subspace. Every* $\mathbf{x} \in \mathcal{H}$ *can be uniquely decomposed*

$$\mathbf{x} = \mathbf{y} + \mathbf{z}, \qquad \mathbf{y} \in \mathcal{V},\ \mathbf{z} \in \mathcal{V}^{\perp} \text{ i.e.}, \mathcal{H} = \mathrm{V} \oplus \mathrm{V}^{\perp},$$

which is called orthogonal projection.

Definition 2.2.4 (Complete orthonormal system) *A subset* $\{v_i\}_{i \in \mathbf{I}}$ *of* $\mathcal{H}$ *is called an orthonormal system (ONS) if* $\langle v_i, v_j \rangle = \delta_{ij}$ *(*δ_{ij} *is Kronecker's delta). It is called complement orthonormal system (CONS) or orthonormal basis if it is ONS and if*

$$\langle \mathbf{x}, \mathbf{v}_j \rangle = 0, \qquad (\forall i \in \mathbf{I}) \Rightarrow \mathbf{x} = 0, \qquad \forall \mathbf{x} \in \mathcal{H}.$$

Facts 2.2.1 *Any ONS in a Hilbert space can be extended to a CONS.*

Definition 2.2.5 (Separable Hilbert space) *A Hilbert space is separable if it has a countable CONS.*

Theorem 2.2.1 (Fourier series expansion, (Kreyszig, 1989)) *Let* $\{\mathbf{v}_i\}_{i=1}^{\infty}$ *is a CONS of a separable Hilbert space. For each* $\mathbf{x} \in \mathcal{H}$*,*

$$\mathbf{x} = \sum_{i=1}^{\infty} \langle \mathbf{x}, \mathbf{v}_i \rangle \mathbf{v}_i \qquad \textit{(Fourier expansion)}$$

$$\|\mathbf{x}\|^2 = \sum_{i=1}^{\infty} |\langle \mathbf{x}, \mathbf{v}_i \rangle|^2 \qquad \textit{(Parseval's equality).}$$

For a general orthonormal system, the Parseval's equality becomes a Bessel inequality

$$\|\mathbf{x}\|^2 \geq \sum_{i=1}^{\infty} |\langle \mathbf{x}, \mathbf{v}_i \rangle|^2.$$

2.3 Feature space and its drawback

In the beginning (1980s) of statistical machine learning the main goal was not inference (estimation) but generalization or a good "predictive" capability Hastie et al. (2009), Schölkopf and Smola (2002). However, some nonparametric inference methods have been also developed for inference Fukumizu et al. (2013), Gretton (2012), and so on. For generalization or a good prediction, We seek to arrive at a function

$$f : \mathcal{X} \to \mathcal{Y}$$

based on training data, $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots (\mathbf{x}_n, \mathbf{y}_n)\}$. The goal, a given test set $\{\mathbf{x}'_i\}_{i=1}^{t}$, arrive at "good" predictions $\{\mathbf{y}'_i\}_{i=1}^{t}$ via f. Define the hypothesis space $\mathcal{H}$ as the space of functions to consider for f. The learning problem can then be summarized as finding a method L that maps a training set to a function of the hypothesis space:

$$L : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}.$$

Define an unknown probability measure on $\mathcal{X}$ and $\mathcal{Y} : \mu(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y})$ but assume that the training samples $\{(\mathbf{x}_i, \mathbf{y}_i)\} \sim p(\mathbf{x}, \mathbf{y})$ is independent and identically distributed (iid).

We need some measure of "similarity" between elements of the input space $\mathcal{X}$ via inner

product space (tractable learning problem). For this purpose, we need a function that takes any pair of values $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{X}$ a real value returns their "similarity":

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R},\ (\mathbf{x}, \mathbf{x}') \to k(\mathbf{x}, \mathbf{x}'),$$

where k refers to as a kernel. This allows us to make statements about how similar the outputs $\mathbf{y}$ with new $\mathbf{y}'$. Similarity measure by inner product

$$\begin{aligned} \langle \mathbf{x}, \mathbf{z} \rangle &= \text{Large} \qquad \text{if } \mathbf{x} \text{ and } \mathbf{z} \text{ similar} \\ &= \text{Small} \qquad \text{if } \mathbf{x} \text{ and } \mathbf{z} \text{ different} \end{aligned}$$

For example, Good measure how similar $\mathbf{x}$, $\mathbf{z}$

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2s^2}\|\mathbf{x}-\mathbf{x}'\|^2}; \quad k(\mathbf{x}, \mathbf{x}') = \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{||\mathbf{x}||||\mathbf{x}'||}.$$

We have to seek a space that incorporates inner product by map $\mathcal{X}$ into a new space with some additional structure. Do linear techniques work in nonlinear inputs space? The answer is negative.

Definition 2.3.1 (Input space) *The space of all original data is called input space. The representation space is the set of all possible object descriptions that can occur in a given problem.*

The following basic linear techniques do not work well in nonlinear input space:

- Correlation analysis,
- Linear regression analysis,
- Fisher discriminant analysis,
- Principal component analysis (PCA),
- Canonical correlation analysis (CCA) and so on.

So we need to seek a new space for which linear techniques are worked well, say feature space.

2.3.1 Feature space

Definition 2.3.2 (Feature space) *Let $\mathcal{X}$ is a set and* $\mathbf{x} \in \mathcal{X}$

$$\Phi := \mathcal{X} \to \mathcal{H},\ \mathbf{x} \to \Phi(\mathbf{x})$$

is a feature map *and the vector* $\Phi(\mathbf{x})$ *in* $\mathcal{H}$ *is called* feature vector. *The space* $\mathcal{H}$ *for all functions via* Φ *is called feature space. It is a finite or infinite dimensional vector space.*

For example, let $\mathcal{X} \in \mathbb{R}$ be the variable for the duration of earthquake in last two months.

$$X \to \begin{bmatrix} \phi_1(X) \\ \phi_2(X) \\ \phi_3(X) \\ \phi_4(X) \end{bmatrix} = \begin{bmatrix} X \\ X^2 \\ X^3 \\ X^4 \end{bmatrix} = \Phi(X), \quad \phi_i\text{'s are the nonlinear maps.}$$

The map $\mathcal{X} \to \mathcal{H}$ not for only similarity, but also has more features:

- induce a geometric structure that we can leverage (deal with the patterns geometrically),
- linearizing of nonlinear space (linear, but nonlinear in input space),
- the freedom to choose the mapping Φ will enable us to design a large variety of learning algorithms.

The main argument is that any low-dimensional structure may be more easily discovered when it becomes embedded in the larger space $\mathcal{H}$, which could be infinite dimensional space.

2.3.2 Computational problems of feature space

For high dimensional feature vector $\Phi(\mathbf{X})$, we are not able to compute the inner product $\langle \Phi(\mathbf{X}), \Phi(\mathbf{X}) \rangle$. For example, simple, bivariate case up to quadratic transforms

$$\mathbf{X} = (X, Y) \to (X, Y, X^2, Y^2, XY)$$

We have five variables to use linear method ($\mathbb{R}^5$). If we consider 3 variables and higher oder up to cubic transforms

$$\mathbf{X} = (X, Y, Z) \rightarrow (X, Y, Z, X^2, Y^2, Z^2, XY, XZ, YZ, X^2Y, X^2Z, XYZ \ldots).$$

If the input vector $\mathbf{X}$ is 100 dimensional and the moments up to the 4th order are used

$$\binom{100}{1} + \binom{100}{2} + \binom{100}{3} + \binom{100}{4} = 4087975 = \dim(\mathcal{H}).$$

Computational cost will be expensive on the inner product of the feature space and impossible for infinite dimension.

2.4 Kernel and positive definite kernel

2.4.1 Kernel

The term kernel has a long history. The meaning of the kernel depends on the subjects that has different meanings in different literatures. In mathematics, the term kernel itself is used with different meanings, for example, in linear algebra where it is used as a synonym for the null space. In the beginning of the last century David Hilbert and other researches, the term kernel has been used as a bivariate function to the field integral operators. The term kernel has been also used for density estimation in the statistical literature, where the kernel $K : \mathbb{R} \rightarrow \mathbb{R}$ is an integrable function satisfying $\int K(x)\,\mathrm{dx} = 1$.

The term kernel has been also used as a positive definite kernel (PDK) in the branch of mathematics since 1950s. The PDK is a large class of kernels, which contains Mercer kernel and so on. It can be regarded as a generalized dot product. Figure 2.1 presents the historical background of the kernel in the 20th century.

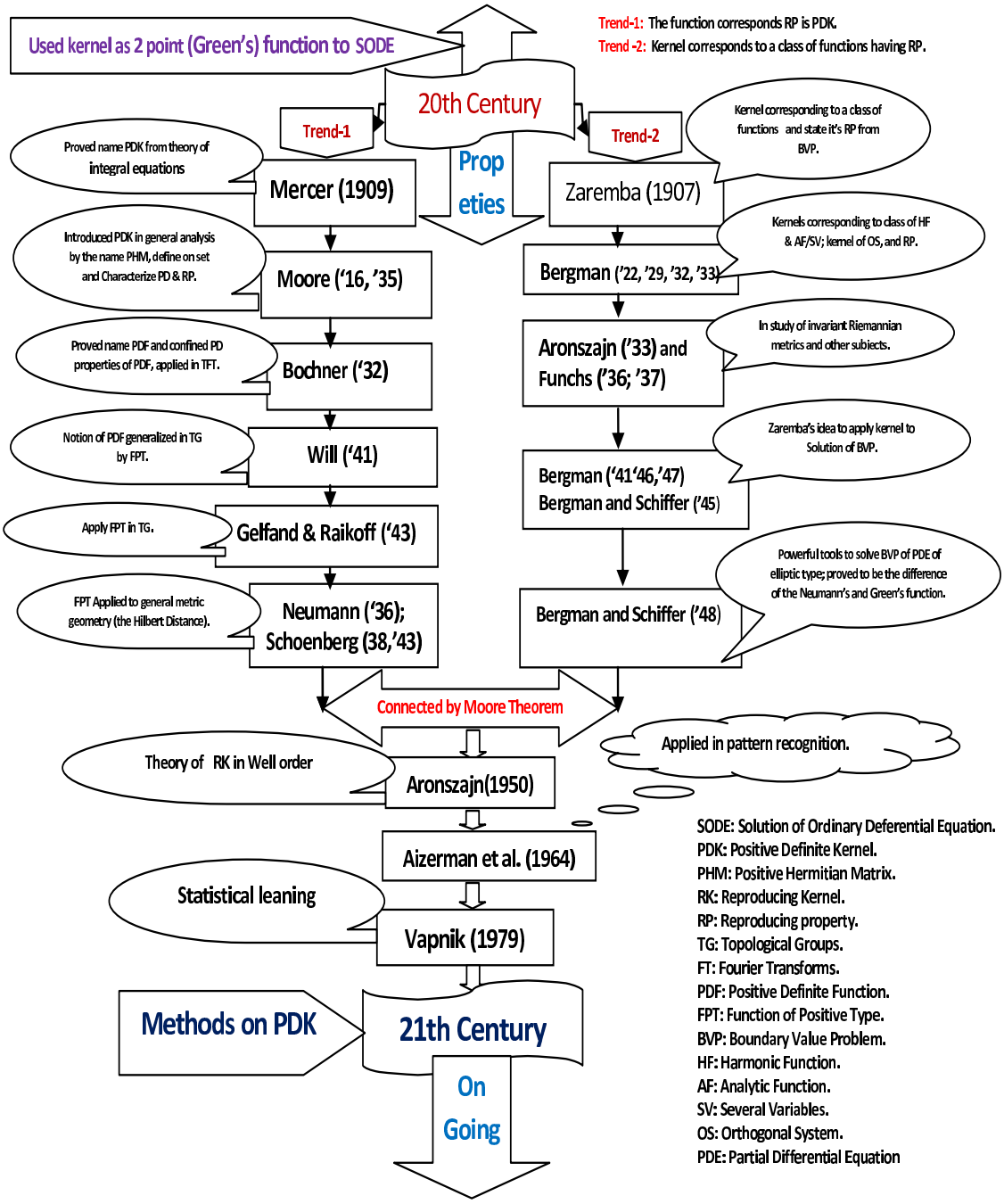

Figure 2.1: Background of the kernel in the 20th century.

Definition 2.4.1 (Kernel) *Let $\mathcal{X}$ be a non-empty arbitrary set. A bivariate function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called kernel on $\mathcal{X}$ if there exists a Hilbert space, $\mathcal{H}$ and a feature map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we have*

$$k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle.$$

In general the kernel may be also defined on the complex field. So, Hilbert space may be also a complex valued Hilbert space.

Algorithms can be applied in dot product space by considering the kernel as a similarity measure. We can construct a kernel of several ways as follows:

- straightforward construction of feature map and space,
- using a sequence of functions $(f_n : \mathcal{X} \to \mathbb{F}, n \in \mathbb{N}$ such that $(f_n(\mathbf{x}))_{n=1}^{\infty}) \in \ell^2, \forall \mathbf{x} \in \mathcal{X}$,
- using algebraic properties of the set of kernels, k_1 and k_2,
  - sum of kernels on same set $(k_1 + k_2)$,
  - scalar multiplication of kernel $(\alpha k) < 0, \qquad \alpha \in \mathbb{F}$,
  - product of kernel on different set, $k_1 k_2$
  - polynomial of kernel with positive coefficients,
  - taking exponential,
- using Taylor series,
- using Fourier series,
- pointwise limit of kernels.

We can define a number of feature maps via a kernel as follows.

- **Aronszajn** map: $\Phi : \mathbf{x} \to k(\mathbf{x}, \cdot)$, $\mathcal{H}_k$ is the associated reproducing kernel Hilbert space (RKHS) $k(\mathbf{x}, \mathbf{y}) = \langle k(\mathbf{x}, \cdot), k(\mathbf{y}, \cdot) \rangle$.
- **Kolmogorov** map: $\Phi : \mathbf{x} \to \mathbf{X}_\mathbf{x}$, $\mathcal{H}_k = L_2(\mathbb{R}^{\mathcal{X}}, \mu)$, where μ is the Gaussian measure $k(\mathbf{x}, \mathbf{y}) = E[\mathbf{X}_\mathbf{x} \mathbf{X}_\mathbf{y}]$.
- **Integral** map: there exists a set T and a measure μ such that on has $\Phi : x \to (\Gamma_x(t))_{t \in T}$, $\mathcal{H}_k = L_2(T, \mu)$, $k(x, y) = \int \Gamma(x, t)\Gamma(y, t) d\mu(t)$.
- **Basis** map: given any orthonormal basis $(f_\alpha)_{\alpha \in I}$ of the RKHS associated with $\mathcal{H}$, on has $\Phi : \mathbf{x} \to (f_\alpha)_{\alpha \in I}$, $\mathcal{H} = \ell_2(I)$ and $k(\mathbf{x}, \mathbf{y}) = \sum_{\alpha \in I} f_\alpha(\mathbf{x}) f_\alpha(\mathbf{y})$.

When infinite sums are involved like in the Basis map, it is important to specify in which sense the sum converges. In general the convergence occurs for each pair.

2.4.2 Advantages of kernel

It is well-known that the feature space suffers the computational problem. In case of high dimensional data the computational cost becomes very high. Using inner product by the kernel we can overcome this problem efficiently. First advantage of the kernel is that in many important special case feature computation will be very inexpensive by $k(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$, where $\mathbf{x}$ and $\mathbf{z}$ are in the arbitrary set $\mathcal{X}$.

Example 2.4.1 (Quadratic kernel) *Let* $\mathbf{x}, \mathbf{z} \in R^m$ *we have*

$$k(\mathbf{x}, \mathbf{z}) = [\mathbf{x}^T \mathbf{z}]^2 = (\sum_{i=1}^{m} x_i z_i)(\sum_{j=1}^{m} x_j z_j) = \sum_{i=1}^{m} \sum_{j=1}^{m} (x_i x_j)(z_i z_j) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle.$$

In case $m = 3$

$$\Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \\ \phi_4(\mathbf{x}) \\ \phi_5(\mathbf{x}) \\ \phi_6(\mathbf{x}) \\ \phi_7(\mathbf{x}) \\ \phi_8(\mathbf{x}) \\ \phi_9(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \end{bmatrix}$$

Cost for kernel, $k(\mathbf{x}, \mathbf{z}) = O(m)$ *is linear but for* $\phi(\mathbf{x}) = O(m^2)$ *is quadratic.*

Example 2.4.2 (Polynomial kernel) *Let* $\mathbf{x}, \mathbf{z} \in R^m$ *using polynomial kernel we have*

$$k(\mathbf{x}, \mathbf{z}) = [\mathbf{x}^T \mathbf{z} + c]^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 + 2c \langle \mathbf{x}, \mathbf{z} \rangle^2 + c^2 = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}). \rangle$$

In case $m = 3$

$$\Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \\ \phi_4(\mathbf{x}) \\ \phi_5(\mathbf{x}) \\ \phi_6(\mathbf{x}) \\ \phi_7(\mathbf{x}) \\ \phi_8(\mathbf{x}) \\ \phi_9(\mathbf{x}) \\ \phi_{10}(\mathbf{x}) \\ \phi_{11}(\mathbf{x}) \\ \phi_{12}(\mathbf{x}) \\ \phi_{13}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{bmatrix}$$

In general $\dim(\mathcal{H}) = \binom{m+d}{d}$ different features consisting all monomials having degree at most d with input dimension m.

The second advantage of the kernel is to use non-vectorial data, e.g., text strings or DNA sequences and so on. Another important advantage is that given a kernel we need neither the explicit form of the feature map nor the feature space, which are not also uniquely determined. Moreover, note that a similar construction can be made for arbitrary kernels and consequently every kernel has many different feature spaces. However, we can always construct a canonical feature space, namely the RKHSs (see Section 2.5). RKHSs have the remarkable and important property that norm convergence implies pointwise convergence.

2.4.3 Positive definite kernel (PDK)

Although we have already seen several techniques to construct kernels, in general, we still have to find a feature space in order to decide whether a given function k is a kernel or not. Since this can sometimes be a difficult task, we will now present a criterion that characterizes $\mathbb{R}$-valued kernels in terms of inequalities (positive semi-definite). By the

definition of positive definite function, if k is a $\mathbb{R}$-valued kernel with feature map $\Phi : \mathcal{X} \to \mathcal{H}$. The kernel k is symmetric since the inner product in $\mathcal{H}$ is symmetric. Moreover, k is also positive definite since for $n \in \mathbb{N}, \alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}$ and $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{X}$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) = \langle \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i), \sum_{j=1}^{n} \alpha_j \Phi(\mathbf{x}'_j) \rangle_{\mathcal{H}} \geq 0.$$

We can check whether a bivariate function is PDK by symmetry and positiveness.

Definition 2.4.2 (Kernel matrix or Gram matrix) *Given a kernel k and inputs $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{X}$, the square matrix*

$$K := (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$$

of order n is called the kernel matrix (Gram matrix) of k with respect to $\mathbf{x}_1, \cdots, \mathbf{x}_n$.

Definition 2.4.3 (Positive semi-definite matrix) *A real symmetric matrix K satisfying*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K_{ij} \geq 0, \ \forall \alpha_i \in \mathbb{R} \tag{2.2}$$

is called positive definite kernel. *It is said to be strictly positive definite if equality in (2.2) occurs for $\alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$.*

Definition 2.4.4 (Positive definite kernel) *A symmetric kernel $k(\cdot, \cdot)$ defined on a non-empty space $\mathcal{X}$ is called* positive definite *if for arbitrary number of points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ the kernel matrix $(k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n}$ is positive semi-definite.*

In kernel methods, the nonlinear feature map is given by a *positive definite kernel*, which provides nonlinear methods for data analysis with efficient computation. It is known that a positive definite kernel k is associated with a Hilbert space $\mathcal{H}$, called *reproducing kernel Hilbert space* (see Section 2.5), consisting of functions on $\mathcal{X}$ so that the function value is reproduced by the kernel (Aronszajn, 1950).

Definition 2.4.5 (Reproducing property) *For any function of RKHS, i.e., $f \in \mathcal{H}$ and a point $\mathbf{x} \in \mathcal{X}$, the function value $f(\mathbf{x})$ is given by*

$$f(\mathbf{x}) = \langle f(\cdot), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}, \tag{2.3}$$

where $\langle , \rangle_{\mathcal{H}}$ *in the inner product of* $\mathcal{H}$*, which is called* reproducing property.

The reproducing property says that each Dirac functional can be represented by the reproducing kernel. A Hilbert function space $\mathcal{H}$ that has a reproducing kernel k is always a RKHS (see Section 2.5). Every RKHS has a (unique) reproducing kernel and that this kernel can be determined by the Dirac functionals.

Definition 2.4.6 (Kernel trick) *By replacing* f *with* $k(\cdot, \tilde{\mathbf{x}})$ *in Eq. (2.3) yields*

$$k(\mathbf{x}, \tilde{\mathbf{x}}) = \langle k(\cdot, \mathbf{x}), k(\cdot, \tilde{\mathbf{x}}) \rangle_{\mathcal{H}} \text{ for any } \mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{X}.$$

To transform data for extracting nonlinear features, the feature map $\mathbf{\Phi} : \mathcal{X} \to \mathcal{H}$ *is defined by*

$$\mathbf{\Phi}(\mathbf{x}) = k(\cdot, \mathbf{x}),$$

which is regarded as a function of the first argument. The inner product of two feature vectors is then given by

$$\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\tilde{\mathbf{x}}) \rangle_{\mathcal{H}} = k(\mathbf{x}, \tilde{\mathbf{x}}).$$

This is known as the kernel trick.

The kernel trick serves as a central equation in the kernel methods. By this trick the kernel can evaluate the inner product of any two feature vectors efficiently without knowing an explicit form of neither $\mathbf{\Phi}(\cdot)$ nor $\mathcal{H}$. With this computation of inner product, many linear methods of classical data analysis techniques can be extended to nonlinear ones with an efficient computation based on Gram matrices. Once Gram matrices are computed, the computational cost does not depend on the dimensionality of the original space.

2.4.4 Well-known PDKs and its parameters

Assume $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite kernel. Then for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, the following kernels are real valued positive definite kernels on $\mathbb{R}$:

i. Linear kernel

$$k_0(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle = \mathbf{x}^T \mathbf{x}'.$$

It is just used the underlying Euclidean space to define the similarity measure. Whenever the dimensionality of $\mathbf{x}$ is very high, this may allow for more complexity in the function class than what we could measure and assess otherwise. It has limitation of linearity.

ii. Polynomial kernel

$$k_P(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d, \ (c \geq 0, d \in \mathbb{N}).$$

Using polynomial kernel it is possible to use the higher order correlation between the data in the different purposes. This kernel incorporates all polynomial interactions up to degree d (provided that $c > 0$). For instance, if we wanted to take only mean and variance into account, we would only need to consider $d = 2$ and $c = 1$. For higher emphasis on mean we need to increase the constant offset a. Polynomial kernels only map data into a finite dimensional space. Due to the finite bounded degree such kernel will not provide us with guarantees for a good dependency measure.

iii. Gaussian radial basis function (RBF) kernel

$$k_G(\mathbf{x}, \mathbf{x}') = e^{-s\|\mathbf{x}-\mathbf{x}'\|^2}, \ (s > 0).$$

Many radial basis function kernels, such as the Gaussian RBF kernel map $\mathbf{x}$ into an infinite dimensional space.

iv. Exponential kernel

$$k_E(\mathbf{x}, \mathbf{x}') = e^{(\alpha \mathbf{x}^T \mathbf{x}')}, \ (\alpha > 0).$$

v. Laplacian kernel

$$k_L(\mathbf{x}, \mathbf{x}') = e^{-\beta\|\mathbf{x}-\mathbf{x}'\|}, \ (\beta > 0).$$

- Binomial kernel: Let $\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 < 1\}$ and $\alpha > 0$. Then

$$k(\mathbf{x}, \mathbf{x}') := (1 - \langle \mathbf{x}, \mathbf{x}' \rangle)^{-\alpha}.$$

Table 2.1: Examples of well-known positive definite kernels and its parameters.

Kernel	$k(\mathbf{x}, \tilde{\mathbf{x}})$	Parameter
Polynomial	$(\langle \mathbf{x}, \tilde{\mathbf{x}} \rangle + c)^d$	$c \geq 0, d \in \mathbb{N}$
Gaussian RBF	$e^{-s\|\mathbf{x}-\tilde{\mathbf{x}}\|^2}$	$s > 0$
Exponential	$e^{(\alpha \mathbf{x}^T \tilde{\mathbf{x}})}$	$\alpha > 0$
Laplace	$e^{-\beta\|\mathbf{x}-\tilde{\mathbf{x}}\|}$	$\beta > 0$

Definition 2.4.7 (Stationary kernels) *Let $\kappa : \mathbb{R}^n \to \mathbb{R}$ be a function. Then κ is called a positive definite function (or function of positive type or Stationary kernels) if*

$$k(\mathbf{x}, \mathbf{z}) = \kappa(\mathbf{x} - \mathbf{z})$$

this type of the kernel is called stationary, a stationary kernel. It depends only on the lag vector separating the two examples $\mathbf{x}$ *and* $\mathbf{z}$ *but not on the examples themselves. A stationary kernel of the form $\phi(x-y)$ is also called a shift invariant (or translation invariant) kernel. Gaussian and Laplacian kernels are example of stationary kernel.*

Definition 2.4.8 (Nonstationary kernels) *A kernel is called nonstatinary, which depends on explicitly on the two examples* $\mathbf{x}$ *and* $\mathbf{z}$ *of $k(\mathbf{x}, \mathbf{z})$. For example Linear kernel and polynomial kernel.*

2.4.5 Valid PDKs and its matrices

Given an arbitrary set $\mathcal{X}$. Assume there exist Φ such that $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ (kernel trick) is a valid PDK, where $\mathbf{x}\,, \mathbf{x}' \in \mathcal{X}$. Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, c_2, \ldots, c_n \in \mathbb{R}$. Then

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \langle \mathbf{\Phi}(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = \langle \sum_{i=1}^{n} c_i \Phi(\mathbf{x}_i), \sum_{i=1}^{n} c_j \Phi(\mathbf{x})_j \rangle = \| \sum_{i=1}^{n} c_i \Phi(\mathbf{x}_i) \|^2 \geq 0$$

i.e. the symmetric matrix

$$\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \ldots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ldots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

is positive semidefinite. The kernel matrix $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n}$ is often called a Gram matrix in statistical machine learning literature. Symmetry an positive definiteness are not only necessary for k to be a PDK but also sufficient.

We do not actually need to have a centered Φ but for some methods we do need $\mathbf{K}$

$$\begin{aligned}
\tilde{\mathbf{K}}_{ij} &= \langle \tilde{\Phi}(\mathbf{x}_i), \tilde{\Phi}(\mathbf{x}_j) \rangle = \langle \Phi(\mathbf{x}_i) - \frac{1}{n}\sum_{a=1}^{n} \Phi(\mathbf{x}_a), \Phi(\mathbf{x}_j) - \frac{1}{n}\sum_{b=1}^{n} \Phi(\mathbf{x}_b) \rangle \\
&= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle - \frac{1}{n}\sum_{b=1}^{n} \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_b) \rangle - \frac{1}{n}\sum_{a=1}^{n} \langle \Phi(\mathbf{X}_a), \Phi(\mathbf{x}_j) \rangle + \frac{1}{n^2}\sum_{a=1}^{n}\sum_{b=1}^{n} \langle \Phi(\mathbf{x}_a), \Phi(\mathbf{x}_b) \rangle \\
&= K_{ij} - \frac{1}{n}\sum_{b=1}^{n} K_{ib} - \frac{1}{n}\sum_{a=1}^{n} K_{aj} + \frac{1}{n^2}\sum_{a=1}^{n}\sum_{b=1}^{n} K_{ab} \\
&= K_{ij} - \frac{1}{n}\sum_{b=1}^{n} K_{ib} 1_{bj} - \frac{1}{n}\sum_{a=1}^{n} 1_{ia} K_{aj} + \frac{1}{n^2}\sum_{a=1}^{n}\sum_{b=1}^{n} 1_{ia} K_{ab} 1_{bj} \\
&= (\mathbf{K} - \mathbf{K}(n^{-1}\mathbf{J}_n) - (n^{-1}\mathbf{J}_n)\mathbf{K} + (n^{-1}\mathbf{J}_n)\mathbf{K}(n^{-1}\mathbf{J}_n))_{ij}, \\
&= (\mathbf{HKH})_{ij} \qquad (2.4)
\end{aligned}$$

where for all i and j, $1_{ij} = 1$, $\mathbf{1}_n = [1, 1, \cdots, 1]^T$, and $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{J}_n$ with $\mathbf{J}_\mathrm{n} = 1_\mathrm{n} 1_\mathrm{n}^\mathrm{T}$.

The centered vector of a test point $\mathbf{x}$ is defied as

$$\begin{aligned}\tilde{\mathbf{k}}_{\mathbf{x}} &= [\tilde{k}(\mathbf{x},\mathbf{x}_1),\cdots,\tilde{k}(\mathbf{x},\mathbf{x}_1)]^T \\ &= \mathbf{k}_{\mathbf{x}} - \frac{1}{n}\mathbf{J}_n\mathbf{k}_{\mathbf{x}} - \frac{1}{n}\mathbf{K}\mathbf{1}_n + \frac{1}{n^2}\mathbf{J}_n\mathbf{K}\mathbf{1}_n \\ &= \mathbf{H}[\mathbf{k}_{\mathbf{x}} - \frac{1}{n}\mathbf{K}\mathbf{1}_n],\end{aligned}$$

where $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x},\mathbf{x}_1),\cdots,k(\mathbf{x},\mathbf{x}_n)]^T$.

Given a set of test points $\mathbf{x}_1^t,\cdots\mathbf{x}_L^t$, as in Eq. (2.1), we can define the centered matrix of order $L\times n$,

$$\tilde{\mathbf{K}}^t = \langle \Phi(\mathbf{x}_i^t) - \frac{1}{n}\sum_{a=1}^{n}\Phi(\mathbf{x}_a), \Phi(\mathbf{x}_j) - \frac{1}{n}\sum_{b=1}^{n}\Phi(\mathbf{x}_b)\rangle$$

in terms of the non centered matrix $\mathbf{K}^t = \langle \Phi(\mathbf{x}_i^t), \Phi(\mathbf{x}_j)\rangle$, we have

$$\tilde{\mathbf{K}}^t = \mathbf{K}^t - \frac{1}{n}\mathbf{J}_{L\times n}^T\mathbf{K} - \frac{1}{n}\mathbf{K}^t\mathbf{J}_{L\times n} + \frac{1}{n}\mathbf{J}_{L\times n}^T\mathbf{K}\frac{1}{2}\mathbf{J}_{L\times n},$$

where $\mathbf{J}_{L\times n}$ is the $L\times n$ matrix with all entries equal to 1.

A good monograph on the theory of positive definite kernels is Steinwart and Christmann, 2008 (Steinwart and Christmann, 2008, Chapter 4). The following sections all unspecified results are taken from that work.

Theorem 2.4.1 *For any set $\mathcal{X}\neq\emptyset$ and a real valued function $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is a PDK, i.e., it is symmetric and positive semi-definite, if and only if there is a mapping $\Phi:\mathcal{X}\to\mathcal{H}$ with a scalar product $\langle\cdot,\cdot\rangle$ such that*

$$k(\mathbf{x},\mathbf{x}') = \langle\Phi(\mathbf{x}),\Phi(\mathbf{x}')\rangle, \qquad \forall\mathbf{x},\mathbf{x}'\in\mathcal{X}.$$

Theorem 2.4.2 *Let $\mathcal{X}$ be a non-empty set and $f_n:\mathcal{X}\to\mathbb{F}$, $n\in\mathbb{N}$, be functions such that $(f_n(\mathbf{x}))_{n=1}^{\infty}\in\ell^2$ for all $\mathbf{x}\in\mathcal{X}$. Then*

$$k(\mathbf{x},\mathbf{x}') := \sum_{n=1}^{\infty} f_n(\mathbf{x})\overline{f_n(\mathbf{x}')}, \qquad \forall\mathbf{x}'\in\mathcal{X}, \tag{2.5}$$

defines a PDK on $\mathcal{X}$.

Definition 2.4.9 (Restriction of PDK) *Let k is a kernel on $\mathcal{X}$, $\tilde{\mathcal{X}}$ be a set, and $L : \mathcal{X} \to \tilde{\mathcal{X}}$ be a map. Then $\tilde{k}$ defined by*

$$\tilde{k}(\mathbf{x}, \mathbf{x}') := k(L(\mathbf{x}), L(\mathbf{x}')), \qquad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X},$$

is a PDK on $\tilde{X}$. In particular, if $\tilde{\mathcal{X}} \subset \mathcal{X}$, then $k_{|\tilde{\mathcal{X}} \times \tilde{\mathcal{X}}}$ is a PDK.

2.4.6 Properties of PDKs

Theorem 2.4.3 *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ be a PDK, $\mathcal{H}_c$ be a $\mathbb{C}$-Hilbert space, and $\Phi : \mathcal{X} \to \mathcal{H}_c$ be a feature map of k. Assume that we have $k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. Then $\mathcal{H}_r = \mathcal{H}_c$ equipped with the inner product*

$$\langle \mathbf{z}, \mathbf{z}' \rangle_{\mathcal{H}_r} := \mathrm{Re}\langle \mathbf{z}, \mathbf{z}' \rangle_{\mathcal{H}_c}, \qquad \mathbf{z}, \mathbf{z}' \in \mathcal{H}_r,$$

is an $\mathbb{R}$-Hilbert space, and $\Phi : \mathcal{X} \to \mathcal{H}_r$ is a feature map of k.

Theorem 2.4.4 *Let $\mathcal{X}$ be a set, $\alpha \geq 0$, and k, k_1, and k_2 be PDKs on $\mathcal{X}$. Then αk and $k_1 + k_2$ are also PDKs on $\mathcal{X}$.*

The preceding theorem 2.4.4 states that the set of PDKs on $\mathcal{X}$ is a cone. It is, however, not a vector space since, in general differences of PDKs are not a PDK.

Theorem 2.4.5 *Let k_1 be a PDK on $\mathcal{X}_1$ and k_2 be a PDK on $\mathcal{X}_2$. Then $k_1 \cdot k_2$ is a PDK on $\mathcal{X}_1 \times \mathcal{X}_2$. In particular, if $\mathcal{X}_1 = \mathcal{X}_2$, then $k(\mathbf{x}, \mathbf{x}') := k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$, $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, defines a PDK on $\mathcal{X}$.*

With the above two theorems, (2.4.4 and 2.4.5) it is easy to construct the non-trivial kernels. To illustrate this, let us assume for simplicity that $\mathcal{X} = \mathbb{R}$. Then for every integer $n \geq 0$, the map k_n defined by

$$k_n(\mathbf{x}, \mathbf{x}') := (\mathbf{x}\mathbf{x}')^n, \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$$

is a PDK by the 2.4.2. Consequently, if $p : \mathcal{X} \to \mathbb{R}$ is a polynomial of the for

$$p(t) = a_0 + a_1 t + \ldots + a_m t^m$$

with non-negative coefficients a_i, then

$$k(\mathbf{x}, \mathbf{x}') := p(\mathbf{x}\mathbf{x}'), \qquad \mathbf{x}, \mathbf{x}' \in \mathcal{X} \tag{2.6}$$

is define a PDK on $\mathcal{X}$ by theorem (2.4.4).

Theorem 2.4.6 *Let $p \geq 0$ and $d \geq 0$ be integers and $c \geq 0$ be a real number. Then k defined by $k(\mathbf{z}, \mathbf{z}') := (\langle \mathbf{z}, \mathbf{z}' \rangle + c)^p$, $\mathbf{z}, \mathbf{z}' \in \mathbb{C}^d$, is a kernel on $\mathbb{C}^d$. Moreover, its restriction to $\mathbb{R}^d$ is a $\mathbb{R}$-valued PDK.*

Note that the polynomial kernel defined by $p = 1$ and $c = 0$ are called a linear kernel. Instead of using polynomials for constructing kernels, one can use functions that can be represented by Taylor series. This is done in the following theorem.

Theorem 2.4.7 (Taylor type kernel) *Let $\mathring{\mathrm{A}}_{\mathbb{C}}$ and $\mathring{\mathrm{A}}_{\mathbb{C}^d}$ be the open unit balls of $\mathbb{C}$ and $\mathbb{C}^d$, respectively . Moreover, let $r \in (0, \infty)$ and $f : r\mathring{\mathrm{A}}_{\mathbb{C}} \to \mathbb{C}$ be holomorphic with Talylor series*

$$f(\mathbf{z}) = \sum_{n=0}^{\infty} a_n \mathbf{z}^n, \ \mathbf{z} \in \mathring{\mathrm{A}}_{\mathbb{C}}$$

If $a_n \geq 0$ for all $n \geq 0$, then

$$k(\mathbf{z}, \mathbf{z}') := f(\langle \mathbf{z}, \mathbf{z}' \rangle_{\mathbb{C}^d}) = \sum_{n=0}^{\infty} a_n \langle \mathbf{z}, \mathbf{z}' \rangle^n_{\mathbb{C}^d}, \qquad \mathbf{z}, \mathbf{z}' \in \sqrt{r}\mathring{\mathrm{A}}_{\mathbb{C}^d},$$

defines a kernel on $\sqrt{r}\mathring{\mathrm{A}}_{\mathbb{C}^d}$ whose restriction to $\mathcal{X} = \{x \in \mathbb{R}^d : \|\mathbf{x}\|_2 < \sqrt{r}\}$ is a real valued kernel. We say that k is a kernel of Taylor type.

Example 2.4.3 (Gaussian RBF kernel) *Let us denote jth component of a complex vector $\mathbf{z} \in \mathbb{C}^d$ by z_j. The complex Gaussian RBF kernel with band width $s > 0$ is given by*

$$k_{s,\mathbb{C}^d}(\mathbf{z}, \mathbf{z}') := e^{-s \sum_{j=1}^{d} (\mathbf{z}_j - \bar{\mathbf{z}}'_j)^2},$$

where $d \in \mathbb{N}$ and $\mathbf{z}, \mathbf{z}' \in \mathbb{C}^d$. If we make a restriction $k_s := (k_{s,\mathbb{C}^d})_{|\mathbb{R}^d \times \mathbb{R}^d}$ is an $\mathbb{R}$-valued kernel, which we call the (real) Gaussian RBF kernel with width s. Obviously, this kernel satisfies

$$k_s(\mathbf{x}, \mathbf{x}') := e^{-s\|\mathbf{x}-\mathbf{x}'\|_2^2}, \qquad \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R},$$

where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$.

Definition 2.4.10 (A family of spaces) *Let $s > 0$ and $d \in \mathbb{N}$. For a given holomophic function $f : \mathbb{C}^d \to \mathbb{C}$ we define*

$$\|f\|_{s,\mathbb{C}^d} := \left(\frac{2^d s^{2d}}{\pi^d} \int_C^d |f(\mathbf{z})|^2 e^{-s^2 \sum_{j=1}^d (z_j - \bar{z}_j')^2} \right),$$

whee dz stands for the complex Lebesgue measure on C^d. Furthermore, we write

$$\mathcal{H}_{s,\mathbb{C}^d} := \left\{ f : \mathbb{C}^d \to \mathbb{C} \middle| f \text{ holomorphic and } \|f\|_{s,\mathbb{C}^d} < \infty \right\}.$$

Example 2.4.4 (RBF kernel) *By considering RBF kernel*

$$\begin{aligned}
k(\mathbf{x}, \mathbf{z}) = e^{-\frac{1}{2s^2}\|\mathbf{x}-\mathbf{z}\|^2} &= e^{-\frac{1}{2s^2}\langle \mathbf{x},\mathbf{x}\rangle} e^{\frac{1}{s^2}\langle \mathbf{x},\mathbf{z}\rangle} e^{-\frac{1}{2s^2}\langle \mathbf{z},\mathbf{z}\rangle} \\
&= e^{-\frac{1}{2s^2}\langle \mathbf{x},\mathbf{x}\rangle} \left[1 + \frac{\langle \mathbf{x}, \mathbf{z}\rangle}{s^2} + \frac{1}{2!} \frac{\langle \mathbf{x}, \mathbf{z}\rangle^2}{(s^2)^2} + \dots \right] e^{-\frac{1}{2s^2}\langle \mathbf{z},\mathbf{z}\rangle} \\
&= e^{-\frac{1}{2s^2}\langle \mathbf{x},\mathbf{x}\rangle} \left[\sum_{i=0}^{\infty} \frac{1}{i!} \frac{\langle \mathbf{x}, \mathbf{z}\rangle^i}{(s^2)^i} \right] e^{-\frac{1}{2s^2}\langle \mathbf{z},\mathbf{z}\rangle} \\
&= \sum_{i=0}^{\infty} \frac{1}{i!} e^{-\frac{1}{2s^2}\langle \mathbf{x},\mathbf{x}\rangle} \frac{\mathbf{x}^i}{(s)^i} \frac{\mathbf{z}^i}{(s)^i} e^{-\frac{1}{2s^2}\langle \mathbf{z},\mathbf{z}\rangle} \\
&= \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle,
\end{aligned}$$

where

$$\Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \\ \vdots \end{bmatrix}, \; \phi_i(\mathbf{x}) = e^{-\frac{1}{2s^2}\langle \mathbf{x},\mathbf{x}\rangle} \frac{1}{i! s_i} \mathbf{x}^i.$$

Definition 2.4.11 (Fourier type kernel) *Let $f : [-2\pi, 2\pi] \to \mathbb{R}$ be a continuous function that can be expanded in a pointwise convergent Fourier series of the form*

$$f(t) = \sum_{n=0}^{\infty} a_n cos(nt).$$

If $a_n \geq 0$ holds for all $n \geq 0$ then $k(\mathbf{x}, \mathbf{x}') := \prod_{i=1}^{d} f(\mathbf{x}_i - \mathbf{x}'_i)$ defines a kernel on $[0, 2\pi)^d$. We say that k is a PDK of Fourier type.

Theorem 2.4.8 *Assume $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite and $\mathbf{x}, \mathbf{y} \in \mathcal{X}$,*

1. *Positive definiteness implies positivity on the diagonal $k(\mathbf{x}, \mathbf{x}) \geq 0$*
2. *Cauchy- Schwarz inequality $|k(\mathbf{x}, \mathbf{y})|^2 \leq k(\mathbf{x}, \mathbf{x})k(\mathbf{y}, \mathbf{y})$*

Proof 2.4.1 *Positive definiteness implies positive on the diagonal and symmetry i.e.*

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x}).$$

The 2×2 gram matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive. With fact $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$, the definition of positive definiteness implies that the eigenvalues of the hermitian matrix

$$\begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) \end{bmatrix} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_1) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) \end{bmatrix}$$

is non-negative, thus its determinant $k(\mathbf{x}_1, \mathbf{x}_1)k(\mathbf{x}_2, \mathbf{x}_2) - |k(\mathbf{x}_1, \mathbf{x}_2)|^2$ i.e.,

$$0 \geq K_{11}K_{22} - K_{12}K_{21} = K_{11}K_{22} - K_{12}K_{12} = K_{11}K_{22} - |K_{12}|^2$$

Theorem 2.4.9 (Limits of PDKs are alos a PDK) *Let (k_n) be a sequence of PDKs on the set $\mathcal{X}$ that converges pointwise to a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, i.e., $\lim_{n\to\infty} k_n(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. Then k is a PDK on $\mathcal{X}$.*

To prove we need to show k is symmetric and positive definite.

For any PDK, via space Φ, $\mathcal{H}$ so that kernel trick holds:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, \mathbf{x}_i), \qquad g(\cdot) = \sum_{j=1}^{n} \beta_j k(., \mathbf{x}_j), \qquad \langle f, g \rangle_{\mathcal{H}} = \sum_{i=1} \sum_{j=1} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j),$$

where $m, n \in \mathbb{N}$, $\alpha_i, \beta_j \in \mathbb{R}$ and $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$

- Symmetry:

$$\langle f, g \rangle = \sum_{i=1} \sum_{j=1} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1} \sum_{j=1} \alpha_i \beta_j k(\mathbf{x}_j, \mathbf{x}_i) = \langle g, f \rangle.$$

- Bilinearity:

$$\langle f, g \rangle = \sum_{i=1} \sum_{j=1} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{j=1} \beta_j f(\mathbf{x}_j) = \sum_{i=1} \alpha_i g(\mathbf{x}_i).$$

- Non-negative:

$$\langle f, f \rangle = \sum_{i=1} \sum_{j=1} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

Let us see what happens if we evaluate the inner product of $f \in \mathcal{H}$ with $k(\cdot, \mathbf{x}_i)$

$$\langle k(\cdot, \mathbf{x}), f(\cdot) \rangle = \sum_{i=1}^{n} \alpha_i \langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{x}_j) \rangle = \sum_{i=1}^{n} \alpha_i \langle k(\mathbf{x}, \mathbf{x}_j) \rangle = f(\mathbf{x}). \tag{2.7}$$

The above reproducing property, together with the Cauchy-Schwartz inequality $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq ||\mathbf{x}||||\mathbf{y}||$, implies that:

$$|f(\mathbf{x})|^2 = |\langle k(\cdot, \mathbf{x}), f \rangle|^2 \leq \langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{x}) \rangle \langle f, f \rangle$$

which in turn implies that $f = 0$ if $\langle f, f \rangle = 0$. Thus the vector space of feature map $\Phi(\mathbf{x}) = k(\cdot, \mathbf{x})$ induces with (2.7)

$$k(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}},$$

$\mathcal{H}$ as simply a pre-Hilbert space. To make it a true Hilbert space we must take its

closure, including the set of all limit points for sequences in $\mathcal{H}$ defined with respect to some (induced) norm.

2.4.7 Interplay between PDK, reproducing kernel and Mercer kernel

A kernel can be also defined on the functional viewpoint since each PDK k on a set $\mathcal{X}$ is associated a Hilbert space $\mathcal{H}_k$ of real-valued functions on $\mathcal{X}$.

Definition 2.4.12 (Reproducing kernel) *Let $\mathcal{X}$ be a non-empty arbitrary set. A bivariate function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called reproducing kernel a Hilbert space $\mathcal{H}$ on $\mathcal{X}$ iff*

i) for all elements of $\mathcal{X}$, the function k will be in $\mathcal{H}$ i.e.,

$$\forall \mathbf{x} \in \mathcal{X}, \quad k(\cdot, \mathbf{x}) \in \mathcal{H},$$

ii) the function k follows reproducing property.

Due to Moore Aronszajn theorem (Aronszajn, 1950), it can be shown that the definition of positive definite kernel and reproducing kernel are equivalent.

Proposition 2.4.1 *The reproducing kernel $k(\cdot, \mathbf{x})$ of a reproducing kernel Hilbert space $\mathcal{H}_k$ is a positive matrix.*

Proof 2.4.2 *Let us assume, $k_{\mathbf{x}}(\cdot) = k(\cdot, \mathbf{x})$ be the a reproducing kernel Hilbert space $\mathcal{H}_k$. We have*

$$\begin{aligned} 0 \leq \| \sum_{i=1}^{n} \alpha_i k_{\mathbf{x}_i} \|^2 &= \langle \sum_{i=1}^{n} \alpha_i k_{\mathbf{x}_i}, \sum_{j=1}^{n} \alpha_j k_{\mathbf{x}_j} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i \langle k_{\mathbf{x}_i}, k_{\mathbf{x}_j} \rangle \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

Hence

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

Theorem 2.4.10 (Moore Aronszajn, 1950) *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel on a set $\mathcal{X}$. Then, there uniquely exists a RKHS $\mathcal{H}_k$ on $\mathcal{X}$ such that*

- $k(\cdot, \mathbf{x}) \in \mathcal{H}_k,\ \forall \mathbf{x} \in \mathcal{X}$.
- *The subspace* $\mathcal{H}$ *of* $\mathcal{H}_k$*and* $\mathcal{H}$= *Span*$\{k(\cdot, \mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ *is dense in* $\mathcal{H}_k$.
- k *is the reproducing kernel on* $\mathcal{H}_k$ *, i.e.,*

$$f(\mathbf{x}) = \langle f, k(\cdot, \mathbf{x})\rangle_{\mathcal{H}_k},\ (\forall \mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H}_k).$$

- $\mathcal{H}_k$ *is the set of functions on* $\mathcal{X}$ *which are pointwise limits of Cauchy sequence in* $\mathcal{H}$ *with the inner product*

$$\langle f, g\rangle \quad = \quad \sum_{i=1} \sum_{j=1} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j)\rangle,$$

where

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, \mathbf{x}_i) \qquad \textit{and} \qquad g(\cdot) = \sum_{j=1}^{n} \beta_j k(\cdot, \mathbf{x}_j).$$

The above theorem tells us one-to 0ne correspondence between PDK and RKHS i.e.,

$k \Leftrightarrow \mathcal{H}_k$.

Theorem 2.4.11 (Mercer (1909)) *Let* $\mathcal{X} \subset \mathbb{R}^n, n \in \mathbb{N}$ *be closed, a strictly positive and finite Borel measure* ν *on* $\mathcal{X}$ *and a continuous* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *satisfying: for any finite of points* $\{\mathbf{x}_i\}_{i=1}^N$ *in* $\mathcal{X}$ *and real numbers* $\{a_i\}_{i=1}^N$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0,$$

and

$$\int_{\mathcal{X}} \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}')^2 \mathrm{d}\nu(\mathbf{x}) \mathrm{d}\nu(\mathbf{x}') < \infty.$$

Let $L_k : L_\nu^2(\mathcal{X}) \to L_\nu^2(\mathcal{X})$ *be a integral operator defined as*

$$L_k f(\mathbf{x}) = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \mathrm{d}\nu(\mathbf{x}'), \ \mathrm{f} \in L_2(\nu)$$

with a countable system of nonnegative eigenvalues $\{\mu_i\}_{i=1}^{\infty}$ *satisfying* $\sum_{i=1}^{\infty} \mu_i^2 < \infty$ *and corresponding orthonormal eigenfunctions* $(e_i)_{i=1}^{\infty}$. *Then*

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \mu_i e_i(\mathbf{x}) e_i(\mathbf{x}'), \ \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \tag{2.8}$$

where the convergence is absolute for each pair $(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$ *and uniform on each compact subset of* $\mathcal{X}$.

We can define the feature map via the Mercer's theorem as follows

$$e_\nu : \mathcal{X} \to \ell^2, \ e_\nu(\mathbf{x}) = (\sqrt{\lambda_i} e_i(\mathbf{x}))_{i=1}^{\infty},$$

where if only $N < \infty$ of the eigenvalues are strictly positive, then $e_\nu : \mathcal{X} \to \mathbb{R}^N$.

The notion of positive definite kernel was not well-known before the 1950s. Instead, researchers have been used kernels that satisfy the conditions of Mercer's theorem, called *Mercer's kernel*. A kernel which satisfied the conditions of Mercer's theorem implies that the kernel also satisfied the kernel trick i.e., $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ but the converse is not true. So all Mercer's kernels are PDKs but all PDKs are not Mercer's kernels. Mercer's theorem is a special case of the basis map. It gives a stronger (uniform) convergence properties of the kernel representation, but needs additional assumption, namely $\mathcal{X}$ has to be compact and the kernel is continuous. Mercer's theorem has played a crucial role in supervised learning, say SVM in which the kernel trick is introduced via Mercer's theorem.

2.5 Reproducing kernel Hilbert space (RKHS)

It is known (Aronszajn, 1950) that a positive definite kernel k is associated with a Hilbert space $\mathcal{H}$, called *reproducing kernel Hilbert space* (RKHS), consisting of functions on $\mathcal{X}$ so that the function value is reproduced by the kernel.

Definition 2.5.1 (Reproducing kernel Hilbert spaces) *Let a set $\mathcal{X} \neq \emptyset$. A RKHS of a kernel k, $\mathcal{H}_k$ over $\mathcal{X}$ is a space of functions $f : \mathcal{X} \to \mathbb{R}$ such that*

- *The function k is defined as $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.*
- *For every $\mathbf{x} \in \mathcal{X}$ and $f \in \mathcal{H}_k$*

$$f(\mathbf{x}) = \langle f(\cdot), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}_k}.$$

  where $k_{\mathbf{x}} = k(\cdot, \mathbf{x}) \in \mathcal{H}_k$ is a function with fixed $\mathbf{x}$ (k has reproducing property).

- $\mathcal{H}_k = \overline{span\{k(\cdot, \mathbf{x}) : \mathbf{x} \in \mathcal{X}\}}$ *i.e., k spans $\mathcal{H}_k$, where $\overline{A}$ denotes the completion set of A.*

The Reproducing kernel Hilbert spaces (RKHS) of a PDK are in a certain sense the smallest feature space of this PDK and consequently can serve as a canonical feature space. Now, we briefly discuss the properties of RKHS.

2.5.1 Properties of RKHS

Reproducing kernel Hilbert spaces have the remarkable and important property that norm convergence implies pointwise convergence. More precisely, let $\mathcal{H}$ be a RKHS, $f \in \mathcal{H}_k$, and $(f_i)_i^{\infty} \subset \mathcal{H}_k$ be a sequence with $\|f_n - f\|_{\mathcal{H}} \to 0$ for $i \to \infty$ (there is a convergence sequence). Then for all $\mathbf{x} \in \mathcal{X}$, we have

$$\lim_{n\to\infty} f_n(\mathbf{x}) = \lim_{n\to\infty} \delta_{\mathbf{x}}(f_n) = f(\mathbf{x}).$$

Reproducing kernels are actually kernels since the *feature map* is defined by the kernels. The reproducing property says that each Dirac functional can be represented by the reproducing kernel. Consequently, a Hilbert function space that has a reproducing kernel is always a RKHS. Every RKHS has a (unique) reproducing kernel and this kernel can be determined by the Dirac functional.

Theorem 2.5.1 *Let $\mathcal{H}_k$ be a Hilbert function space over $\mathcal{X}$ with a reproducing kernel k. Then $\mathcal{H}_k$ be a RKHS over $\mathcal{X}$ and $\mathcal{H}_k$ is also a feature space of k, where the feature map*

$\Phi : \mathcal{X} \to \mathcal{H}_k$ *is defined as*

$$\Phi(\mathbf{x}) = k(\cdot, \mathbf{x}),\ \forall \mathbf{x} \in \mathcal{X}.$$

Theorem 2.5.2 *Let $\mathcal{H}_k$ be a RKHS over $\mathcal{X}$. Then $k : \mathcal{X} \times \mathcal{X} \to \mathbb{K}$ defined by*

$$k(\mathbf{x}, \mathbf{x}') := \langle \delta_{\mathbf{x}}, \delta_{\mathbf{x}'} \rangle,\ \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X},$$

is the only reproducing kernel on $\mathcal{H}_k$. Furthermore, if $(\mathbf{e}_i)_{i \in I}$ is an orthonormal basis (ONB) or complete orthonormal system of $\mathcal{H}_k$ then for all $\mathbf{x}, \mathbf{x}' \in X$, we have

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i \in I} e_i(\mathbf{x}) \overline{e_i(\mathbf{x}')}, \tag{2.9}$$

where the convergence is absolute.

The theorem 2.5.2 tells us, a RKHS uniquely determines its reproducing kernel, which is actually a kernel. The ONB in the theorem2.5.2 is not necessarily countable. However, RKHSs over separable metric spaces having a continuous kernel are always separable and hence all their ONBs are countable. In particular, the RKHSs of Gaussian RBF kernels always have countable ONBs. A non separable Hilbert space of continuous functions on a separable topological space has no reproducing kernel.

Theorem 2.5.3 *Let $\mathcal{X} \neq 0$ and k be a kernel over $\mathcal{X}$ with the feature space $\mathcal{H}$ and feature map $\Phi : \mathcal{X} \to \mathcal{H}$. Then*

$$\mathcal{H}_k := \{\langle w, \Phi(\cdot) \rangle_{\mathcal{H}} : w \in \mathcal{H}\} \tag{2.10}$$

equipped with the norm

$$\|f\|_{\mathcal{H}_k} := \inf \{\|w\|_{\mathcal{H}} : w \in \mathcal{H} \ \ with\ f = \langle w, \Phi(\cdot) \rangle_{\mathcal{H}}\} \tag{2.11}$$

is the only RKHS of k. In particular both definitions are independent of the choice of $\mathcal{H}$

and Φ_0 *and the operator* $V : \mathcal{H} \to \mathcal{H}_k$ *defined by*

$$Vw := \langle w, \Phi(\cdot)\rangle_{\mathcal{H}},\ w \in \mathcal{H}$$

is a metric surjection, i.e., $V\mathring{A}_{\mathcal{H}} = \mathring{A}_{\mathcal{H}_k}$ *, where* $\mathring{A}_{\mathcal{H}}$ *and* $\mathring{A}_{\mathcal{H}_k}$ *are the open unit balls of* $\mathcal{H}$ *and* $\mathcal{H}_k$*, respectively.*

Observations on the theorem 2.5.3: the RKHS $\mathcal{H}_k$ of a given kernel k as the "smallest" feature space of k in the sense that there is a canonical metric surjection $\mathbf{V}$ from any other feature space $\mathcal{H}_0$ of k onto $\mathcal{H}_k$. The soft margin SVM produce decision functions of the form $x \to \langle w, \Phi(x)\rangle$, where $\Phi : \mathcal{X} \to \mathcal{H}_k$ is a feature map of k and $w \in \mathcal{H}_k$ is a suitable weight vector. Now, (2.10) states that the RKHS associated with k consists exactly of all possible functions of this form. Moreover, (2.10) shows that this set of functions does not change if we consider different feature spaces or feature maps of k. The Theorem 2.5.3 can often be used to determine the RKHS of a given kernel and its modifications such as restrictions and normalization. To illustrate this let us recall that every $\mathbb{C}$-valued kernel on $\mathcal{X}$ that is actually $\mathbb{R}$-valued has an $\mathbb{R}$-feature space.

Corollary 2.5.1 *Let* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ *be a kernel and* $\mathcal{H}_k$ *its corresponding* $\mathbb{C}$*-RKHS. If we actually have* $k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$*, then*

$$\mathcal{H}_{k_R} := \inf\left\{f : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \,\middle|\, \exists_g \in \mathcal{H}_k \ \ with\, Re\, g = f\right\},$$

equipped with the norm

$$\|f\|_{\mathcal{H}_{k_R}} := \inf\left\{\|g\|_{\mathcal{H}_k} : g \in \mathcal{H}_k\, with\, Re\, g = f\right\}, \quad f \in \mathcal{H}_{k_R},$$

is the $\mathbb{R}$*-RKHS of the* $\mathbb{R}$*-valued kernel* k*.*

Given a RKHS, $\mathcal{H}_k$ and its kernel $k(\mathbf{x}, \mathbf{y})$ on $\mathcal{X}$, then for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:

- $k(\mathbf{x}, \mathbf{x}) \geq 0$
- $k(\mathbf{y}, \mathbf{x}) = k(\mathbf{x}, \mathbf{y})$
- $|k(\mathbf{y}, \mathbf{x})|^2 \leq k(\mathbf{y}, \mathbf{y})k(\mathbf{x}, \mathbf{x})$

- Let $\mathbf{x}_0 \in \mathcal{X}$. Then the following are equivalent:
  - $k(\mathbf{x}_0, \mathbf{x}_0) = 0$.
  - $k(\mathbf{y}, \mathbf{x}_0) = 0,\ \forall \mathbf{y} \in \mathcal{X}$.
  - $f(\mathbf{x}_0) = 0,\ \forall f \in \mathcal{H}$.

2.5.2 Representer theorem

Representer theorem implies optimizer is the linear combination of kernels subject to the sample points

Theorem 2.5.4 (Kimeldorf and Wahba (1971)) *Let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \in \mathcal{X}$ and $\mathcal{H}_k$ be a reproducing kernel Hilbert space with a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, a symmetric positive semi-definite function of the compact domain. For any function $L : \mathbb{R}^n \to \mathbb{R}$ and any nondecreasing function $\Omega : \mathbb{R} \to \mathbb{R}$. If*

$$J^* = \min_{f \in \mathcal{H}} J(f) = \min_{f \in \mathcal{H}} \{\Omega(\|f\|_{\mathcal{H}}) + L(f(\mathbf{x}_1), \dots f(\mathbf{x}_n))\}$$

is well-defined, then there are some $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}$, such that

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, \mathbf{x}_i)$$

achieves $J(f) = J^*$. Further more if Ω is increasing, then each minimizer of J(f) can be expressed in the form $f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, \mathbf{x}_i)$.

Example 2.5.1 *Let $\mathcal{X}$ be a set and $(e_1, e_2, \dots, e_n)$ be an orthonormal basis in $\mathcal{H}$ also define*

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} e_i(\mathbf{x}) \overline{e}_i(\mathbf{x}').$$

Then for any $\mathbf{x}' \in \mathcal{X}$

$$k(\cdot, \mathbf{x}') = \sum_{i=1}^{n} \overline{e}_i(\mathbf{x}') e_i(\cdot)$$

belongs to $\mathcal{H}_k$ and for any function

$$\varphi(\cdot) = \sum_{i=1}^{n} \lambda_i e_i(\cdot)$$

in $\mathcal{H}$, we have

$$\begin{aligned} \forall \mathbf{x}' \in \mathcal{X},\ \langle \varphi, k(\cdot, \mathbf{x}') \rangle_{\mathcal{H}} &= \langle \sum_{i=1}^{n} \lambda_i e_i(\cdot), \sum_{j=1}^{n} \overline{e}_j(\mathbf{x}') e_j(\cdot) \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \overline{e}_j(\mathbf{x}') \langle e_i(\cdot), e_j(\cdot) \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n} \lambda_i e_i(\mathbf{x}') = \varphi(\mathbf{x}') \end{aligned}$$

Any finite dimensional Hilbert space of functions has a reproducing kernel.

Example 2.5.2 *Let $k(i, j) = \delta_{ij}$ (delta function or Kronecker symbol, equal to* 1 *if $i = j$ and* 0 *otherwise). Then*

$$\begin{aligned} &\forall \quad j \in \mathbb{N},\ k(\cdot, j) = (0, 0, \ldots, 1, \ldots, 0) \in \mathcal{H} \qquad (\textit{1 at the jth place}) \\ &\forall \quad j \in \mathbb{N},\ \forall \mathbf{x} = (\mathbf{x}_i)_{i \in \mathbb{N}} \in \mathcal{H},\ \langle \mathbf{x}, k(\cdot, j) \rangle_{\mathcal{H}} = \sum_{i \in \mathbb{N}} x_i \overline{\delta}_{ij} = \mathbf{x}_j. \end{aligned} \tag{2.12}$$

k is the reproducing kernel of $\mathcal{H}$.

2.6 Methods based on RKHS

The kernel based methods are developed by combining the kernel trick and representer theorem, i.e.,

Kernel trick + Representer theorem = Foundation of kernel methods.

A number of kernel methods have been proposed as in the supervised learning, in unsupervised learning, in nonparametric inference and so on.

2.6.1 Supervised learning

Supervised learning is the main and well-know first type of machine learning. It consists of input-output pairs. For given a labeled set of input-output pairs $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$, the goal of supervised learning is to learn a function, $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{D}$ is a training set with n training data points. The useful supervised kernel methods on RKHS are as follows:

- Support vector machine for classification (SVM, Boser et al., 1992).
- Support vector machine for regression (Smola and Schölkopf, 1998).
- Kernel ridge regression (Saunders et al., 1998).

2.6.2 Unsupervised learning

Unsupervised learning, knowledge discovery is the second main type of machine learning. It consists of only input data and arguably more typical of human learning. For given an input $\mathcal{D} = \{(X_i)\}_{i=1}^n$ the goal of unsupervised methods are to discover interesting structure in the data where $\mathcal{D}$ is a training set with n data points. Useful unsupervised kernel methods of RKHS are as follows:

- Kernel principal component analysis (Schölkopf et al., 1998) .
- Kernel canonical correlation analysis (Akaho, 2001).
- Kernel K-mean cluster analysis (Kim et al., 2005).
- Gradient-based kernel dimension reduction for regression (Fukumizu and Leng, 2014).

2.6.3 Nonparametric inference

Nonparametric inference methods on RKHS are the recent developed kernel methods. Nonparametric inference aims to identify very general processes from the data. The useful kernel methods of nonparametric inference are as follows:

- Bayesian inference with positive definite kernels (Fukumizu et al., 2013).
- Kernel two-sample test (Gretton, 2012).

Chapter 3

Automatic Way of Finding Hyperparameters in Kernel Principal Component Analysis

3.1 Motivation

Dimension reduction is an essential part of modern data analysis, where we often need to handle large dimensional data. The purpose of dimension reduction may be visualized, noise reduction, and pre-processing for further analysis. Among others, the principal component analysis (PCA, Pearson, 1901) is one of the most famous methods to reduce the dimensionality by projecting data onto a low-dimensional subspace with largest variance.

Kernel principal component analysis (kernel PCA, Schölkopf et al., 1998) has been proposed as a nonlinear extension of the standard PCA, and has been applied to various purposes including feature extraction, denoising, and pre-processing of regression. Kernel PCA is an example of the so-called kernel methods (Schölkopf and Smola, 2002), which aim to extract nonlinear features of the original data by mapping them into a high-dimensional *feature space* (reproducing kernel Hilbert space, RKHS). This mapping is called *feature map*. A number of methods have been proposed as kernel methods, which include support vector machine (SVM, Boser et al., 1992), a novel multiclass SVM algorithm using mean reversion and coefficient of variance (Premanode et al., 2013), kernel ridge regression (Saunders et al., 1998), kernel canonical correlation analysis (Akaho,

2001, Bach and Jordan, 2002), Bayesian Inference with Positive Definite Kernels (Fukumizu et al., 2013), Gradient-Based Kernel Dimension Reduction for Regression (Fukumizu and Leng, 2014) and so on.

It is well known that the performance of a kernel method is highly dependent on the choice of the kernel. For supervised learning such as SVM and kernel ridge regression, cross-validation is popularly used for choosing the hyperparameters of a kernel algorithm, such as parameters in a kernel (e.g., bandwidth of Gaussian RBF kernel), with the objective function of learning. On the other hand, no well-founded methods have been proposed in general for unsupervised learning such as kernel PCA and kernel canonical correlation analysis.

This chapter focuses on kernel PCA and proposes a method for choosing hyperparameters: parameters in a kernel and the number of kernel principal components. In the case of standard linear PCA, the algorithm can be formulated as minimization for self-regression with reduced rank, and cross-validation approaches have been proposed for choosing the number of components (Krzanowski, 1987, Wold, 1978). In contrast, while a similar regression formulation is possible for kernel PCA, the cross-validation approach is not applicable straightforwardly for choosing a kernel in kernel PCA: the error of the regression is given by the RKHS norm of the feature space associated with the kernel, and thus the cross-validation errors are not comparable for different kernels.

As detailed in Section 3.2, the proposed method for choosing the hyperparameters of kernel PCA uses cross-validation for the reconstruction errors of pre-images in the original space. The pre-image of a feature vector is defined by an approximate inverse image of the feature map (Mika et al., 1999). Various methods have been already proposed to calculate the pre-image of a feature vector, as explained in Section 3.2.1 (Mika et al., 1999, Kwok and Tsang, 2003, Bakir et al., 2004, Rathi et al., 2006, Arias et al., 2007, Zheng et al., 2010). In the proposed method, given an evaluation data in the cross-validation, we compute the pre-image of the corresponding feature vector projected onto the subspace given by kernel PCA, and then evaluate the reconstruction error of the evaluation point. A kernel and the number of components corresponding to the minimum average reconstruction error are chosen as the optimum ones. We demonstrate the effectiveness of this method experimentally with various synthesized and real-world datasets.

3.1.1 Kernel principal component analysis (kernel PCA)

Kernel PCA (Schölkopf et al., 1998) conducts principal component analysis for the feature vectors. More precisely, given data points $\mathbf{X}_i \in \mathcal{X}$, $i = 1, 2, \ldots, n$, kernel PCA outputs a set of principal functions by the following two-step procedure: (i) transform the data nonlinearly into the feature space $\mathcal{H}$, i.e., $\mathbf{X}_i \mapsto \Phi(\mathbf{X}_i)$, (ii) solve the linear PCA problem for the feature vectors, i.e., solve the directions in $\mathcal{H}$ for which the variance of $\{\Phi(\mathbf{X}_i)\}$ along those directions is maximized.

The algorithm of kernel PCA is described as follows (for the detail, see Schölkopf et al. (1998)). Let $\tilde{\Phi}(\mathbf{X}) := \Phi(\mathbf{X}) - \frac{1}{n}\sum_{j=1}^{n}\Phi(\mathbf{X}_j)$ be the centered feature vector. The estimated covariance matrix is given by $\mathbf{H} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\Phi}(\mathbf{X}_i)\tilde{\Phi}(\mathbf{X}_i)^T$ with the centered feature vectors. The principal directions $g \in \mathcal{H}$ are given by the unit eigenvectors corresponding to the largest eigenvalues, and thus the problem is converted to solving the eigenequation

$$\mathbf{H}g = \tilde{\lambda}g.$$

By using the kernel trick, this problem is reduced to the generalized eigen value problem that finds $g = \sum_{i=1}^{n}\alpha_i\tilde{\Phi}(\mathbf{X}_i)$ such that

$$M\alpha = n\tilde{\lambda}\alpha, \quad \text{subject to} \quad \alpha^T M\alpha = 1, \tag{3.1}$$

where M is the $n \times n$ centered Gram matrix defined by $M = CKC$ with $K_{ij} = k(\mathbf{X}_i, \mathbf{X}_j)$ and $C = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$. Here I_n is the identity matrix of size n, and $\mathbf{1}_n$ is the vector with n ones. The constraint $\alpha^T M\alpha = 1$ corresponds to the condition $\langle g_j, g_h\rangle = \delta_{jh}$, where δ_{jh} is the Kronecker's delta.

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ denote the ordered eigenvalues of M with associated eigenvectors $\alpha_1, \ldots \alpha_n$, where $\alpha_j = (\alpha_{1j} \ldots \alpha_{nj})^T$. The vectors are normalized so that $\alpha_j^T M\alpha_h = \delta_{jh}$. The j-th principal direction $g_j \in \mathcal{H}$ is then given by

$$g_j = \frac{1}{\sqrt{\lambda_j}}\sum_{i=1}^{n}\alpha_{ij}\tilde{\mathbf{\Phi}}(\mathbf{X}_i),$$

and the j-th principal component of the data point $\mathbf{X}_i$ is given by

$$\langle g_j, \tilde{\Phi}(\mathbf{X}_i)\rangle = \frac{1}{\sqrt{\lambda_j}}(M\alpha_j)_i = \sqrt{\lambda_j}\alpha_{ij}.$$

For a test point $\mathbf{X}$ out of the sample, the j-th principal component is similarly given by

$$\langle g_j, \tilde{\Phi}(\mathbf{X})\rangle = \frac{1}{\sqrt{\lambda_j}}\sum_{i=1}^{n}\tilde{k}(\mathbf{X}, \mathbf{X}_i)\alpha_{ij},$$

where $\tilde{k}(x, y) = k(x, y) - \frac{1}{n}\sum_{i=1}^{n} k(x, \mathbf{X}_i) - \frac{1}{n}\sum_{i=1}^{n} k(\mathbf{X}_i, y) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(\mathbf{X}_i, \mathbf{X}_j)$ is the centered kernel.

3.1.2 Choice of kernel

The result of kernel PCA obviously depends on the choice of the kernel. It is often the case that the kernel has some parameters like the popular examples shown in Table 2.1. In such a case, these parameters may have a strong influence on the results. To depict the influence, using *wine* data (see Section 3.3) we show the plots of the first two kernel principal components with different values of inverse-bandwidth parameter s in the Gaussian RBF kernel, and degree d and constant c in the polynomial kernel (Figure 3.2). From the figure, we see that in both the kernels the results of kernel PCA depend strongly on the parameters, and an appropriate choice is indispensable for the method to give reasonable low-dimensional representation of data.

It is known that the standard PCA can be formulated as a self-regression or reconstruction problem; namely, the first r principal components of centered data $\{\mathbf{X}_i\}_{i=1}^{n} \subset \mathbb{R}^d$ are equal to the projections $B\mathbf{X}_i$ given by the reduced rank regression

$$\min_{A,B}\sum_{i=1}^{n}\|\mathbf{X}_i - AB\mathbf{X}_i\|^2 \quad \text{subject to } BB^T = I_r,$$

where A and B are $d \times r$ and $r \times d$ matrices, respectively. Based on this regression formulation, the cross-validation approach (Stone, 1974) has been used for the standard PCA to choose the number of components (Wold, 1978, Krzanowski, 1987) by minimizing the above self-regression errors.

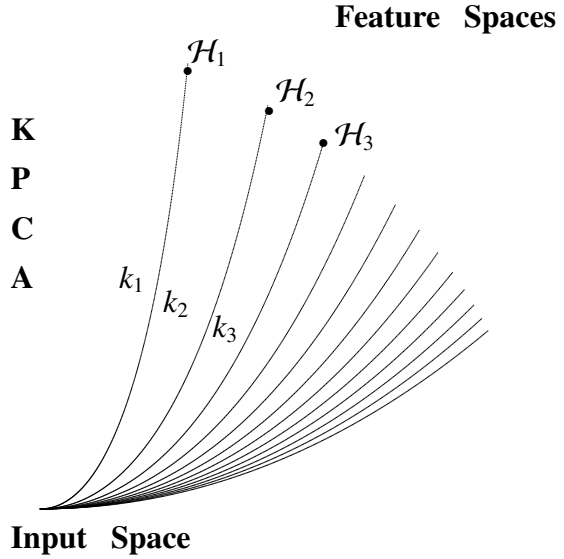

Figure 3.1: Individual reproducing kernel Hilbert space for each kernel of the kernel principal component analysis (KPCA).

In a similar manner, the kernel PCA can be also formulated as the self-regression of the centered feature vectors. In fact, it is easy to see that the first r principal directions are given by

$$\min_{f_j, g_j \in \mathcal{H}} \sum_{i=1}^{n} \left\| \tilde{\Phi}(\mathbf{X}_i) - \sum_{j=1}^{r} f_j \langle g_j, \tilde{\Phi}(\mathbf{X}_i) \rangle \right\|_{\mathcal{H}}^2,$$

where $f_j, g_j \in \mathcal{H}$ with $\langle g_j, g_\ell \rangle_{\mathcal{H}} = \delta_{j\ell}$. One might expect that this self-regression formulation could be applied to the cross-validation method for choosing a kernel in kernel PCA. This is not possible, however, because the above regression error is measured by the RKHS norm given by the kernel, and thus the errors are not comparable among different kernels. This problem is shown in the Figure 3.1.

The goal of this chapter is thus to propose a method of choosing a kernel (and the number of components) in kernel PCA by introducing a criterion comparable for different kernels (Alam and Fukumizu, 2014).

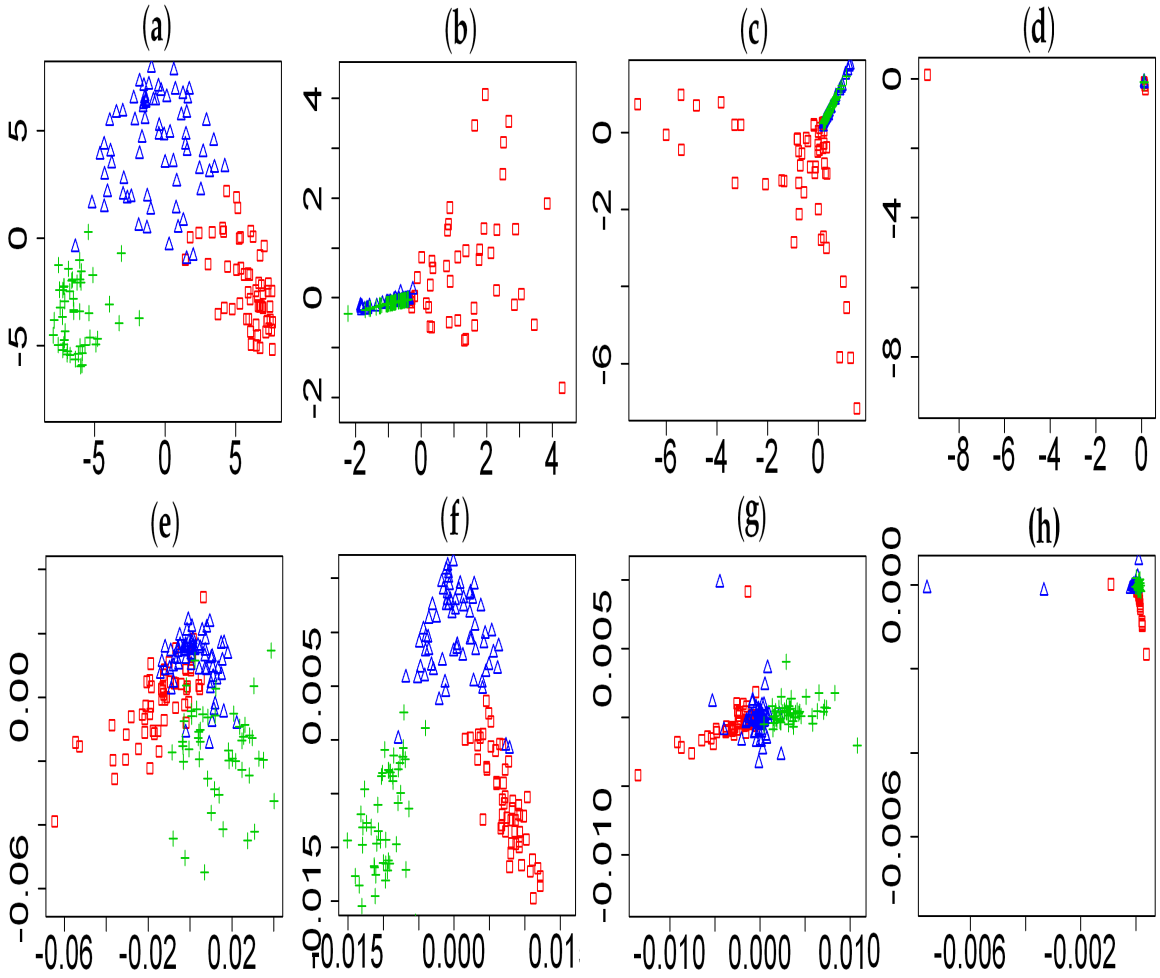

Figure 3.2: Scatter plots of the first two kernel principal components for *wine* data: Gaussian RBF kernel is used in the top panel (a) $s = 0.05$ (b) $s = 0.75$ (c) $s = 1$ (d) $s = 10$, and polynomial kernel in the bottom (e) $c = 0.001$, $d = 2$ (f) $c = 10$, $d = 2$ (g) $c = 1$, $d = 3$ (h) $c = 1$, $d = 4$.

3.2 Proposed methods

The proposed method for choosing a kernel and the number of component uses cross-validation by the comparable reconstruction errors in the original space. To evaluate the errors, we need to solve the pre-image of the feature vectors projected on the subspace given by the principal directions. We first give a brief review of pre-image methods.

3.2.1 Pre-Image of kernel PCA

While many kernel methods provide their output in the form of feature vectors in the RKHS, in some problems we want to find a point in the original space. In Mika et al.

(1999), kernel PCA is applied to a denoising task, in which an image corresponding to the RKHS vector obtained by kernel PCA is used as a denoised version of the original image.

Given a vector f in RKHS $\mathcal{H}$, it is in general not possible to find a rigorous pre-image, that is a point $\mathbf{X}$ in the original space such that $\Phi(\mathbf{X}) = f$ holds exactly. We thus define an (approximate) *pre-image* of f by the minimizer of

$$\min_{\mathbf{Z}\in\mathcal{X}} \|f - \Phi(\mathbf{Z})\|^2_{\mathcal{H}}.$$

In the original paper, Mika et al. (1999) have used the fixed-point iterative method. Many other approaches have also been proposed to solve the pre-image problem. A non-iterative approach of distance constraint has been proposed by Kwok and Tsang (2003), while it is dependent on the choice of neighborhood. An approach of learning a pre-image map was developed by Bakir et al. (2004). To apply this technique, we need an additional regularization parameter. Some authors have extended these approaches in different ways (Rathi et al., 2006, Arias et al., 2007, Zheng et al., 2010). More recently, a two-stage closed-form approach has been also proposed (Honeine and Richard, 2011). These advanced methods, however, usually require some tuning parameters. We use the fixed-point method in our proposed method, since it has a simple form for Gaussian RBF kernel.

We here explain the fixed-point method for solving the pre-image problem in the kernel PCA setting. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_n \in \mathbb{R}^m$ be the training data for kernel PCA, and $g_j = \sum_i \alpha_{ji}\tilde{\Phi}(\mathbf{X}_i)$ $(j = 1, \ldots, \ell)$ be the unit principal directions. The projector onto the subspace spanned by $\{g_j\}_{j=1}^{\ell}$ is denoted by $\mathbf{P}_\ell$, i.e., $\mathbf{P}_\ell f = \sum_{j=1}^{\ell}\langle f, g_j\rangle g_j$. Given test point $\mathbf{X}$ in the original space, the feature vector projected onto the principal subspace is given by $\mathbf{P}_\ell\tilde{\Phi}(\mathbf{X})$. The pre-image of this vector in the RKHS is defined by the minimizer of

$$\rho(\mathbf{Z}) = \left\|\mathbf{P}_\ell\tilde{\Phi}(\mathbf{X}) - \tilde{\Phi}(\mathbf{Z})\right\|^2_{\mathcal{H}}. \tag{3.2}$$

It is easy to see that

$$\begin{aligned}\rho(\mathbf{Z}) &= \left\|\tilde{\Phi}(\mathbf{Z})\right\|^2_{\mathcal{H}} - 2\langle\tilde{\Phi}(\mathbf{Z}), \mathbf{P}_\ell\tilde{\Phi}(\mathbf{X})\rangle_{\mathcal{H}} + \left\|\mathbf{P}_\ell\tilde{\Phi}(\mathbf{X})\right\|^2_{\mathcal{H}} \\ &= \tilde{k}(\mathbf{Z}, \mathbf{Z}) - 2\sum_{i=1}^{\ell}\gamma_i\tilde{k}(\mathbf{Z}, \mathbf{X}_i) + \Omega,\end{aligned} \tag{3.3}$$

where $\gamma_i = \sum_{j,h} \alpha_{ji}\alpha_{jh}\tilde{k}(\mathbf{X}_h, \mathbf{X})$ and Ω is a constant independent of $\mathbf{Z}$.

For Gaussian RBF kernel, Eq.(3.3) is equal to $\rho(\mathbf{Z}) = 1 - 2\sum_{i=1}^{n} \gamma_i e^{-s\|\mathbf{X}_i - \mathbf{Z}\|^2} + \Omega$, and by setting the derivative zero we obtain the fixed-point algorithm:

$$\mathbf{Z}_{t+1} = \frac{\sum_{i=1}^{n} \gamma_i e^{-s\|\mathbf{X}_i - \mathbf{Z}_t\|^2}}{\sum_{j=1}^{n} \gamma_j e^{-s\|\mathbf{X}_j - \mathbf{Z}_t\|^2}} \mathbf{X}_i = \frac{\sum_{i=1}^{n} a_i \mathbf{X}_i}{\sum_{j=1}^{n} a_j}, \tag{3.4}$$

where $a_i = \gamma_i e^{-s\|\mathbf{X}_i - \mathbf{Z}_t\|^2}$.

In the case of polynomial kernels, the fixed point condition does not derive such an iterative form as the Gaussian RBF kernel. We thus use the steepest descent method for Eq.(3.3) in our experiments on polynomial kernels in Section 3.3.3.

3.2.2 Hyperparameters choice

For the objective function of cross-validation, we use reconstruction errors between a test point $\mathbf{X}$ and the corresponding pre-image $\mathbf{Z}$ of the projected feature vector $\mathbf{P}_\ell \tilde{\Phi}(\mathbf{X})$ given by kernel PCA. The reconstruction errors are measured by the distance of the original space $\mathcal{X} = \mathbb{R}^m$. By this approach, unlike the regression error in the RKHS, we can consider comparable errors for different kernels. The architecture of the proposed method is given in Figure 3.3. The algorithm of the kernel choice in kernel PCA is given in Figure 3.4. We describe the leave-one-out cross validation (LOOCV) for simplicity, but the extension to the general K-fold cross-validation is straightforward. By a similar algorithm we are able to select the number of principal components or any other hyperparameters.

In solving approximate pre-images, the fixed-point or the steepest descent method may be trapped by local minima. To avoid this problem, we use five initial points for the optimization algorithm, and choose the best one. As shown in the next section, the obtained pre-images give appropriate results.

Note also that the fixed-point method may not work well for a very large inverse-bandwidth s, since the term of the nearest $\mathbf{X}_i$ is dominant in the right hand side of Eq. (3.4) so that $\mathbf{Z}_t$ may stay at $\mathbf{X}_i$. In the experiments, we set a reasonable parameter range of s by checking the kernel PCA results with two components.

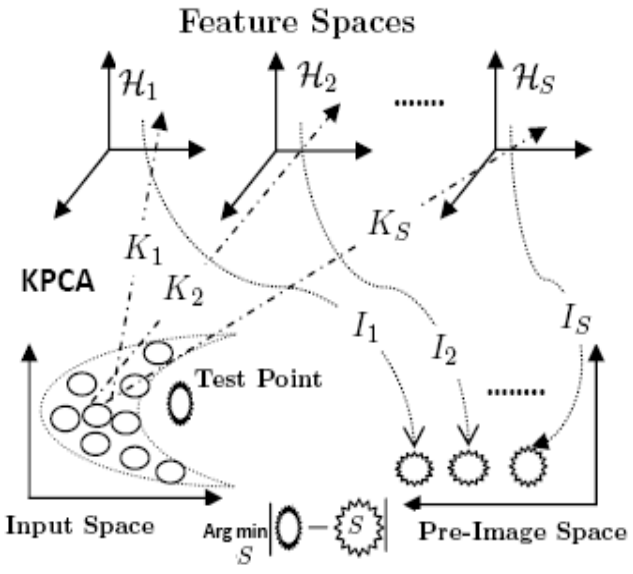

Figure 3.3: Architecture of kernel choice in kernel PCA.

Table 3.1: Computational cost (in second) of the proposed method for synthesized data-2 with different data sizes (n) and the numbers of components (ℓ)

n/ℓ	2	4	6	8	10
100	20.3	20.3	22.3	28.5	29.0
200	86.8	87.0	102	110	152
400	512	549	610	684	896
600	1.54×10^3	1.55×10^3	1.56×10^3	1.63×10^3	2.23×10^3
800	3.39×10^3	3.40×10^3	3.67×10^3	3.51×10^3	4.77×10^3
1000	6.06×10^3	6.51×10^3	6.50×10^3	6.52×10^3	1.07×10^4

3.2.3 Computational cost

To illustrate the computational cost of the proposed method, the CPU time (in second) for six different sizes of data (n) and five numbers of components (ℓ) using synthesized data-2 are shown in Table 3.1. The CPU time increases as the sample size is larger, since the computation of LOOCV and the optimization of pre-images is heavier for larger samples. The configuration of the computer is Intel (R) Core (TM) i7 CPU 920@ 2.67 GHz., memory 12.00 GB and 64-bit operating system. We have used 'kernlab' package in R program for implementation of the kernel PCA. The Gaussian RBF kernel is used with inverse bandwidth $s = 50$.

Input: $D = \{\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_n\}$ in $\mathbb{R}^m$. Parameters $\{s_1, \ldots, s_T\}$ for kernel k_s. Threshold TH.

1. Set $h = 1$.
2. Do the following steps:
   (1) Set $i = 1$.
   (2) Solve kernel PCA for $D - \{\mathbf{X}_i\}$ with kernel k_{s_h} (Eq. 3.1).
   (3) Compute the approximate pre-image $\mathbf{Z}_i^h$ for $\mathbf{X}_i$ using the fixed-point method Eq.(3.3). The iteration stops if $\|\mathbf{Z}_{t+1} - \mathbf{Z}_t\| < TH$.
   (4) Compute the reconstruction error $E_i^h = \|\mathbf{X}_i - \mathbf{Z}_i^h\|^2$.
   (5) $i := i + 1$.
   (6) If $i > n$, BREAK; otherwise go to (2).
3. Compute the LOOCV error $E^h = \frac{1}{n} \sum_{i=1}^{n} E_i^h$.
4. $h := h + 1$.
5. If $h > T$, END. Otherwise, go to 2.
6. $h_{opt} := \arg\min_h E^h$.

Output: $s_{opt} := h_{opt}$.

Figure 3.4: Algorithm of kernel choice in kernel PCA with Gaussian RBF kernel.

3.3 Experimental results

We apply the proposed method for choosing the parameters in a kernel and the number of principal components in kernel PCA for various datasets. The Gaussian RBF kernel is used, except Section 3.3.3, where the polynomial kernel is discussed. We use two synthesized and seven real-world datasets, which are summarized in Table 3.2. For the real-world datasets, we standardize each variable of data before applying kernel PCA. In solving pre-images, we take initial values from the uniform distribution on the interval $[-1, 1]$. The detailed discussions on the results will be shown in Section 3.4.

3.3.1 Synthesized data

We use two synthesized datasets to illustrate the effectiveness of the proposed method. Each dataset is of two dimension, and have three clusters.

Table 3.2: The configuration of datasets for hyperparmeters choice in kernel principal component analysis (kernel PCA).

dataset	# data	Dimension	# classes
Synthesized-1	175	2	(3)
Synthesized-2	150	2	(3)
Wine	178	13	3
Diabetes	145	3	3
BUPA	345	6	2
Fertility	100	9	2
Zoo	101	16	7
USPSG-500	500	256	5
Food	961	6	-

Synthesized data-1. 175 data are generated along three circles of different radii with small noise:

$$\mathbf{X}_i = r_i \begin{pmatrix} \cos(\mathbf{Z}_i) \\ \sin(\mathbf{Z}_i) \end{pmatrix} + \epsilon_i, \tag{3.5}$$

where $r_i = 1$, 0.5 and 0.25, for $i = 1, \ldots, 100$, $i = 101, \ldots, 150$, and $i = 151, \ldots, 175$, respectively, $\mathbf{Z}_i \sim U[-\pi, \pi]$ and $\epsilon_i \sim \mathcal{N}(0, 0.01\, I_2)$ independently.

Synthesized data-2. This is an example taken from Schölkopf and Smola (2002, Chapter 14). The dataset has 150 points, which consists of 50 points from each of three Gaussian distributions with means $(-0.5, -0.1)$, $(0, 0.7)$ and $(0.5, 0.1)$ and variance 0.1.

We prepare the inverse bandwidth parameters $s \in \{0.05, 1, 5, 10, 25, 50\}$ and $s \in \{1, 5, 10, 20, 50, 100, 200\}$ for *synthesized data-1* and *synthesized data-2*, respectively, and calculate the LOOCV reconstruction errors by pre-images. To see the variations over sampling, we generate 100 samples for each case of data 1 and 2, and make box plots. Figure 3.5 shows (a): scatter plots of a sample of the original datasets, (b): the box plots, and (c,d): the scatter plots of first two kernel principal components with the best kernel bandwidths (c) and with other ones (d). We can see by comparing (c) and (d) that the proposed method chooses a hyperparameter that can separate three clusters clearly, which suggests the effectiveness of the method. Note that kernel PCA does not use the explicit information of the three clusters, while they are displayed with different colors and markers for visualization purpose.

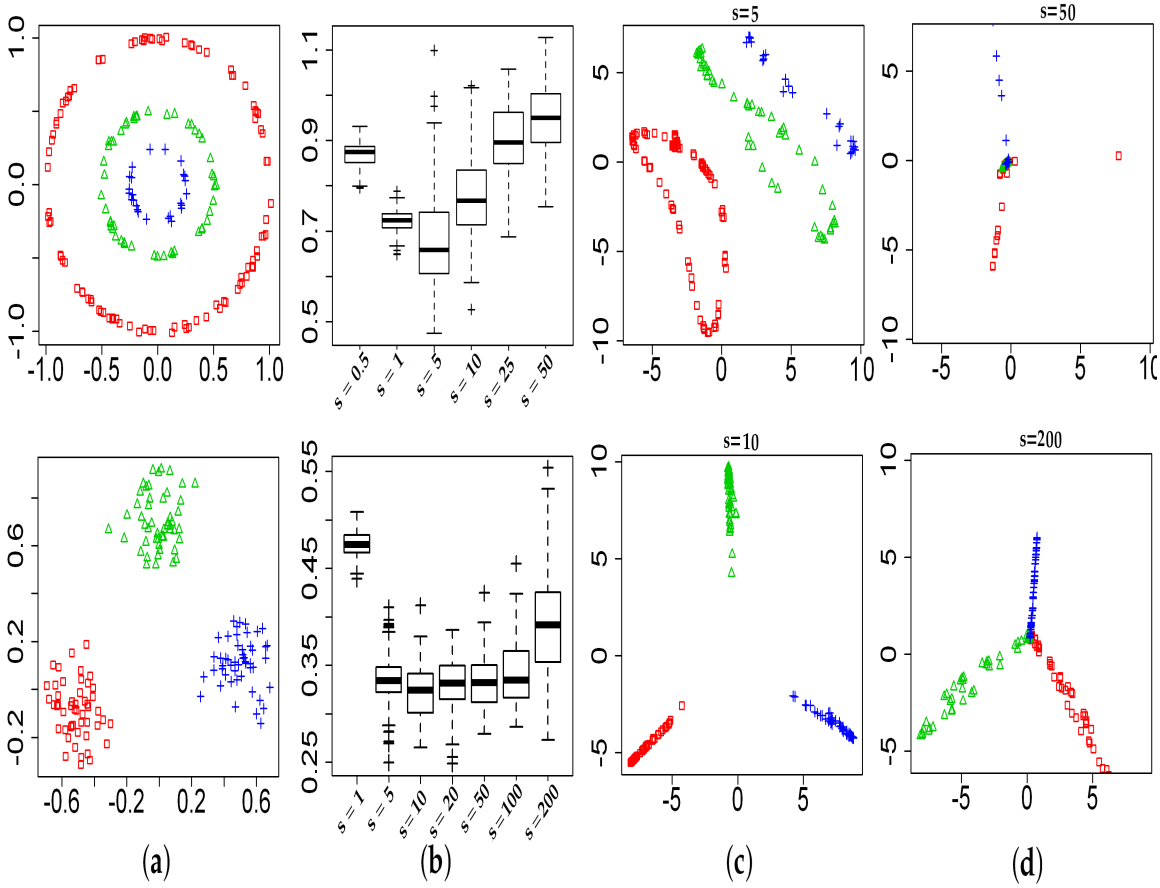

Figure 3.5: Kernel PCA for synthesized data-1 (top) and synthesized data-2 (bottom). (a) Scatter plot for the two variables of a sample. (b) Box plots of the leave-one-out cross validation (LOOCV) reconstruction errors for 100 samples. (c, d) scatter plots of the first two kernel principal components using (c) the best inverse kernel widths ($s = 5, 10$) and (d) larger bandwidths $s = 50, 200$.

3.3.2 Real world problems

We first apply the proposed method to five datasets: *wine*, *diabetes*, *BUPA liver disorders*, *fertility*, and *zoo*, the former three of which are taken from Izenman (2008) and available at the website of the book, and the latter two are taken from the UCI machine learning repository (Bache and Lichman, 2013).

As the kernel PCA is an unsupervised method, the evaluation of results is not straightforward. Since kernel PCA is often used as a pre-processing technique for regression and classification, we evaluate the LOOCV classification errors with the k-NN classifier ($k = 5$) to see the appropriateness of the hyperparameters chosen by the proposed method. Note that we do not use the class labels for kernel PCA, but use them only for evaluating the

classification errors.

We consider a set of inverse bandwidths $s \in \{0.05, 0.10, 0.25, 0.50, 0.75, 1.00, 10.00\}$ and six numbers of kernel principal components $\ell \in \{2, 3, 4, 5, 8, 10\}$ for each dataset. The LOOCV reconstruction errors used in the proposed method and the LOOCV classification errors for all the hyperparameters are shown in Table 3.3, from which we see that the selected hyperparameters attain the minimum or close to the minimum classification error for all the datasets. This suggests that the proposed method provides appropriate hyperparameters that maintain the cluster structure effective for the classification tasks.

We next apply the proposed method for larger datasets in dimensionality and sample size. *USPS* data (Song et al., 2008) consists of 16×16 grayscale images of handwritten digits, and thus the dimensionality is 256. The original dataset has 2007 images, but we draw 100 images from each of five digits $1, 2, 3, 4, 5$, and add Gaussian noise with mean 0 and standard deviation 0.01. The dataset is referred to as *USPSG-500*. We take seven inverse bandwidths $s \in \{0.0001, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025\}$ and eight numbers of kernel principal components $\ell \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. The LOOCV reconstruction errors in the proposed method are shown in Table 3.4, in which the minimum is attained at $s = 0.01$ and $\ell = 64$. The kNN ($k = 5$) misclassification rates estimated with LOOCV are also listed in the table.

We next apply the proposed method to *the nutritional value of food*, which is not for classification. The dataset has 961 food items with six nutritional components as attributes (Izenman, 2008, Chapter 7). We consider seven values of inverse bandwidths $s \in \{0.001, 0.1, 0.5, 0.75, 1, 5, 10, 100, 200\}$ and five numbers of components $\ell \in \{1, 2, 3, 4, 56\}$. The results are displayed in Table 3.5. The smallest LOOCV reconstruction error is attained at $s = 0.5$ and $\ell = 2$. Since, unlike classification tasks, it is not straightforward to evaluate the performance of the proposed method, we show the scatter plots of the first two kernel principal components using three values of inverse bandwidths $s \in \{0.001, 0.5, 200\}$ in Figure 3.6.

3.3.3 Polynomial kernel

We use the proposed method for choosing the hyperparameters in the polynomial kernel. Using *wine* dataset, we consider seven values of offset parameters $c \in \{0.1, 0.5, 1, 5, 10, 25, 50\}$,

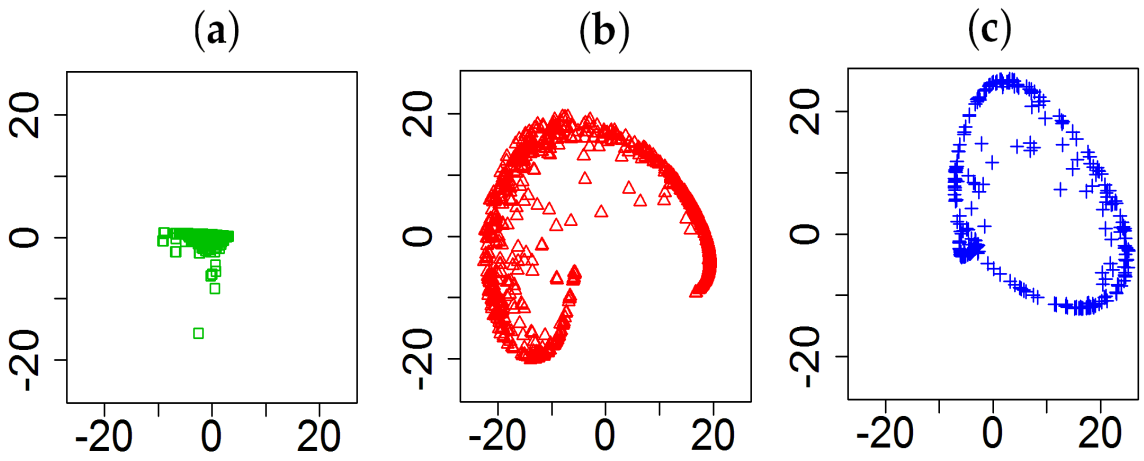

Figure 3.6: Visualization of the first two kernel principal components of *food* data (a) $s = 0.001$, (b) $s = 0.5$ and (c) $s = 200$

two values of degree $d \in \{2, 3\}$, and four numbers of kernel principal components $\ell \in \{2, 3, 4, 5\}$. The results are given in Table 3.6. We observe that the smallest LOOCV reconstruction error is attained in the area close to the minimum classification error.

3.4 Discussion

While kernel PCA has been applied in various areas of the machine learning, such as dimensionality reduction, feature extraction, de-noising, and so on (Schölkopf and Smola, 2002, Rathi et al., 2006, Hofmann, 2007, Zheng et al., 2010, Feng and Liu, 2013), in most cases the kernel and a number of features are chosen in a heuristic way. Recently, multi-kernel PCA (Multi-kernel PCA, Ren et al., 2013) has been also proposed, which applies the combination of multiple kernels instead of choosing one. It is well known, however, that the multi-kernel approach results in a computationally heavy algorithm, which may need advanced optimization technique. The method proposed in this paper, in contrast, is based on the reconstruction errors in the original space, which can be regarded as a natural extension of the aim of the standard linear PCA. The required computation is simply cross-validation with a basic optimization algorithm such as the fixed-point or gradient method.

We provide detailed discussions on the experimental results for real-world data sets in Section 3.3. For classification data sets, we can see from Tables 3.3 and 3.4 that the hyper-parameter (bandwidth parameter in the Gaussian RBF kernel and the number of principal components) gives the best or close to best LOOCV classification error: the best for *wine*

data, and the second or third best for the other 5 data sets. In all cases, we observe that the chosen hyperparameters are close to the best parameters for the classification error. These experimental observations imply that the proposed method gives appropriate hyperparameters, with which the low dimensional features obtained by kernel PCA represent effective information of data.

From Table 3.5 and Figure 3.6, we can see that the hyperparameter chosen by the proposed method provides the features with a clearer structure than the other two hyperparameters used in (a) and (c). For this data set, Izenman (2008) provides detailed analysis on the results of kernel PCA with a hand-tuned bandwidth parameter: a meaningful "curve" structured is observed in the result of two-dimensional kernel PCA. As shown in Figure 3.6, the proposed method automatically chooses such a hyperparameter that accords with the observation in Izenman (2008).

We can also observe from Table 3.6 that the proposed method chooses the hyperparameters for kernel PCA with polynomial kernel so that the corresponding LOOCV for classification error attains the third best. This accords with the observation on the other cases with the Gaussian RBF kernel, and demonstrates the appropriateness of the proposed method.

Regarding the computational cost of the proposed method, the proposed method needs to solve the pre-image problem for each of the data, which may cause a computational issue for large data set. Table 3.1 shows that the computational time increases roughly quadratically with respect to the sample size. To reduce the computational cost, it may be possible to use only a part of data for evaluating reconstruction errors in choosing hyperparameters.

Table 3.3: Five real-world data sets: leave-one-out cross validation (LOOCV) reconstruction errors and LOOCV classification errors for inverse bandwidths (s) and the number of components (ℓ). The minimum values are written in bold fonts, and the classification errors with the hyperparameters chosen by the proposed method are underlined.

	Reconstruction errors						Classification errors					
s/ℓ	2	3	4	5	8	10	2	3	4	5	8	10
Wine												
0.05	3.749	3.846	3.952	3.713	3.893	4.040	5.056	3.933	3.371	2.809	3.371	2.809
0.10	**3.418**	3.495	3.582	3.560	3.556	3.845	**2.247**	2.809	3.371	2.247	2.809	3.371
0.25	3.422	3.596	3.531	3.885	3.584	3.733	**2.247**	2.809	4.494	3.933	5.618	5.618
0.50	3.518	3.603	3.651	3.719	3.790	3.723	3.933	5.057	6.180	5.618	7.303	8.427
0.75	3.789	3.703	3.751	3.858	3.882	3.939	25.281	7.303	6.180	6.180	7.865	9.551
1.00	3.788	3.923	3.883	3.919	3.807	3.825	33.708	30.337	9.551	8.427	9.551	10.674
10.00	4.131	4.070	4.005	4.073	4.119	4.134	39.888	41.573	42.697	41.011	38.764	42.135
Diabetes												
0.05	2.343	2.398	2.183	7.591	16.027	20.605	20.690	19.310	19.310	**20.000**	**20.000**	**20.000**
0.10	1.761	1.872	1.795	1.713	4.879	6.913	23.448	19.310	19.310	24.828	24.138	25.517
0.25	1.598	1.467	1.660	1.636	2.066	2.751	20.690	**20.000**	19.310	20.690	21.379	**20.000**
0.50	1.505	**1.318**	1.492	1.476	1.597	1.712	22.069	20.690	**20.000**	21.379	21.379	21.379
0.75	1.555	1.494	1.560	1.519	1.575	1.716	21.379	**20.000**	22.069	22.069	22.069	21.379
1.00	1.626	1.617	1.609	1.530	1.647	1.512	20.690	23.448	24.138	20.690	**20.000**	20.690
10.00	2.362	2.167	2.152	2.098	2.292	2.269	37.241	37.241	40.000	34.483	36.552	37.241
BUPA												
0.05	2.964	2.751	2.758	2.190	5.462	5.300	42.319	43.479	**40.580**	42.029	42.029	42.029
0.10	2.439	2.232	2.325	2.042	4.266	4.838	48.116	46.667	41.739	47.826	48.116	47.826
0.25	2.064	2.042	**2.012**	2.123	2.175	2.269	50.145	48.696	42.029	50.145	50.145	50.145
0.50	2.138	2.148	2.077	2.238	2.166	2.071	50.145	46.957	44.638	49.855	49.855	49.855
0.75	2.253	2.196	2.147	2.364	2.138	2.241	53.333	42.609	49.855	53.333	53.333	53.623
1.00	2.128	2.177	2.154	2.282	2.256	2.123	50.145	43.189	47.826	50.145	50.145	50.145
10.00	2.464	2.447	2.467	2.427	2.392	2.481	44.058	44.928	44.928	44.058	44.058	44.058
Fertility												
0.05	3.955	4.132	4.100	3.911	3.876	3.811	13.000	14.000	16.000	13.000	13.000	13.000
0.10	3.570	3.568	3.560	8.067	3.490	3.428	15.000	11.000	12.000	15.000	15.000	15.000
0.25	**3.325**	3.330	3.349	3.279	3.442	3.407	11.000	14.000	15.000	11.000	11.000	11.000
0.50	3.601	3.592	3.630	3.713	3.764	3.559	13.000	11.000	11.000	13.000	13.000	13.000
0.75	3.896	3.848	3.911	4.031	3.624	3.673	**10.000**	**10.000**	12.000	**10.000**	**10.000**	**10.000**
1.00	3.989	3.936	3.892	3.919	3.774	3.819	12.000	15.000	13.000	12.000	12.000	12.000
10.00	3.678	3.663	3.568	3.714	3.489	3.500	12.000	15.000	13.000	12.000	12.000	12.000
Zoo												
0.05	4.581	4.644	5.460	6.051	7.434	5.957	12.871	12.871	13.861	12.871	11.881	11.881
0.10	3.861	3.858	3.816	3.820	4.886	5.369	14.851	11.881	12.871	15.842	16.832	14.851
0.25	3.607	3.615	3.632	3.748	3.863	3.871	19.802	15.842	**10.891**	17.822	19.802	19.802
0.50	3.572	4.078	**3.460**	3.637	3.935	3.667	22.772	12.871	11.881	22.772	22.772	21.782
0.75	3.523	3.591	3.801	3.750	3.893	4.140	27.723	27.723	26.733	26.733	24.752	24.752
1.00	3.738	3.853	3.866	3.999	3.896	4.013	24.752	25.743	30.693	22.772	24.752	25.743
10.00	4.013	4.049	3.992	4.024	4.037	4.006	56.436	53.465	48.514	55.446	55.446	56.436

Table 3.4: USPSG-500: LOOCV reconstruction errors and LOOCV classification errors (bold numbers indicate the minimum value).

s/ℓ	2	4	8	16	32	64	128	256
	Reconstruction errors in the proposed method							
0.0001	1139.810	1203.316	1159.02	752.494	134.936	130.678	143.080	534.779
0.001	129.168	129.627	124.333	110.106	143.470	82.734	69.729	203.068
0.0025	42.422	40.708	44.493	38.588	51.707	26.516	92.497	107.448
0.0050	18.967	21.120	22.642	20.010	18.957	20.592	26.828	33.991
0.0075	18.989	15.903	16.963	14.804	14.369	13.909	14.879	17.523
0.010	16.648	15.081	14.161	12.785	12.485	**12.444**	15.787	14.270
0.025	13.339	13.498	13.149	13.085	13.915	14.173	14.086	14.595
	Classification errors (%)							
0.0001	32.00	11.20	4.60	2.00	3.00	3.00	3.4	4.0
0.001	31.00	12.20	4.40	2.20	2.80	3.00	3.00	4.20
0.0025	31.40	11.60	4.40	2.60	2.20	3.00	3.20	3.40
0.0050	31.60	11.00	4.60	3.00	**1.80**	2.40	3.60	4.40
0.0075	28.20	11.40	4.80	3.40	**1.80**	2.80	3.20	5.20
0.010	31.00	15.20	4.40	3.80	3.00	2.20	2.60	5.00
0.025	45.80	25.40	7.60	5.20	5.60	6.80	5.20	15.60

Table 3.5: LOOCV reconstruction errors for *food* data.

s/ℓ	1	2	3	4	5	6
0.001	20.226	18.741	18.334	18.361	18.462	13.901
0.1	2.215	2.024	1.977	1.840	1.849	1.956
0.5	1.923	**1.738**	2.143	2.097	2.034	1.922
0.75	1.817	1.908	1.883	1.891	1.850	1.930
1	1.854	1.844	1.813	1.798	2.050	1.927
5	2.306	2.214	2.128	2.229	2.203	2.238
10	2.380	2.286	2.200	2.239	2.808	2.259
100	1.987	1.982	1.943	2.014	2.088	2.234
200	2.070	2.066	2.097	2.123	2.234	2.192

Table 3.6: Polynomial kernel for *wine* data: LOOCV reconstruction errors and the LOOCV classification errors (bold numbers indicate the minimum value).

	$\ell = 2$		$\ell = 3$		$\ell = 4$		$\ell = 5$	
$c\backslash d$	2	3	2	3	2	3	2	3
	Reconstruction errors in the proposed method							
0.1	4.165	3.807	4.059	3.818	4.108	3.821	4.153	3.805
0.5	4.051	3.781	3.978	3.758	4.003	3.768	3.952	3.805
1.0	3.976	3.837	3.888	3.869	3.966	3.709	4.023	3.819
5.0	3.752	3.859	3.759	3.813	4.108	3.803	4.153	3.739
10.0	3.784	3.780	3.740	3.810	4.003	3.762	3.952	3.792
25.0	3.755	3.820	**3.709**	3.730	3.966	3.768	4.023	3.775
50.0	3.761	3.782	3.735	3.724	3.750	3.777	3.736	3.743
	Classification errors (%)							
0.1	18.539	17.978	16.292	3.933	15.730	5.056	15.730	4.494
0.5	14.045	17.978	11.798	3.933	11.798	3.933	14.045	4.494
1.0	15.730	16.292	12.360	3.371	11.798	3.933	8.989	3.933
5.0	2.247	3.371	3.933	3.371	3.933	3.371	**1.685**	3.933
10.0	2.809	**1.685**	3.371	2.809	**1.685**	3.371	2.247	2.809
25.0	3.933	3.371	2.809	2.247	4.494	2.247	2.247	2.247
50.0	4.494	3.933	2.809	2.809	4.494	2.247	2.247	2.247

Chapter 4

Classical, Robust and Kernel Canonical Correlation Analysis

4.1 Motivation

Canonical correlation analysis (CCA) is a multivariate procedure for assessing the linear relationship between two sets of variables (or features) (Hotelling, 1936). We refer to it as a classical CCA. Many statisticians, biometricians, economists, social scientists (Anderson, 2000, Press, 1987) and many other researchers apply it in diverse field of knowledge. But this classical method has a number of limitations, say linear association and model assumptions. It is also sensitive to outliers. Therefore, there is essential need for increase robustness, nonlinear association and flexibility of the feature selection.

A number of robust CCA were compared and discussed by Branco *et al.*, (Branco et al., 2005). Taskinen et al. (2006) obtained influence function and asymptotic distributional properties of classical CCA based on robust estimates of the covariance matrix (Skocaj et al., 2004). Many researchers developed a few robust methods of CCA and suggested that from the viewpoint of robustness and computation the performance of minimum covariance determinate (MCD) estimator is the best (Branco et al., 2005). In this chapter, we consider CCA based on MCD estimator as an estimator of class of robust methods.

Because of the linearity assumption classical CCA gives us a naive measurement in case of nonlinear datasets. Even robust methods can fail to find a worthy relationship in such type of data. We need to seek such a method that gives us accurate measurements in case of

nonlinear data. Kernel canonical correlation (kernel CCA) is a such type of powerful tool.

Kernel CCA has been proposed as a nonlinear extension of CCA (Akaho, 2001, Melzer et al., 2001, Bach and Jordan, 2002). Kernel CCA is thus a nonlinear technique to extract the effective dependent features and to measure the relationship between two or several sets of variables in reproducing kernel Hilbert space (RKHS) instead of the original input space. A similar but different algorithm has been also proposed by Lai and Fyfe (2000).

Over the last decade, kernel CCA has been used for various purposes, including pre-processing for classification, contrast function of independent component analysis, test of independence between two sets of variables (Hardoon et al., 2004, Bach and Jordan, 2002, Huang et al., 2009a, Alzate and Suykens, 2008), and have been applied in many domains such as genomic data, computer graphics and computer-aided drug discovery (Yamanishi et al., 2003, Samarov et al., 2011). The theoretical consistency and restricted kernel CCA has been also discussed (Fukumizu et al., 2007, Hardoon and Shawe-Taylor, 2009, Otopal, 2012).

In recent work, the influence function of kernel principal component analysis (kernel PCA) and a robust kernel PCA has been theoretically derived (Huang et al., 2009b). One observation of their analysis is that kernel PCA with a bounded kernel such as Gaussian is robust in that sense that the influence function does not diverge, while for kernel PCA with unbounded kernels such as polynomial the IF goes to infinity. This can be understood by the boundedness of the transformed data in the feature space by a bounded kernel. While this is not a result for CCA, but for PCA, it is reasonable to expect that kernel CCA with a bounded kernel is also robust. This consideration motivates us to do some empirical studies on the robustness of kernel CCA. It is very important to know how kernel CCA is effected by outliers and to develop measures of accuracy. Therefore, we do intend to study a number of conventional robust estimate and kernel CCA with different functions, but with fixed parameters of the kernel in this chapter.

4.2 Classical and robust canonical correlation analysis

Classical CCA can be seen as the problem of finding basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors

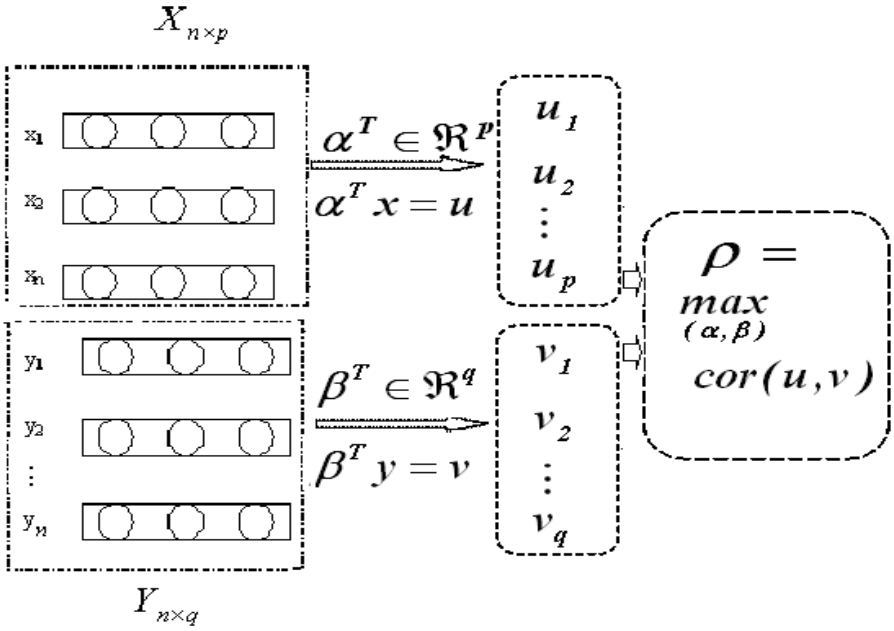

Figure 4.1: The system of CCA.

are mutually maximized. The system of classical CCA is given in Figure 4.1.

Let $\{(X_i, Y_i);\ i = 1, 2, \ldots, n\}$ be the training sample from the pair of multivariate variables $(\mathbf{X}, \mathbf{Y})$ with $\mathbf{X} \in \mathbb{R}^p$ and $\mathbf{Y} \in \mathbb{R}^q$. The classical canonical correlation is to find the directions α and β so that the correlation between the projections of $\mathbf{X}$ onto α and of $\mathbf{Y}$ onto β is maximized such that

$$\begin{aligned} \rho &= \max_{\alpha\in\mathbb{R}^p,\beta\in\mathbb{R}^q} \frac{\mathrm{Cov}[\alpha^{\mathrm{T}}\mathrm{X}, \beta^{\mathrm{T}}\mathrm{Y}]}{\sqrt{\mathrm{Var}[\alpha^{\mathrm{T}}\mathrm{X}]\mathrm{Var}[\beta^{\mathrm{T}}\mathrm{Y}]}} \\ &= \max_{\alpha\in\mathbb{R}^p,\beta\in\mathbb{R}^q} \frac{\alpha^T \mathrm{Cov}_{\mathrm{XY}} \beta^{\mathrm{T}}}{\sqrt{\alpha^T Var_{XX}\alpha}\sqrt{\beta^T var_{YY}\beta}} \end{aligned} \tag{4.1}$$

where Cov[ ] and Var[ ] be the population covariance and variance matrix respectively.

The classical covariance and correlation matrices as well as eigenvector and values are highly sensitive to outlying observations as was shown in the context of Classical CCA. The aim of robust methods is to ensure high reliability and stability of the estimates of statistical characteristics in the case of deviations from the adopted distribution model assumptions (Hampel et al., 1986, Huber and Ronchetti, 2009, Marrona et al., 2006). Robustness consideration is necessary in both supervised and unsupervised problems for every data analysis

technique. An obvious robust approach of canonical correlation is to estimate using robust sample covariance or correlation matrix.

4.3 Measure of robustness

The basic robustness measures are qualitative robustness, influence function and breakdown point. Their are three main concepts to judge an estimator from the viewpoint of robust estimation (Huber and Ronchetti, 2009). Qualitative robustness and breakdown point are the global reliability and influence function is the local reliability concept.

4.3.1 Qualitative robustness

Qualitative robustness measure how much an estimator or test statistic changes with changing of the distribution. The qualitative robustness means small change of a distribution impels small change of an estimator or test statistic. An empirical measure of qualitative robustness has been proposed by Alam et al. (2010).

Definition 4.3.1 (Qualitative robustness index) *To measure the effect of contamination on different estimators at different contamination models we use qualitative robustness index, QRI:*

$$\text{QRI} = \frac{1}{\sum |\xi^{(\epsilon)}_{100\alpha} - \xi^{(\epsilon)\text{c}}_{100\alpha}|}, \tag{4.2}$$

where $\xi^{(\epsilon)}_{100\alpha} = 100^{th}_{\alpha}$ *percentile of the simulated sampling distribution of the estimators at standard model, and* $\xi^{(\epsilon)c}_{100\alpha} = 100^{th}_{\alpha}$ *percentile of the simulated sampling distribution of different estimators at contaminated model. We consider* $\alpha \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.5, 0.9, 0.95, 0.975, 0.99, 0.995\}$ *for our standard model and contaminated model. The range of QRI values is* $[\frac{1}{11}, \infty]$. *A stable estimator has larger QRI value.*

4.3.2 Influence function

The most central concept in Hampel's fundamental contribution to the theory of robustness is the influence function (originally termed as influence curve). The influence function

of an estimator measures how much an individual observation changes the value of the estimator (Hampel et al., 1986).

Definition 4.3.2 *(Empirical influence function (sensitivity curve)). Let us consider an estimator* $\{T_n, n \in \mathbb{N}\}$ *and a sample* $(X_1, X_2, \cdots X_{n-1})$ *of* $n-1$ *observations. Then the sensitivity curve is defined as*

$$\mathrm{SC_n(X) = n[T_n(X_1, X_2, \cdots, X_{n-1}, X) - T_{n-1}(X_1, X_2, \cdots, X_{n-1})]}$$

as a function for X. This is simply a translated and rescaled version of the empirical influence function.

4.3.3 Breakdown point

Breakdown point measure the smallest amount of contamination that can be caused an estimator to take on arbitrarily large aberrant values. This concept is most useful in a finite sample.

Definition 4.3.3 *(Finite sample breakdown point). The finite-sample breakdown point* ϵ^* *of the estimator* T_n *at the sample* $(X_1, X_2, \cdots, X_n)$ *is given by*

$$\epsilon^*(T_n : X_1, X_2, \cdots, X_n) := \frac{1}{n}\{m; max_{i_1,\cdot i_m} sup_{Y_1,\cdots Y_m} |T_n(Z_1, Z_2, \cdots Z_n)| < \infty\}$$

where the sample $(Z_1, \cdots, Z_m)$ *is obtained by replacing the m data points* $X_{i_1}, \cdots, X_{i_m}$ *by the arbitrary values* $Y_1, \cdots, Y_m$.

4.4 Kernel canonical correlation analysis

Given two sets of random variables **X** and **Y**, the aim of kernel CCA is to seek functions in the RKHS, $f_1(\cdot) \in \mathcal{H}_X$ and $f_2(\cdot) \in \mathcal{H}_Y$, for which the correlation (Corr) of the random

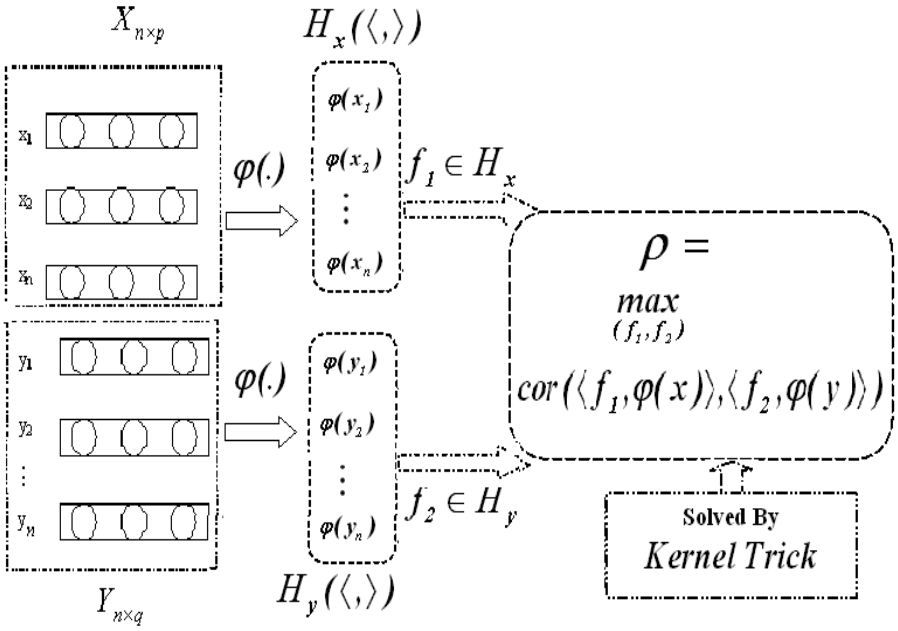

Figure 4.2: The system of kernel CCA.

variables $f_1(\mathbf{X})$ and $f_2(\mathbf{Y})$ is maximized. The optimization problem can be represented as

$$\max_{\substack{f_1\in\mathcal{H}_X, f_2\in\mathcal{H}_Y \\ f_1\neq 0,\, f_2\neq 0}} \mathrm{Corr}(f_1(\mathbf{X}), f_2(\mathbf{Y})). \tag{4.3}$$

The optimizers $f_1(\cdot)$ and $f_2(\cdot)$ are determined up to scale (Bach and Jordan, 2002, Suetani et al., 2006). A system of kernel CCA is given in Figure 4.2.

Using a finite sample, we are able to estimate the desired functions. Given an i.i.d sample $(\mathbf{X}_i, \mathbf{Y}_i)_{i=1}^n$ from a joint distribution F_{XY}, by the representer theorem, we can assume that the functions have the form $f_1(\cdot) = \sum_{i=1}^n a_X^i k_X(\cdot, \mathbf{X}_i)$ and $f_2(\cdot) = \sum_{i=1}^n a_Y^i k_Y(\cdot, \mathbf{Y}_i)$, where $k_X(\cdot, \mathbf{X})$ and $k_Y(\cdot, \mathbf{Y})$ are the associated kernel functions. The kernel Gram matrices are defined by $\mathbf{K}_X := (k_X(\mathbf{X}_i, \mathbf{X}_j))_{i,j=1}^n$ and $\mathbf{K}_Y := (k_Y(\mathbf{Y}_i, \mathbf{Y}_j))_{i,j=1}^n$. We need the centered kernel Gram matrices

$$\mathbf{M}_X = \mathbf{C}\mathbf{K}_X\mathbf{C} \quad \text{and} \quad \mathbf{M}_Y = \mathbf{C}\mathbf{K}_Y\mathbf{C},$$

where $\mathbf{C} = \mathbf{I}_n - \frac{1}{n}\mathbf{B}_n$ with $\mathbf{B}_n = \mathbf{1}_n\mathbf{1}_n^T$ and $\mathbf{1}_n$ is the vector with n ones. The empirical estimate

of (4.3) is then given by

$$\max_{\substack{f_1\in\mathcal{H}_X, f_2\in\mathcal{H}_Y \\ f_1\neq 0,\, f_2\neq 0}} \frac{\widehat{\mathrm{Cov}}(f_1(\mathbf{X}), f_2(\mathbf{Y}))}{[\widehat{\mathrm{Var}}(f_1(\mathbf{X}))]^{1/2}[\widehat{\mathrm{Var}}(f_2(\mathbf{Y}))]^{1/2}} \tag{4.4}$$

where

$$\begin{aligned}
\widehat{\mathrm{Cov}}(f_1(\mathbf{X}), f_2(\mathbf{Y})) &= \frac{1}{n}\mathbf{a}_X^T\mathbf{M}_X\mathbf{M}_Y\mathbf{a}_Y, \\
\widehat{\mathrm{Var}}(f_1(\mathbf{X})) &= \frac{1}{n}\mathbf{a}_X^T\mathbf{M}_X^2\mathbf{a}_X, \\
\widehat{\mathrm{Var}}(f_2(\mathbf{Y})) &= \frac{1}{n}\mathbf{a}_Y^T\mathbf{M}_Y^2\mathbf{a}_Y.
\end{aligned}$$

It is known that the straightforward implementation with the above Gram matrix expression causes an ill-posed problem, and thus a regularization approach is needed to construct a meaningful estimator (Akaho, 2001, Alam et al., 2010). The penalized optimization problem is given by

$$\begin{aligned}
&\max_{\mathbf{a}_X,\mathbf{a}_Y} \frac{1}{n}\mathbf{a}_X^T\mathbf{M}_X\mathbf{M}_Y\mathbf{b}_Y \\
&\text{subject to} \quad \hat{\mathbf{W}}_X = \frac{1}{n}\mathbf{a}_X^T\mathbf{M}_X^2\mathbf{a}_X + \kappa\mathbf{a}_X^T\mathbf{M}_X\mathbf{a}_X = 1, \\
&\hat{\mathbf{W}}_Y = \frac{1}{n}\mathbf{a}_Y^T\mathbf{M}_Y^2\mathbf{a}_Y + \kappa\mathbf{a}_Y^T\mathbf{M}_Y\mathbf{a}_Y = 1,
\end{aligned} \tag{4.5}$$

where κ is a regularized parameter.

The kernel is used in training and predicting. The parameter of this function can be set to any function of class kernel that computes the inner product in feature space between two vector arguments. In this chapter, we use fixed hyperparameters, inverse bandwidth $s = 1$ of the Gaussian RB kernel, Laplacian kernel, and Polynomial function with $c = 1$ and $d = 2$.

4.5 Experimental results

In this section, we address the results of the simulation, influence function and breakdown plot respectively. We generate multivariate normal (MVN) data by considering two covariance matrices CVM_1 and CVM_2 that are given below. We also draw data from uniform

distribution on $[-\pi, \pi]$ and take *sin* and *cos* function on generated data in different ways. The first half of the variables are treated as $\mathbf{X}$ whereas the rest variables are as $\mathbf{Y}$. We calculate bias, standard error, mean square error, QRI of first canonical correlation at MVN model and contaminated multivariate Normal (CMVN) models as well as transformed data. The results are represented in different tables and graphics.

$$CVM_1 = \begin{bmatrix} 1.0 & 0.8702 & -0.3657 & -0.3896 & -0.4931 & -0.2263 \\ 0.8703 & 1.0 & -0.3529 & -0.5522 & -0.6456 & -0.1915 \\ -0.3657 & -0.3529 & 1.0 & 0.1506 & 0.2250 & 0.0349 \\ -0.3896 & -0.5522 & 0.1506 & 1.0 & 0.6957 & 0.4957 \\ -0.4938 & -0.6456 & 0.22503 & 0.695 & 1.0 & 0.6692 \\ -0.2263 & -0.1915 & 0.03493 & 0.4957 & 0.669 & 1.0 \end{bmatrix} \text{ and}$$

$$CVM_2 = \begin{bmatrix} 1.000 & 0.505 & 0.569 & 0.602 \\ 0.505 & 1.000 & 0.422 & 0.467 \\ 0.569 & 0.422 & 1.000 & 0.926 \\ 0.602 & 0.467 & 0.926 & 1.000 \end{bmatrix}$$

4.5.1 Simulation results

In this section, we report a simulation study with different population canonical correlation coefficients, ρ. We take five different sample sizes $n \in \{50, 500, 1000, 1500, 2000\}$ using MVN and CMVN. To perform in simulation study, we consider three experiments as follows.

Experiment-1 (Data generated from multivariate normal and MVN with 6% contamination). In this experiment, we generate data from the MVN model as an ideal model and MVN with 6% fixed outliers as contaminated model. We consider five sample sizes $n \in \{50, 500, 1500, 2000\}$ that are replicated 2000 times. In case of $n = 1500$ and 2000 we take fewer replications to avoid much calculation and time. We calculate bias, mean square error (MSE) and QRI which are given in the Table 4.1 and Table 4.2 respectively.

From Table 4.1, we see in both cases, bias and MSE the performance of classical measure is undoubtedly better than its counterparts at ideal model, whereas at contaminated models the situation is opposite, i.e., it is worst. When robust measure and kernel measures are compared, it is found that the robust measure has clearly the edge over kernel measures

Table 4.1: Bias and mean square error of simulated data (bold numbers indicate the minimum value).

		Bias					MSE				
		Sample Sizes, n					Sample Sizes, n				
Estimators	Model	50	500	1000	1500	2000	50	500	1000	1500	2000
	MNV	**0.016**	**0.0023**	**0.0012**	**0.0001**	**0.0004**	0.0503	**0.0187**	0.0115	**0.0084**	0.0089
CC	MVNC	0.6667	0.6628	0.6628	0.6572	0.6527	0.4761	0.4463	0.4463	0.4375	0.4319
	MNV	0.0487	0.0038	0.0013	0.0021	0.0011	0.0757	0.0243	0.013	0.0090	0.0095
RC	MVNC	**0.0473**	**0.0007**	**0.0008**	**0.0013**	0.009	0.0734	0.0135	0.0134	0.0150	0.0095
	MNV	0.1880	0.0361	0.0137	0.0070	0.0047	0.0494	0.0381	0.0129	0.0086	**0.0085**
KG	MVNC	0.1910	0.0154	0.0156	0.0123	**0.0087**	0.0486	**0.0133**	**0.0129**	0.0120	0.0083
	MNV	0.2044	0.1739	0.0602	0.0418	0.0312	0.0229	0.0543	0.0253	0.0088	**0.0083**
KL	MVNC	0.2044	0.0652	0.0654	0.0408	0.0294	**0.0418**	0.0258	0.0255	0.0166	**0.0075**
	MNV	0.0919	0.0119	0.0051	0.0211	0.0013	**0.0470**	0.0231	**0.0114**	**0.0084**	0.0088
KP	MVNC	0.1812	0.0457	0.0456	0.0522	0.0678	0.0479	0.0168	0.0164	0.0147	0.0157

at both models regarding bias, but in terms of MSE all kernel measures, specially KG, show superior performances. At the contaminated model KG has a smaller MSE than the robust one at all the sample sizes while at the uncontaminated model it performs better thrice out of five times.

In Table 4.2, we observe that both robust and kernel methods demonstrate more stable behavior (bold number indicates the maximum value) than classical methods at all the sample sizes and the robust one is the best.

Table 4.2: The value of qualitative robustness index.

	Sample Sizes, n				
Estimates	50	500	1000	1500	2000
CC	0.138	0.136	0.137	0.138	0.140
RC	16.52	**3.59**	**98.76**	**40.37**	**42.68**
KG	**22.34**	1.73	40.43	8.37	20.35
KL	∞	1.12	18.4325	5.36	11.91
KP	0.982	1.27	1.82	1.69	1.31

Experiment-2 (Data generated from multivariate normal and transformation with

population canonical correlation, $PCC = 0.79$ **and** $PCC = 0.63$**).**

In this experiment we generate data from CVM_1 having first $PCC = 0.79$ and take transformation on **X**-set data by $\sin(2X_1)$, $\cos(X_2)$, $\sin(3X_3)$ and **Y**-set data by $\cos(2Y_1)$, $\sin(Y_2)$, $\cos(3Y_3)$. Similar work is done for CVM_2 with first PCC = 0.63 taking transformations, $\sin(x_1)$, $\sin(3x_2)$ and $\cos(2y_1)$, $\cos(3y_2)$ for set **X** and **Y**, respectively. We sketch the box plots using first canonical coefficient of 2000 simulated samples for five estimators, which are represented in Figure 3(a), and Figure 3(b), respectively.

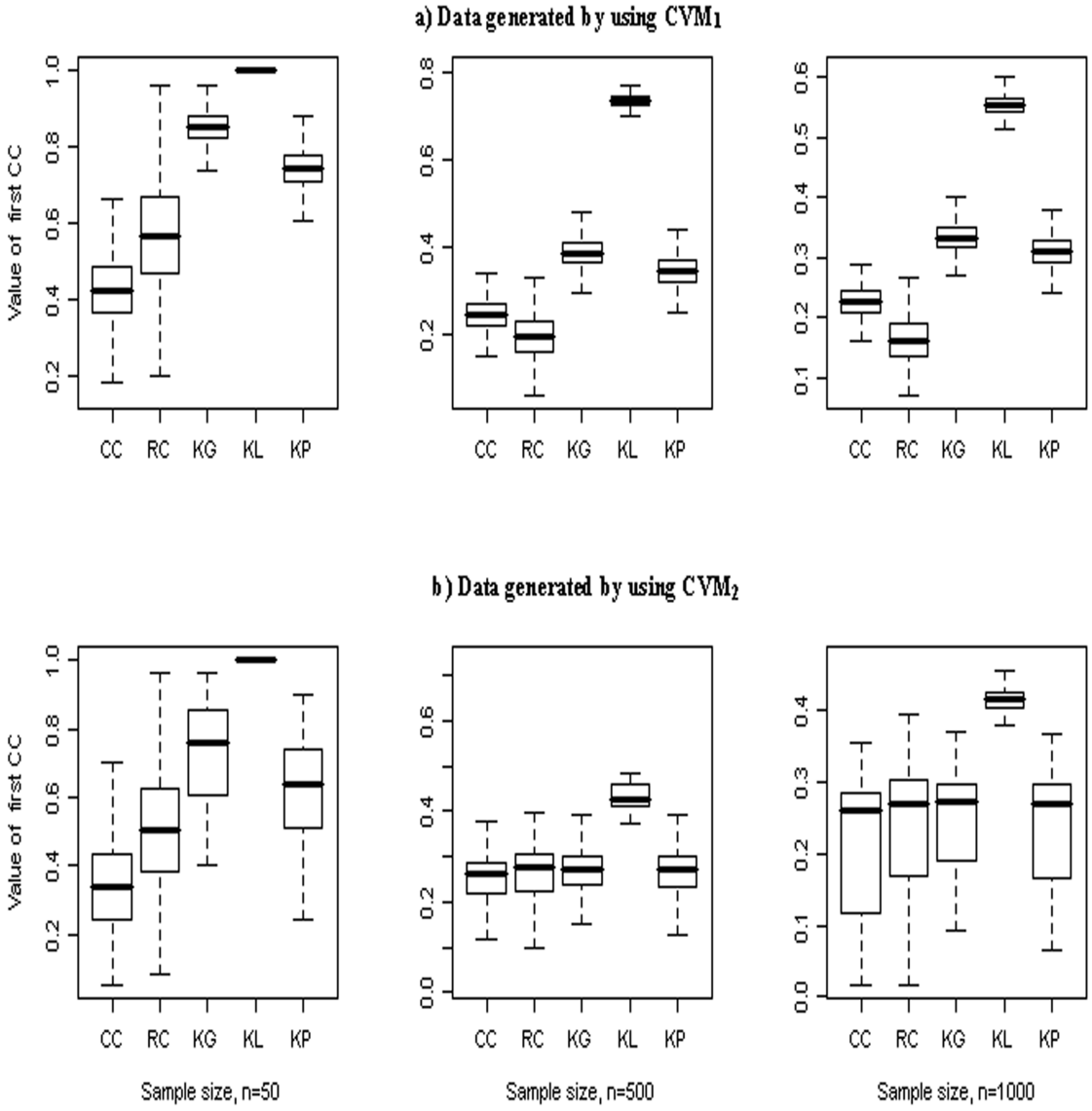

Figure 4.3: Box plots of canonical correlation coefficient of five estimators using population canonical correlation, PCC = 0.79 and PCC = 0.63.

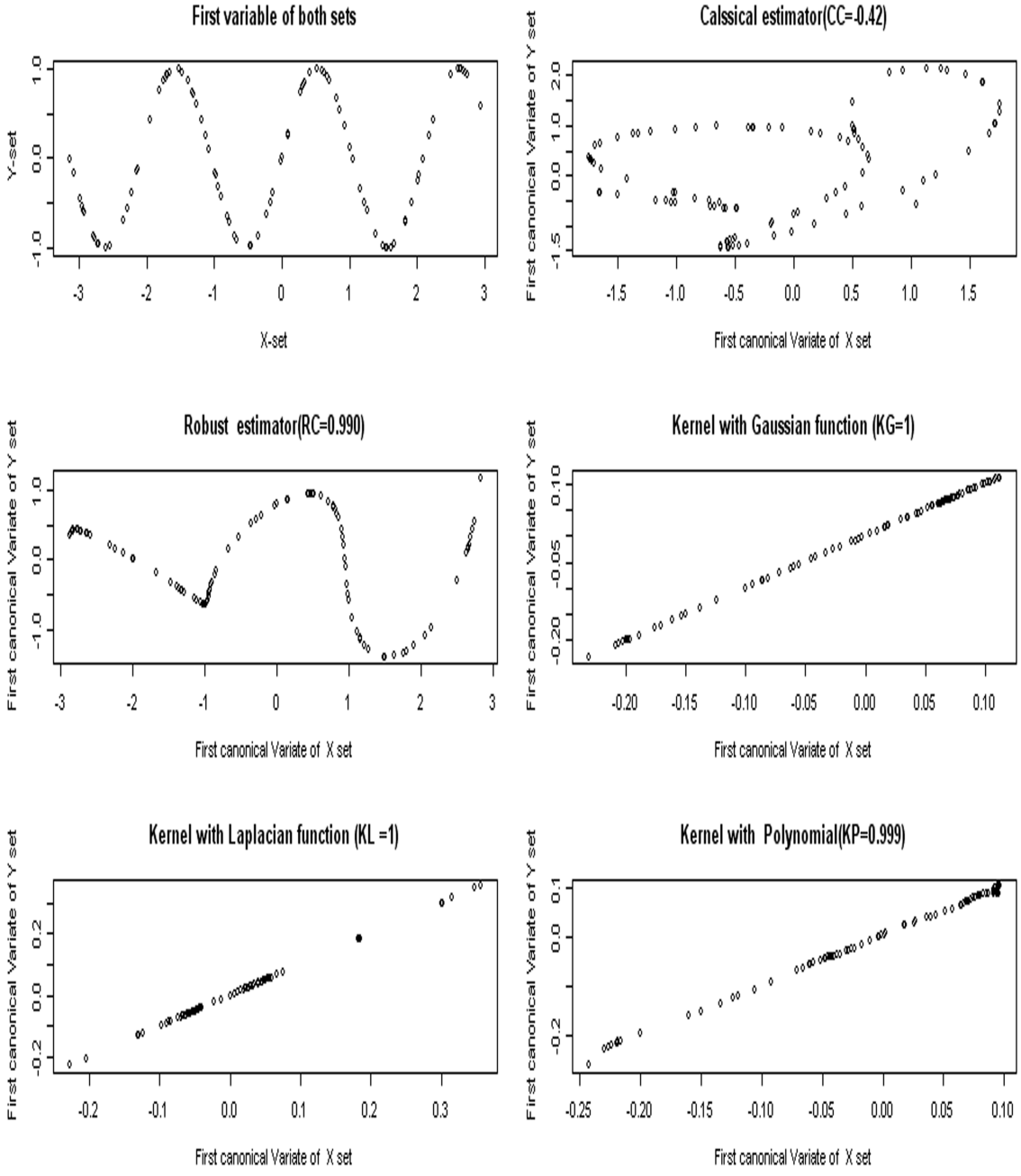

Figure 4.4: Scatter plots of (X_1, Y_1) (top left) and first canonical variates

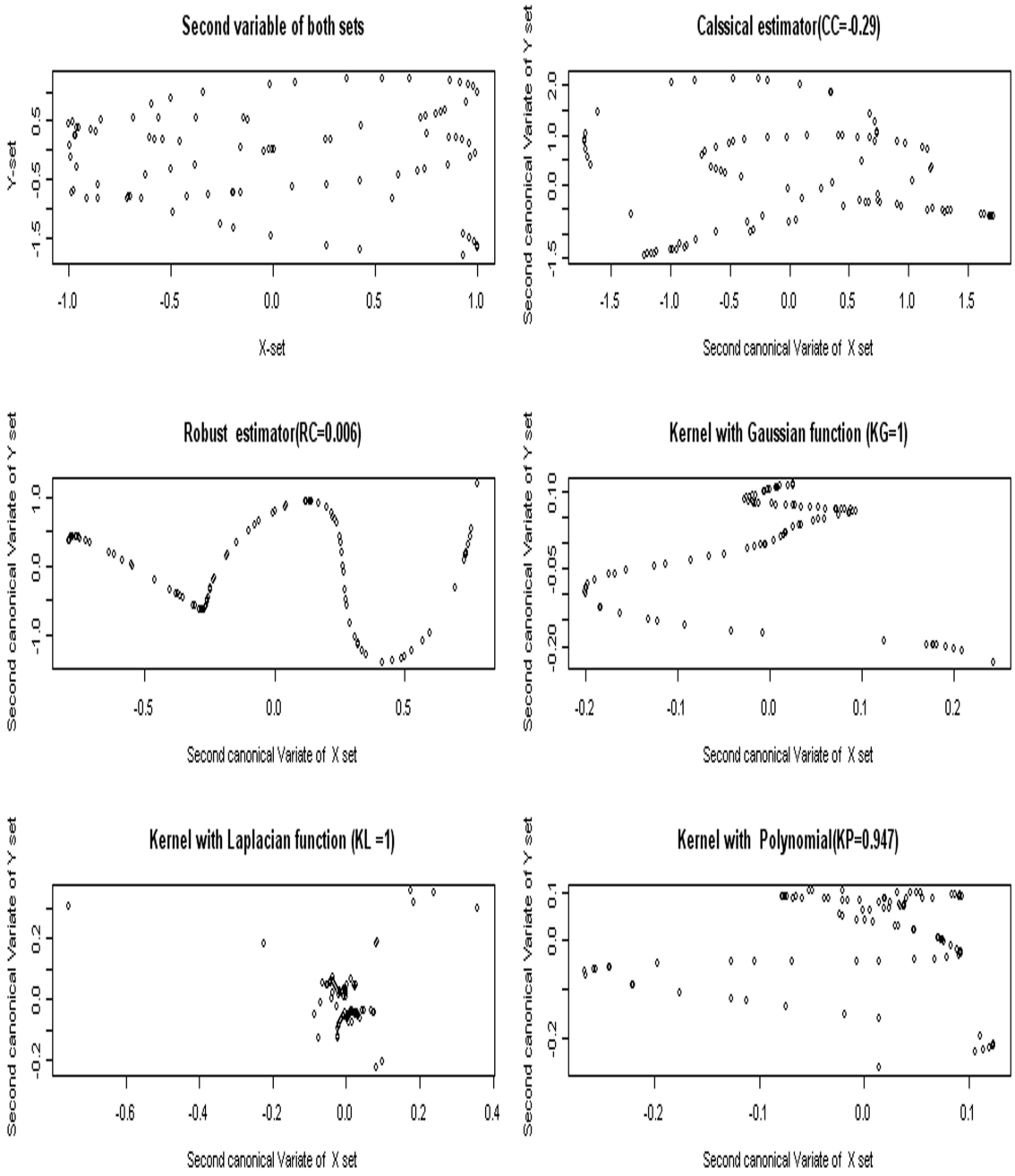

Figure 4.5: Scatter plots of (X_2, Y_2) (top left) and 2nd canonical variates

From the Figure 4.3 we observe that in both cases of transformed data classical and robust method fail equally to capture the nonlinear relationship in the data, whereas kernel CCA, specially KG and KL succeed in detecting the relationship with smaller variation and greater central values.

Experiment-3 (Generated from uniform distribution). In this experiment, we take data according to Akaho (2001), but sample size 100. At first θ is computed from the uniform distribution with range $[-\pi, \pi]$. After that a pair of two dimensional variables $\mathbf{X}$ and $\mathbf{Y}$ are generated by $\mathbf{X} = [\theta, \sin 3\theta]$ and $\mathbf{Y} = e^{\theta/4}[\cos 2\theta, \sin 2\theta]$. Figure 4.4 and Figure 4.5 show that the classical method fails completely to address the relationship whereas robust measure performs better. On the other hand, the kernel method offers us the best performance, providing linear relationship, especially for first canonical variates. Gaussian kernel (KG) and Laplacian kernel are found as the best ones in providing us with linear relationship.

4.5.2 Sensitivity analysis

In this experiment, we perform a simulation study to confirm some aspects of our findings with the help of sensitivity curves. First, we make 200 samples of size, $n = 50, n = 500$ and $n = 1000$ from the MVN using CVM_1 and compute sensitivity curve, $SC_n(v, T_n; V)$ for each of these samples. We also consider an outlier $v = 500, 1000, 1500$ and $0.5, 0.1, 0.15$ for x-set and y-set respectively. That is, the final sample sizes are $n = 51, n = 501$ and $n = 1001$. Box plots of these 200 numbers are given in Figure 4.6(a), for MVN using CVM_1. We repeat the work by taking transformation on each variable as $sin(2x_1)$, $cos(x_2)$, $sin(3x_3)$ and $cos(2y_1)$, $sin(y_2)$, $cos(3y_3)$ for $\mathbf{X}$-set and $\mathbf{Y}$-set respectively; box plots of the measures are given in Figure 4.6(b). Both figures uphold acute non-robustness of classical measure. The robust measure is more affected when data is transformed. KG and KL share the best performance.

4.5.3 Breakdown analysis

In the experiment we discuss the largest amount of contamination (proportion of atypical points) in a data set that an estimator may tolerate, i.e., it still gives some information about

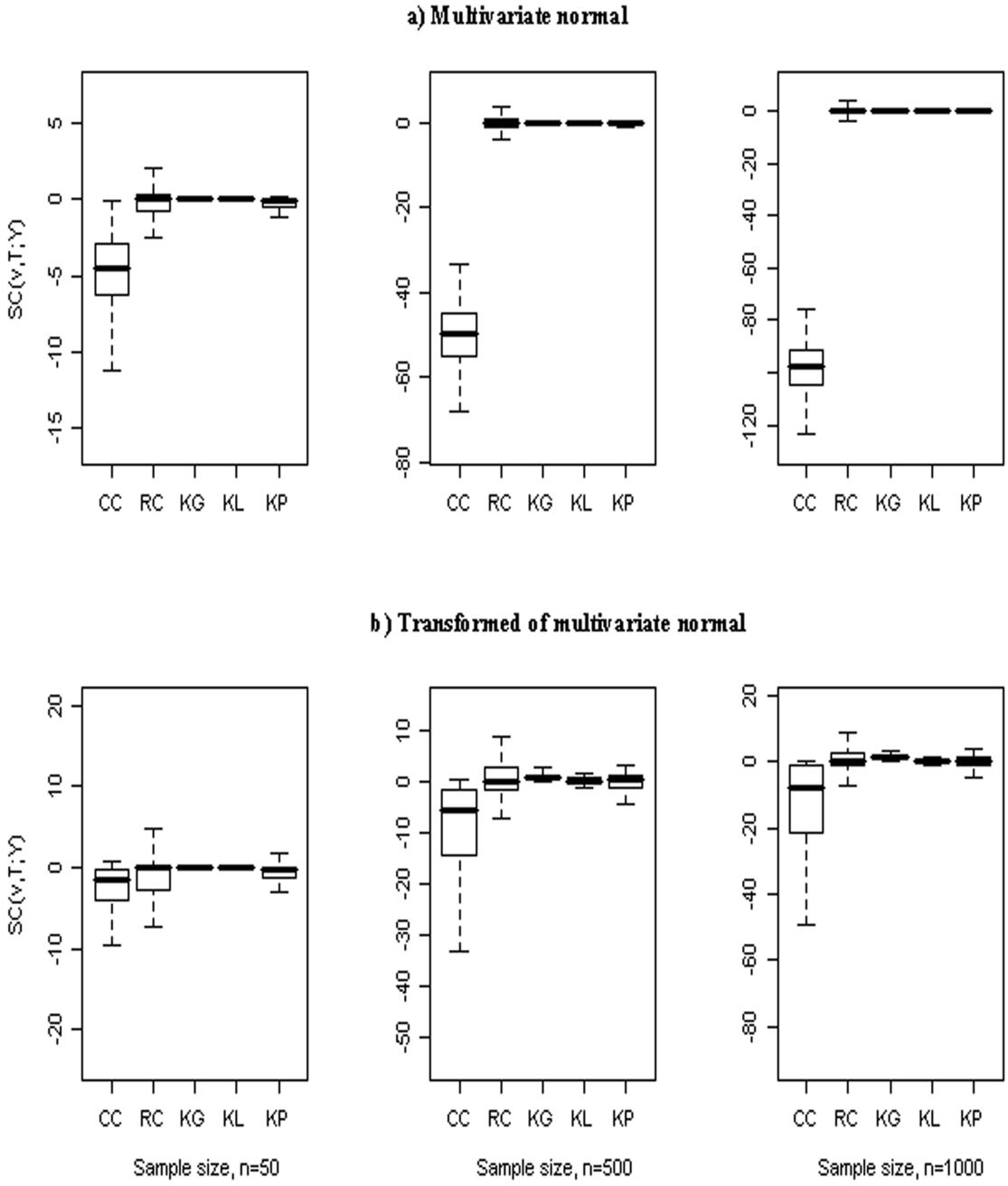

Figure 4.6: Box plots of sensitivity value over 200 samples a) multivariate normal data and b) transform multivariate normal data.

the parameter. We generate 250 samples of size 500 from a multivariate normal distribution using CVM_2 and contaminated distribution by 9 times CVM_2 with $(0, 1, 5, \ldots 30\%)$ contamination. We present the percentage of contamination and the mean squared errors in horizontal axes and in vertical axes respectively. The different lines correspond to different estimators (C for classical; R, robust; G, Gaussian kernel; L, Laplacian kernel and P, polynomial kernel). We repeat this work taking transformation on generated variables. As visible in Figure 4.7(a) and Figure 4.7(b) we observe the effects of replacing several data values by outliers. From the figures it is evident that in case of linear data, the robust measure is the best one, closely followed by G and L; but kernel estimators, especially Laplacian kernel is the best for transforming data.

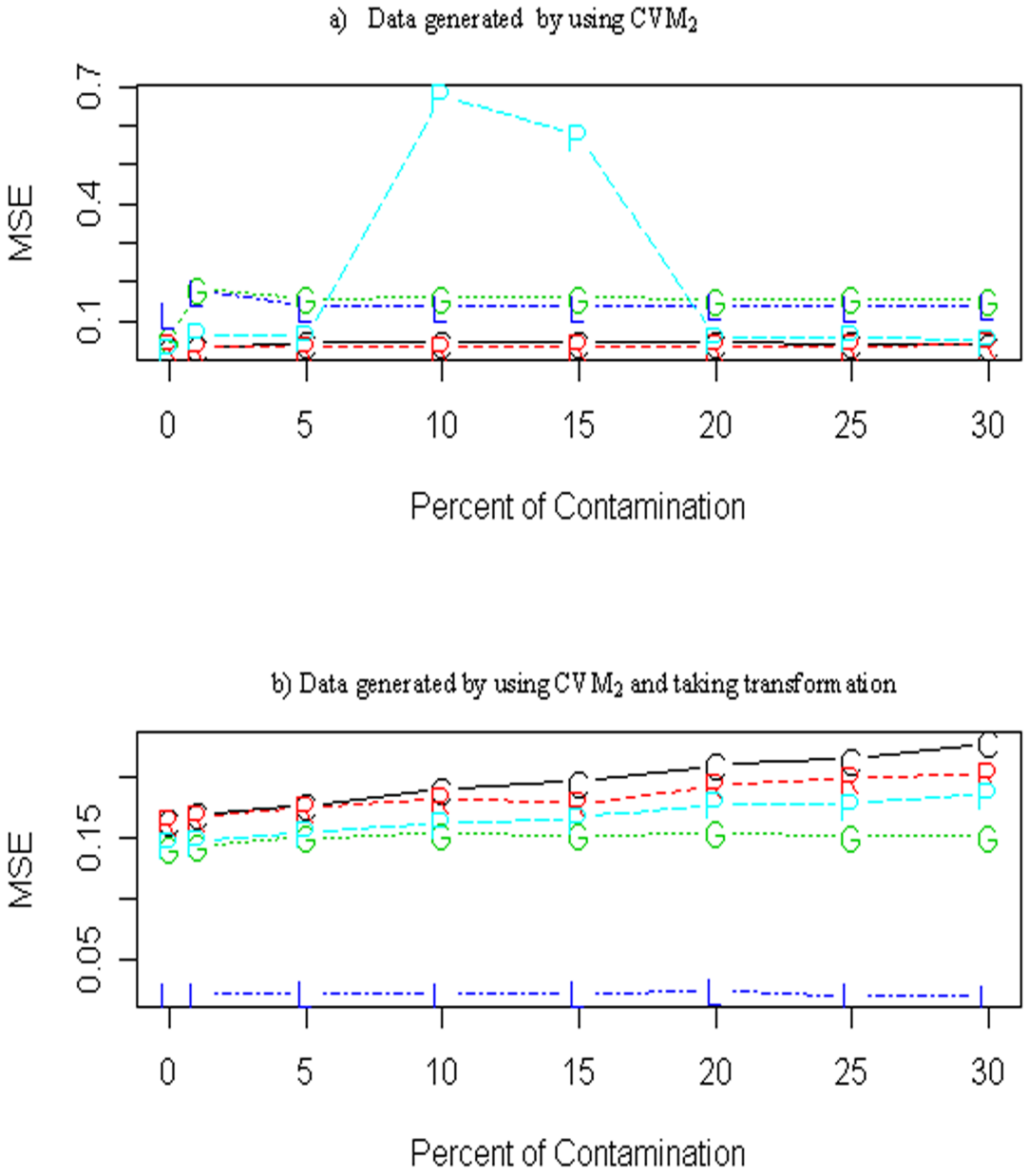

Figure 4.7: Breakdown plots for first canonical correlation coefficient.

Chapter 5

Higher-order Regularized Kernel Canonical Correlation Analysis

5.1 Motivation

Kernel methods have been successfully used in various data analysis as a technique to extract nonlinear structure from data with positive definite kernels. The principle is that any low-dimensional nonlinear structure may be more easily discovered when it is embedded in a larger, even infinite dimensional space. For this purpose, data are transformed from the original input space, where linear methods may not work well, into a feature space via a feature map where linear methods are expected to work better.

To obtain desirable results with kernel methods, in practice, appropriate choice of kernels and other associated parameters in the methods is indispensable. For supervised learning such as the support vector machine, the cross-validation is one of the most popular and useful ways of choosing the kernel and parameters (Arlot, 2010, Woen and Perry, 2009, Stone, 1974). On the other hand, there are no general well-founded methods available for unsupervised learning such as kernel principal component analysis and kernel canonical correlation analysis (kernel CCA) (Alam and Fukumizu, 2011), and some heuristics have been often used for choosing the parameters of kernel CCA (Huang et al., 2009a, Hardoon and Shawe-Taylor, 2009). In this chapter kernel CCA refer as a standard kernel CCA

The first goal of this chapter to discusses application of the cross-validation approach to choosing the kernel and its parameters for the standard kernel CCA. The correlation

value in the standard kernel CCA is not necessarily appropriate for the objective function for cross-validation. Alternatively, based on the known formulation of CCA (and standard kernel CCA) as an alternating regression problem (Breiman and Friedman, 1985, Shawe-Taylor and Cristianini, 2004), it is possible to apply the cross-validation of the prediction errors for choosing the kernel and parameters. Nonetheless, this approach still has a serious problem in the standard kernel CCA: it is observed that with the Gaussian RBF kernel the cross-validation based on the prediction error results in a decreasing function of the inverse bandwidth, and the obtained features corresponding to a small cross-validation error result in an ill-posed solution with all the data concentrated with a few points, as demonstrated in Section 5.2. Such unfavorable results are caused by the fact that the constraints in the standard kernel CCA are given by the 2nd order statistics (e.g., variance) of the canonical variates; this would suffice for a complete statistical description of the Gaussian distribution, but not in general. With the rich function classes given by positive definite kernels, we need much stronger constraints to regulate the canonical variate sufficiently to make the cross-validation applicable.

The second goal of this chapter to propose a kernel CCA subject to the higher-order constraints (see Section 5.4). In the proposed method, not only the 2nd order moment, but also the 4th order moment of canonical variates is constrained to select the tuning parameters using the cross-validation technique. Namely, in addition to the standard constraints of unit variance, we regulate the 4th order moments close to 3; this value is based on the theoretical result showing that an one-dimensional projection of high dimensional data is close to Gaussian random variable (Diaconis and Freedman, 1984). We demonstrate the effectiveness of the proposed higher-order regularized kernel CCA, combined with the cross-validation, in measuring the relationship and extracting effective features for classification using various synthesized and real world problems.

5.2 Cross-validation for the standard kernel canonical correlation analysis

For the standard kernel CCA with Gaussian RBF kernel, we need to select a proper inverse bandwidth s and a regularization parameter κ. It is well known the parameter s has a strong

influence on the result of kernel CCA. A guideline to select the regularization parameter has been proposed by Hardoon *et al.* (Hardoon et al., 2004) and a heuristic technique has been also used for choosing the bandwidth (Hardoon and Shawe-Taylor, 2009). To the best of our knowledge, however, a well-founded technique for choosing the parameters has not yet been established.

The cross-validation (CV) approach is popularly used for choosing parameters of kernel methods, such as the bandwidth parameter in Gaussian kernel, especially in supervised learning. Note that it is not possible to apply the leave-one-out CV with canonical correlation value, since the correlation is not computable with one data. While the CV with the canonical correlation value has been used for choosing the bandwidth in kernel CCA (Suetani et al., 2006), where very dense data from a chaotic dynamics are discussed, it is not easy in general to obtain reliable canonical correlation values by the k-fold CV for small data points.

It is known that CCA and kernel CCA can be regarded as an alternating regression (Breiman and Friedman, 1985, Shawe-Taylor and Cristianini, 2004). It is easy to observe that the kernel CCA is expressed as

$$\max_{f_1\in\mathcal{H}_X, f_2\in\mathcal{H}_Y} \mathrm{Corr}(f_1(\mathbf{X}), f_2(\mathbf{Y})) = \min_{f_1\in\mathcal{H}_X, f_2\in\mathcal{H}_Y} \|f_1(\mathbf{X}) - f_2(\mathbf{Y})\|^2$$

under the condition $\mathrm{Var}[f_1(\mathbf{X})] = \mathrm{Var}[f_2(\mathbf{Y})] = 1$. With this interpretation, the problem can be cast into a supervised setting, and we can apply cross-validation of the prediction errors for choosing the parameters. The CV is then applied in the following manner for each parameter to be selected. First, calculate the canonical coefficient vectors $\mathbf{a}_X^{-v}$ and $\mathbf{a}_Y^{-v}$ based on the v-th training set (T^{-v}) using standard kernel CCA. They give the estimators $f_1^v(\cdot) = \sum_{i\in T^{-v}} a_X^{-v} k(\cdot, \mathbf{X}_i)$ and $f_2^v(\cdot) = \sum_{i\in T^{-v}} a_Y^{-v} k(\cdot, \mathbf{Y}_i)$. Next, compute the regression error

$$\widehat{PE_v} = \frac{1}{|T^v|} \sum_{\mathbf{X}_j\in T^v} \|\tilde{f}_1^v(\mathbf{X}_j) - \tilde{f}_2^v(\mathbf{Y}_j)\|^2,$$

for the corresponding v-th test set (T^v), where $\tilde{f}_1 = f_1^v/(\widehat{V}_v[f_1])^{1/2}$ and $\tilde{f}_2 = f_2^v/(\widehat{V}_v[f_2])^{1/2}$ are the normalization of f_1^v and f_2^v, respectively, by the empirical variance with the training

data

$$\widehat{V}_v[f_1] = \frac{1}{|T^{-v}|}\sum_{i\in T^{-v}} f_1^v(X_i)^2 - \Big(\frac{1}{|T^{-v}|}\sum_{i\in T^{-v}} f_1^v(X_i)\Big)^2 = \frac{1}{|T^{-v}|}a_X^{-vT} M_X^{-v} a_X^{-v}$$

and similar $\widehat{V}_v[f_2]$. After computing the regression errors in all the candidate parameters, we choose the one that gives the minimum regression error.

CV using correlation or prediction error depends on the data concentration. When the data are concentrated in only a few extreme points with perfect or nearly perfect correlation, on the one hand, the CV error will be very small which satisfied the objective of the standard kernel CCA (maximum correlation of canonical variate) but on the other hand, the canonical variates do not follow any well-posed distribution. In classification problem, we can use CV based on classification rates, but the smallest classification error does not correspond to the high correlated features for the standard kernel CCA, in general.

For the standard kernel CCA, however, the above cross-validation approach does not necessarily choose a good parameter in general. We demonstrate this problem using an example of the nutrimouse data (see Section 5.4) with the Gaussian RBF kernel. In Figures 5.1 we show eight scatter plots of the first canonical variates using eight inverse bandwidths, $s \in \{1, 10, 20, 30, 40, 50, 60, 70\}$, together with the cross-validation errors. As we see, the larger values of inverse bandwidth provide smaller errors (≈ 0), but the solution are ill-posed: high correlation is achieved by the features with most data concentrating on a few points. This example illustrates that a straightforward application of cross-validation for choosing a kernel is not appropriate. The constraints of the variance in the kernel CCA does not regulate sufficient for a large variety of nonlinearity given by different kernels.

In the experimental studies of this chapter, we use only Gaussian RBF kernel, but the most of the arguments in this chapter apply to other popular nonlinear kernels such as Lapacian.

5.3 Distribution of a low dimensional projection

The proposed method (see Section 5.4), we regularize the 4th order moment, so that it is triple of the variance. This is based on the theoretical result by Diaconis and Freed-

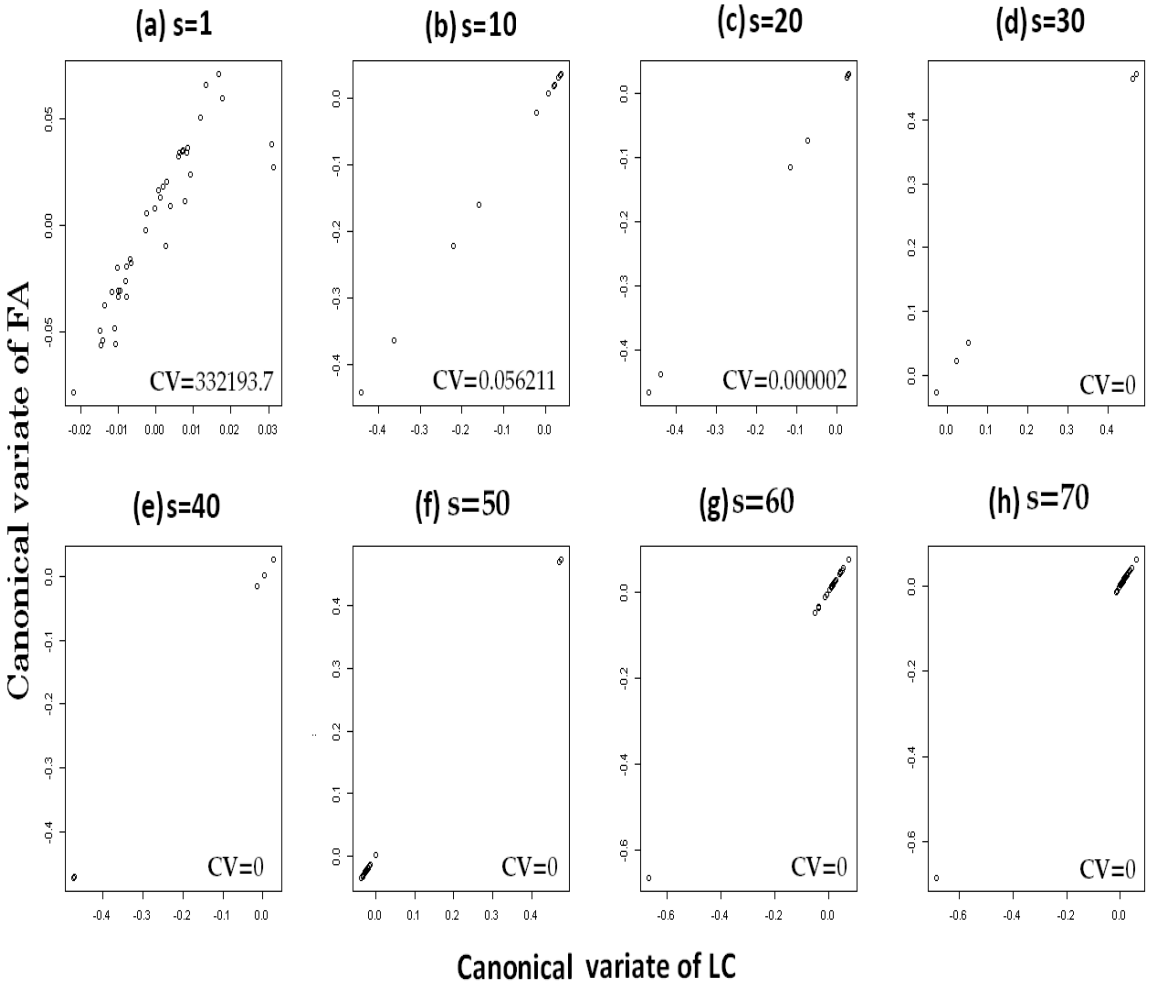

Figure 5.1: Standard kernel CCA: the scatter plots of the 1st canonical variates for the nutrimouse dataset (liver cells and hepatic fatty acids) using the Gaussian RBF kernel with eight inverse bandwidths s and fixed regularization coefficient $\kappa = 10^{-4}$. The 10-fold cross-validation errors are also embedded.

man (Diaconis and Freedman, 1984), which shows the characteristics of the projected low dimensional features from high dimensional data approximately follow the Gaussian distribution in general. While this result does not directly apply to the standard kernel CCA, which constructs the projection depending on data, we use it as a guide to define our penalty on the 4th order moments. By a quantitative analysis, Meckes (2009) has been shown the situation where orthogonal projections are asymptotically Gaussian.

Theorem 5.3.1 (Diaconis and Freedman (1984)) *Given $X_1, X_2, \cdots X_n$ be deterministic vectors in $\mathbb{R}^m$. Suppose that n, m and the X_i depend on a hidden index τ, so that as $\tau \to \infty$, so do n and m. Suppose that $\exists\, \sigma \in (0, \infty)$ and for all $\nu > 0$,*

$$\frac{1}{n}\mathrm{CN}\left\{\mathrm{j} \leq \mathrm{n} : \left|\langle \mathbf{X}_{\mathrm{j}}, \mathbf{X}_{\mathrm{j}}\rangle - \sigma^2 \mathrm{p}\right| > \nu \mathrm{m}\right\} \to 0, \text{ as } \tau \to \infty, \tag{5.1}$$

$$\frac{1}{n^2}\mathrm{CN}\left\{\mathrm{j}, \mathrm{k} \leq \mathrm{n} : \left|\langle \mathbf{X}_{\mathrm{j}}, \mathbf{X}_{\mathrm{k}}\rangle > \nu \mathrm{m}\right|\right\} \to 0, \text{ as } \tau \to \infty, \tag{5.2}$$

where CN{A} *is stand for cardinality number of a set A. Let* $T \in \mathbb{U}^{m-1}$ *is distributed uniformly on the sphere, and consider a random measure* μ_τ^T *with mass* $\frac{1}{n}$ *at the points* $\langle T, \mathbf{X}_1\rangle, \cdots, \langle T, \mathbf{X}_n\rangle$. *The measure,* $\mu_\tau^T \to \mathcal{N}(0, \sigma^2)$ *weekly in probability as* $\tau \to \infty$.

Theorem 5.3.1 says that one dimensional projections of a large number (approximately same lengths and nearly orthogonal) of high-dimensional data vectors are close to Gaussian regardless of the structure of the data. The conditions (5.1) and (5.2) are not too strong; in particular, even though only m vectors can be exactly orthogonal in $\mathbb{R}^m$, the 2^m vertices of a unit cube centered at the origin satisfy condition (5.2) for rough orthogonality (Meckes, 2009).

Dümbgen and Counte-Zerial (2013) have given necessary and sufficient conditions that the sequence of distribution, $(F^m)_{m\geq\ell}$ such the most low ℓ-dimensional orthogonal projections of the probability distribution on m dimensional space F are similar to some distribution G on $\mathbb{R}^\ell$. The limiting distribution is a mixture of centered, spherically symmetric Gaussian distributions.

Theorem 5.3.2 (Dümbgen and Counte-Zerial (2013)) *The following two statements on the sequence,* $(F^{(m)})_{m\geq\ell}$ *are equivalent:*

(a1) There exists a probability measure G on $\mathbb{R}^\ell$ *such that*

$$\Gamma^T F \to_{w,p} G \text{ as } \mathrm{m} \to \infty.$$

(a2) If $\mathbf{X}_1 = \mathbf{X}^{(m)}$ *and* $\mathbf{X}_2 = \mathbf{X}^{(m)}$ *be are independent random vectors with distribution F,*

then

$$\mathcal{L}\left(\frac{\langle \mathbf{X}_1, \mathbf{X}_1 \rangle}{m}\right) \rightarrow_w \mu \quad \text{and} \qquad \frac{\langle \mathbf{X}_1 \mathbf{X}_2 \rangle}{\mathrm{m}} \rightarrow_{\mathrm{p}} 0$$

as $m \rightarrow \infty$ *for some probability measure* μ *on* $[0, \infty)$*, where* $\mathcal{L}(x)$ *is the distribution of* x*.*

The limit distribution G in (a1) is a normal mixture, i,e.,

$$G = \int \mathcal{N}_{\ell,\nu} \, \mu(\mathrm{d}\nu),$$

where $\mathcal{N}_{\ell,\nu}$ *stand for the Gaussian distribution on* $\mathbb{R}^\ell$ *with mean vector* 0 *and covariance matrix* νI_ℓ*.*

A details proof of the Theorem 5.3.1 and 5.3.2 are in Section 2 (Diaconis and Freedman, 1984) and in Section 4 (Dümbgen and Counte-Zerial, 2013), respectively.

5.4 Higher-order regularized kernel CCA (hrKCCA)

We have observed in Section 5.2 that in the standard kernel CCA the 2nd order regularization is not sufficient; for a small bandwidth (large inverse bandwidth) in the Gaussian RBF kernel, the standard kernel CCA gives high correlation but the resulting features are not meaningful with most data accumulating at only a few points. We propose to introduce a penalty on the 4th order moments of the canonical variates for a solution to this ill-posedness.

5.4.1 Method

The 4th order moment of $f_1(\mathbf{X})$ and $f_2(\mathbf{Y})$ are given by $\mu_X^4 = E[(f_1(\mathbf{X}) - E[f_1(\mathbf{X})])^4]$ and $\mu_Y^4 = E[(f_2(\mathbf{Y}) - E[f_2(\mathbf{Y})])^4]$, and their empirical estimates are

$$\begin{aligned} \hat{\mu}_X^4 &= \frac{1}{n} \sum_{i=1}^{n} [(\mathbf{M}_X \mathbf{a}_X)_i]^4 = \frac{1}{n} \mathbf{1}^T \mathbf{f}_1^{(4)}, \\ \hat{\mu}_Y^4 &= = \frac{1}{n} \sum_{i=1}^{n} [(\mathbf{M}_Y \mathbf{a}_Y)_i]^4 = \frac{1}{n} \mathbf{1}^T \mathbf{f}_Y^{(4)}, \end{aligned}$$

where $\mathbf{f}_1^{(4)} = [\bar{f}_{11}^{(4)}, \ldots, \bar{f}_{1n}^{(4)}]^T$ with the centered feature values

$$\bar{f}_{1i}^{(4)} = [(\mathbf{M}_X \mathbf{a}_X)_i]^4 = [\sum_{j=1}^{n} M_{Xij} a_{Xj}]^4,\ i = 1, 2, \ldots, n,$$

and similar for $\mathbf{f}_2^{(4)}$.

We penalize the 4th order moments, $\frac{1}{n}\mathbf{1}^T\mathbf{f}_1^{(4)}$ and $\frac{1}{n}\mathbf{1}^T\mathbf{f}_2^{(4)}$ so that they are close to 3: the motivation is to regulate the distribution of the canonical variates, so that they have approximately the same kurtosis as the standard normal distribution. This is expected to reduce the ill-posedness of the standard kernel CCA with the kernel chosen by CV.

The optimization problem of hrKCCA is then given by

$$\begin{aligned}
&\max_{\mathbf{a}_X, \mathbf{a}_Y} \frac{1}{n}\mathbf{a}_X^T \mathbf{M}_X \mathbf{M}_Y \mathbf{b}_Y \\
&\text{subject to} \\
&\hat{\mathbf{W}}_X = \frac{1}{n}\mathbf{a}_X^T \mathbf{M}_X^2 \mathbf{a}_X + \kappa \mathbf{a}_X^T \mathbf{M}_X \mathbf{a}_X = 1, \\
&\hat{\mathbf{W}}_Y = \frac{1}{n}\mathbf{a}_Y^T \mathbf{M}_Y^2 \mathbf{a}_Y + \kappa \mathbf{a}_Y^T \mathbf{M}_Y \mathbf{a}_Y = 1, \\
&\quad \hat{\mu}_X^4 = \hat{\mu}_Y^4 = 3.
\end{aligned} \tag{5.3}$$

where κ is a regularized parameter. We convert this to a regularization problem

$$\begin{aligned}
L(\mathbf{a}_X, \mathbf{a}_Y) = \frac{1}{n}\mathbf{a}_X^T \mathbf{M}_X \mathbf{M}_Y \mathbf{a}_Y - \nu\Big[\hat{\mathbf{W}}_X - 1\Big]^2 - \nu\Big[\hat{\mathbf{W}}_Y - 1\Big]^2 \\
- \lambda[\hat{\mu}_X^4 - 3]^2 - \lambda[\hat{\mu}_Y^4 - 3]^2,
\end{aligned} \tag{5.4}$$

where ν and λ are regularization coefficients. These coefficients are chosen by cross-validation. For simplicity, we assume the same regularization coefficients for X and Y, and set $\nu = c\lambda$, where c is a fixed trade-off between the 4th order and 2nd order information. As we demonstrate in Section 5.5.1, the results of the proposed method are not so sensitive to the choice of c (Alam and Fukumizu, 2013).

The maximization of Eq. (5.4) can be done with a nonlinear programming method: gradient based unconstrained methods, or penalty methods (Kelley, 1999). We use the

steepest ascent method. The gradient of Eq. (5.4) with respect to $\mathbf{a}_X$ and $\mathbf{a}_Y$ are given by

$$\begin{aligned}\nabla_{a_X} L &= \frac{1}{n}\mathbf{M}_X\mathbf{M}_Y\mathbf{a}_Y - \frac{4c}{n}\lambda\Big[\hat{\mathbf{W}}_X - 1\Big](\mathbf{M}_X^2 + \kappa\mathbf{M}_X)\mathbf{a}_X - \frac{8}{n}\lambda\Big[\frac{1}{n}\mathbf{1}^T\mathbf{f}_X^{(4)} - 3\Big]\mathbf{M}_X\mathbf{f}_X^{(3)},\\ \nabla_{a_Y} L &= \frac{1}{n}\mathbf{M}_Y\mathbf{M}_X\mathbf{a}_X - \frac{4c}{n}\lambda\Big[\hat{\mathbf{W}}_Y - 1\Big](\mathbf{M}_Y^2 + \kappa\mathbf{M}_Y)\mathbf{a}_Y - \frac{8}{n}\lambda\Big[\frac{1}{n}\mathbf{1}^T\mathbf{f}_Y^{(4)} - 3\Big]\mathbf{M}_Y\mathbf{f}_Y^{(3)},\end{aligned}$$

respectively, where $\mathbf{f}_1^{(3)} = [\bar{f}_{11}^3, \ldots, \bar{f}_{1n}^3]^T$ and $\mathbf{f}_2^{(3)}$ are similarly defined to $\mathbf{f}_1^{(4)}$. We call this modified kernel CCA *higher-order regularized kernel CCA* (hrKCCA).

In a similar manner, the regularization problem to calculate the p-th ($p = 1, 2, \ldots, n$) canonical variates such that $\mathrm{Cov}(\mathrm{f}_1^{(\mathrm{p})}(\mathbf{X}), \mathrm{f}_1^{(\mathrm{q})}(\mathbf{X})) = 0$ if $p \neq q$ and 1 if $p = q$ having maximum correlation with $f_2^{(p)}(\mathbf{Y})$ is then given by

$$\begin{aligned}L(\mathbf{a}_{X^p}, \mathbf{a}_{Y^p}) &= \frac{1}{n}\mathbf{a}_{X^p}^T\mathbf{M}_X\mathbf{M}_Y\mathbf{a}_{Y^p} - \nu\Big[\hat{\mathbf{W}}_X - 1\Big]^2 - \nu\Big[\hat{\mathbf{W}}_Y - 1\Big]^2\\ &- \lambda\Big[\hat{\mu}_{X^p}^4 - 3\Big]^2 - \lambda\Big[\hat{\mu}_{Y^p}^4 - 3\Big]^2 + \sum_{i=1}^{p-1}\gamma_{X_i}\mathbf{a}_{X^p}^T\mathbf{M}_X\mathbf{a}_{X_i^p} + \sum_{i=1}^{p-1}\gamma_{Y_i}\mathbf{a}_{Y^p}^T\mathbf{M}_Y\mathbf{a}_{Y_i^p},\end{aligned} \tag{5.5}$$

where $\mathbf{a}_{X^p}$ and $\mathbf{a}_{Y^p}$ be the pth directions of $\mathbf{X}$ and $\mathbf{Y}$, respectively. For simplicity, we assume the same regularization coefficients for all variables: $\gamma_{X_1} = \gamma_{X_2} = \ldots = \gamma_{X_{(p-1)}} = \gamma_{Y_1} = \gamma_{Y_2}, = \ldots = \gamma_{Y_{(p-1)}} = \gamma$. It is possible to select the regularization coefficient γ by cross-validation. In all experiments, we use $\gamma = \lambda$ and λ is selected by cross-validation.

5.4.2 Kernel choice for hrKCCA

The higher-order regularized kernel CCA can be suitably combined with cross-validation for choosing kernels and parameters. As described in Section 5.2, the leave-one-out or k-fold cross-validation for small data points are not applicable for kernel CCA. In this work, we employ the mean square errors to perform the cross-validation for hrKCCA. The algorithm for selecting the hyperparameters (bandwidth and regularization coefficient) is described in Figure 5.2, in which only the choice of parameter in the Gaussian kernel is shown for simplicity, but other parameters can be chosen in a similar way.

step 1. Input the dataset $(\mathbf{X}_i, \mathbf{Y}_i)_{i=1}^n$.

step 2. Choose $s_1, \ldots, s_L$ to be L values of the free parameter s in Gaussian kernel.

step 3. Consider C-fold cross-validation, and partition the data into C portions ($v = 1, 2, \ldots, C$).

step 4. For each $\ell = 1, \ldots, L$, do the following procedure:

(i) Compute the centered Gram matrices $\mathbf{M}_X$ and $\mathbf{M}_Y$ using Gaussian RBF kernel $k(\mathbf{X}_i, \mathbf{X}_j) = \exp(-\mathrm{s}_\ell \|\mathbf{X}_\mathrm{i} - \mathbf{X}_\mathrm{j}\|^2)$.

(ii) Optimize Eq.(5.4) based on the training set (T^{-v}) to find the canonical coefficient vectors $\mathbf{a}_X^{-v}$ and $\mathbf{a}_Y^{-v}$. The canonical covariates are $f_1^v(\mathbf{X}_{vj}) = \sum_{i \in T^{-v}} a_X^{-v} k(\mathbf{X}_j, \mathbf{X}_i)$ and $f_2^v(\mathbf{Y}_{vj}) = \sum_{i \in T^{-v}} a_Y^{-v} k(\mathbf{Y}_j, \mathbf{Y}_i)$.

(iii) Evaluate the prediction error $\widehat{PE_v(s_\ell)} = \|\tilde{f}_1^v(\mathbf{X}_v) - \tilde{f}_2^v(\mathbf{Y}_v)\|^2$ for the vth test set (T^v), where $\tilde{f}_1^v(\mathbf{X}_v)$ and $\tilde{f}_2^v(\mathbf{Y}_v)$ are normalized by the variance of the training data.

(iv) Average the C prediction errors: $\widehat{PE}_{CV}(s_\ell) = \frac{1}{C} \sum_{v=1}^{C} \widehat{PE_v(s_\ell)}$.

step 5. Choose the value of s, $\widehat{s_{CV}}$, that minimizes the prediction error, i.e., $\widehat{s_{CV}} = \arg\min_{s_\ell} \widehat{PE}_{CV}(s_\ell)$.

Figure 5.2: Algorithm of the higher-order regularized kernel CCA.

5.4.3 Computational issues

In the proposed method, we need to compute the 4th order moments. Since the kernel CCA has been based on only the 2nd order moments, it is obvious that the time complexity of the proposed method will be higher than the kernel CCA. The number of iterations for convergence of the algorithm depends on the data; in the extreme case, if we consider the same dataset for both the variables, we will obtain the perfect correlation with a few numbers of iterations. The computational time increases linearly in the numbers of iteration, while in each iteration the computational cost is $O(n^2)$ where n is the sample size. Note that we do not need matrix inversion in the gradient method. To illustrate the computational cost, the results for three different data sizes and three different numbers of iteration (I) using an example 3 (E_3, details in Section 5.4.1) are tabulated in Table 5.1. The configuration of the computer is Intel (R) Core (TM) i7 CPU 920@ 2.67 GHz., memory 12.00 GB and 64-bit operating system. We have used 'kernlab' package in R program for implementation of the standard kernel CCA (KCCA) with fixed regularization coefficient, $\kappa = 10^{-4}$. We could

choose the regularization coefficient by cross-validation, but have fixed it for simplicity.

Table 5.1: Time complexity (in second) for example 3: the proposed kernel CCA with three numbers of iterations (I) and the standard kernel CCA (KCCA). The Gaussian kernel is used for the both methods, and n is the sample size.

$\#I/n$	100	500	1000
100	0.44	32.33	272.63
500	2.07	159.90	1219.07
1000	4.14	325.00	2683.19
KCCA	0.11	7.76	73.35

5.5 Experiments

We experimentally verify the effectiveness of the proposed method in extracting the dependent features in comparison with the kernel CCA. We compare the performance of the proposed method with the original kernel CCA using the synthesized examples and real world datasets. In addition, the classification results using low dimensional subspace (1 and 2) of the proposed method are compared with other existing classification techniques. The number of data and dimensions of response ($\mathbf{Y}$) and explanatory ($\mathbf{X}$) variables for all experimental datasets are given in Table 5.2. In all experiments we fixed the regularization $\kappa = 10^{-4}$.

5.5.1 Synthesized examples

We use four synthesized data which have different marginal distributions and nonlinear transformations. In the following examples, i and j correspond to ith data point and jth dimension, respectively.

Example 1 (E_1: Gaussian distribution and $\log_e$ transformation). Given multivariate normal data, $\mathbf{Z}_i \in \mathbb{R}^{12} \sim \mathbf{N}(\mathbf{0}, \Sigma)$ ($i = 1, 2, \ldots, 500$) where Σ is taken from Johnson and Wichern (p. 555)(Johnson and Wichern, 2007). We divide $\mathbf{Z}_i$ into two sets of variables ($\mathbf{Z}_{i1}$,$\mathbf{Z}_{i2}$), and use the first five variables of $\mathbf{Z}_i$ as the explanatory variable $\mathbf{X}$, and *log* transformation of the absolute value of the remaining seven variables ($\log_e(|\mathbf{Z}_{i2}|)$) as the response variable $\mathbf{Y}$.

Table 5.2: The configuration of datasets along with purposes (estimate of dependence feature (EDF), measure of association (MA) and estimate of low dimensional space (ELDS)) of all experimental datasets.

			Dim.		
		#of Data	**X**	**Y**	Purpose
Artificial	E_1	500	5	7	EDF
	E_2	300	500	500	
	E_3	1000	1000	1000	EDF
	E_4	200	35	25	
Real data	$Nutrimouse, D_1$	40	120	21	MA
	$DBWorld, D_2$	64	4702	242	
	$Psychological, D_3$	600	3	5	
	$Carbig, D_4$	390	3	2	
	$Wine, D_5$	178	13	3	ELDS
	$BUPA, D_6$	345	6	2	
	$Diabetes, D_7$	145	5	3	
	$Subjects, D_2$	64	4702	2	
	$Bodies, D_2$	64	242	2	
	$KTH\ Human\ actions, D_8$	600	384000	6	
	UMD, D_9	48	6144000	12	

Example 2 (E_2: Uniform marginal and periodic transformation). We use uniform $[-\pi, \pi]$ marginal distribution, and transform the data by two periodic sin and cos functions to make **X** and **Y**, respectively, with additive Gaussian noise:

$$\begin{aligned} Z_i &\sim U[-\pi, \pi], \qquad \eta_i \sim N(0, 0.05), \ \ i = 1, 2, \ldots, 300, \\ \mathbf{X}_{ij} &= \sin(j * Z_i) + \eta_i, \\ \mathbf{Y}_{ij} &= \cos(j * Z_i) + \eta_i, \qquad j = 1, 2, \ldots, 500. \end{aligned}$$

Example 3 (E_3: Laplace marginal, periodic and nonlinear transformation). The periodic transformation by cosine function for one set of variables **Y** and the cubic transformation for the other set of variables **X** are generated from the marginal Laplace $[0, 1]$ distribution:

$$\begin{aligned} Z_i &\sim L[0, 1], \qquad \eta_i \sim N(0, 0.05), \ \ i = 1, 2, \ldots, 1000, \\ \mathbf{X}_{ij} &= j * Z_i^3 + \eta_i, \\ \mathbf{Y}_{ij} &= \cos(j * Z_i) + \eta_i, \qquad j = 1, 2, \ldots, 1000. \end{aligned}$$

For cross-validation in Examples 1-3, we consider six inverse bandwidths for the three examples, $s \in \{90, 100, 110, 120, 130, 140, 150\}$ for E_1, $s \in \{1, 10, 20, 30, 40, 50\}$ for E_2, and $s \in \{1, 25, 40, 50, 60, 75\}$ for E_3. The same set $\lambda \in \{0.50, 0.75, 0.90, 1.05, 1.25, 1.50\}$ is used for the regularization parameter. The 10-fold cross-validation errors are tabulated in Table 5.3. The smallest cross-validation errors corresponding to the inverse bandwidth, $s = 130$, $s = 30$, and $s = 25$ for E_1, E_2 and E_3, respectively, and the same regularization coefficient $\lambda = 1.05$.

To see the influence of the trade-off parameter c for $\nu = c\lambda$ between the variance and the 4th order moment (see Section 5.4.1), we have visualized in Figures 5.3 the first canonical variates with the three values $c = 10, 20$, and 30 for all the three examples. We can observe that the results of the proposed method are not so sensitive to the value of c, if it is large enough. We thus fix $c = 10$ in all of the experiments in this paper to reduce the number of free parameters chosen by CV. We can also see that the standard kernel CCA corresponding to the smallest cross-validation error provides ill-posed solutions: only two concentrated points with perfect correlation. In contrast, the proposed method provides well-posed solutions showing a high correlation (0.988, 0.993 and 0.975) with reasonable distributions.

Example 4 (E_4: Uniform marginal and nonlinear transformation). We demonstrate that the CV errors of the standard kernel CCA decreases as the inverse bandwidth of Gaussian kernel increases. We consider a linear set of variables $\mathbf{X}$ and nonlinear set of variables $\mathbf{Y}$ having the same uniform $[-2, 2]$ marginal distribution as follows:

$$
\begin{aligned}
Z_i &\sim U[-2, 2], \qquad \eta_i \sim N(0, 0.05), \;\; i = 1, 2, \ldots, 200, \\
\mathbf{X}_{ij} &= j * Z_i + \eta_i, \qquad j = 1, 2, \ldots, 25, \\
\mathbf{Y}_{ij} &= j * Z_i^2 + \eta_i, \qquad j = 1, , 3 \ldots, 9, \\
\mathbf{Y}_{ij} &= j * Z_i^3 + \eta_i, \qquad j = 2, 4 \ldots, 10.
\end{aligned}
$$

We have generated 100 samples. We take eight inverse bandwidths: $s_1 = 225, s_2 = 250, s_3 = 275, s_4 = 300, s_5 = 325, s_6 = 350, s_7 = 375, s_8 = 400$, and calculate the 10-fold cross-validation errors from each sample. The box plots and line plots (inset) using mean values for the kernel CCA and the proposed method using two regularization coefficients λ

Table 5.3: Cross-validation errors of the three examples ($E_1 - E_3$) using different inverse bandwidths s and regularization coefficients λ for the proposed method and standard kernel CCA (KCCA).

	s/λ	0.50	0.75	0.90	1.05	1.25	1.50	KCCA
	90	0.015879	0.003996	0.008281	0.004276	0.002858	0.010321	0
	100	0.002744	0.006109	0.008728	0.008750	0.008935	**0.002094**	0
	110	0.007011	0.003913	0.003087	0.009613	**0.001950**	0.004661	0
E_1	120	0.002909	0.004651	0.004728	0.000857	0.005236	0.002331	0
	130	**0.002658**	**0.001999**	0.003187	**0.000495**	0.003572	0.005274	0
	140	0.002853	0.002455	**0.001785**	0.001812	0.003594	0.003747	0
	150	0.005741	0.002104	0.002377	0.003061	0.002092	0.002119	0
	1	0.004364	0.002335	0.003014	0.003124	0.004833	0.008518	0
	10	0.003215	0.002947	0.002061	0.001261	0.003802	**0.001956**	0
	20	0.001894	**0.001439**	0.002148	0.001934	**0.01192**	0.003705	0
E_2	30	0.003969	0.002372	**0.001939**	**0.000449**	0.001505	0.004887	0
	40	**0.001111**	0.004647	0.003106	0.001947	0.001731	0.004896	0
	50	0.003036	0.002635	0.003496	0.004956	0.003544	0.001546	0
	1	**0.015007**	0.066003	0.028483	0.019171	0.027136	0.014781	0.000532
	25	0.201227	**0.019617**	**0.017199**	**0.003252**	0.009107	**0.003634**	0
	40	0.106513	0.023754	0.019312	0.008741	**0.004017**	0.004525	0
E_3	50	0.253164	0.034919	0.0205772	0.021162	0.005307	0.005333	0
	60	0.467826	0.073340	0.018540	0.006271	0.003427	0.004216	0
	75	0.156971	0.027303	0.020168	0.018833	0.045894	0.004159	0

(0.9 and 1.05) are visualized in Figures 5.4. We observe that for the standard kernel CCA, the cross-validation error is decreasing as the increase of the inverse bandwidth s. From this observation, the CV does not work for choosing an appropriate bandwidth of the kernel CCA. On the other hand, the CV error attains the minimum value at a point so that we can select an appropriate bandwidth parameter for the proposed method.

5.5.2 Real world datasets

We apply the proposed method to real-world data sets. In the first part of this section, we investigate the relationship between two sets of variables, comparing the proposed hrKCCA and standard kernel CCA. In the second part of this section, we find low dimensional feature spaces for classification tasks by the proposed method. The results of the proposed method are compared with other classification approaches such as the linear discriminate analysis (LDA), quadratic discriminate analysis (QDA), support vector machine (SVM), decision tree, Bayesian networks and some exiting human action recognition methods.

Table 5.4: Cross-validation errors for *nutrimouse* dataset.

s/λ	0.5	0.75	0.90	1.05	1.25	KCCA
1	1.368060	1.332372	1.224655	1.140264	1.140264	332193.7
10	0.001563	0.001431	0.002952	0.001339	0.003788	0.056211
20	**0.000970**	0.002959	0.003232	0.002591	0.003366	0.000002
30	0.002536	0.003248	**0.001028**	0.003605	0.001666	0
40	0.002468	0.001077	0.001874	0.002151	**0.001479**	0
50	0.001307	0.002300	0.001511	0.002653	0.001892	0
60	0.001120	**0.000664**	0.001876	**0.001058**	0.002285	0
70	0.001024	0.001314	0.001569	0.001084	0.002362	0

Dependent features and measure of relationship

Nutrimouse data, D_1. *Nutrimouse* dataset is given by a nutrition study of the forty mice. It was published by Martin et al. (2007) that has been also used in the 'CCA' package of R program to measure the relationship of two sets of variables: Liver cells and Hepatic fatty acids. We have already shown the results of the standard kernel CCA with this dataset to illustrate its limitation in Section 2.

Note that *nutrimouse* data has more dimension than sample size. It is well known that if the sample size is smaller than the dimension, we are not able to use linear CCA, in general.

We calculate the 10-fold cross-validation errors for kernel CCA and the proposed method using regularization coefficient $\lambda \in \{0.5, 0.75, 0.90, 1.05, 1.25\}$, and the inverse bandwidth $s \in \{1, 10, 20, 30, 40, 50, 60, 70\}$. The results are tabulated in the Table 5.4. We can see that for the proposed method it is possible to select an appropriate bandwidth and regularization coefficient ($s = 60$, $\lambda = 0.75$) corresponding to the smallest CV error, while the kernel CCA fails to find a good parameter (error goes to 0 as $s \to \infty$). The scatter plots using the eight inverse bandwidths with the best regularization coefficient $\lambda = 0.75$ are also shown in the Figures 5.5. We can say by comparing this figure with the Figures 5.1 that the proposed method has a well-posed solution with highly dependent features (the smallest CV error corresponding to the Figure 5.5(g)).

***DBWorld* datasets for measure of association,** D_2. This dataset consists of the subjects and bodies of emails, which are represented by *bag-of-words* features. The sample size is 64, and there are 242 dimensional features for subjects, and 4702 features for bodies. The dataset is available at the UCI machine learning repository (Bache and Lichman, 2013).

Psychological dataset, D_3. This is one of the most well known datasets to measure the relationship of psychological variables and academic variables; the former consists of the locus of control, self-concept, and motivation, while the latter of reading, writing, math, science, and additional gender variable (http://www.ats.ucla.edu/stat/sas/dae/canonical.htm). The sample size is 600. With the linear CCA, the relationship is 0.46, which implies weak linear dependence.

Carbig dataset, D_4. Carbig dataset contains various measured variables for automobiles. It has been used in the MATLAB Statistics Toolbox with 392 data points (without missing values). To use 10-fold cross-validation, we take 390 data points without first and last observations.

To measure the association for the above datasets ($D_2 - D_4$), we apply the proposed method with cross-validation. For the cross-validation, we use the inverse bandwidths $s \in \{50, 60, 70, 80, 90\}$, $\{30, 40, 50, 60, 70\}$, and $\{50, 60, 70, 80, 90\}$ for DBworld, Psychological and Carbig, respectively; the regularization coefficients are set $\lambda \in \{0.50, 0.75, 0.90, 1.05, 1.25\}$, $\{0.075, 0.090, 0.1\}$, and $\{0.01, 0.025, 0.050, 0.075, 0.09, 0.1\}$ for the respective datasets. The selected parameters are $(s, \lambda) = (20, 1.05)$, $(60, 0.09)$, and $(60, 0.09)$. The first canonical correlation of the proposed method are 0.989, 0.985 and 0.998 for datasets D_2, D_3 and D_4, respectively. The scatter plots of first canonical variates are visualized in Figs. 5.6 ((a) standard kernel CCA and (b) the proposed method). From this visualization, on the one hand, we can see that the standard kernel CCA with CV has provided high dependence features with ill-posed solution, but on the other hand, the proposed method is able to extract dependent features with well-posed distributions, for all the datasets.

Low dimensional space for classification

In this subsection, we use seven real world datasets for classification from the UCI repository (Bache and Lichman, 2013): *wine*, *BUPA liver disorders*, *diabetes*, *DBWorld for subjects*, *DBWorld for bodies*, *KTH* and *UMD* dataset to estimate low dimensional canonical features of the input space using the proposed method. We then use the features for the classification task. For the ℓ-class classification problem, the ℓ dimensional binary vectors $(1, 0, \ldots 0), (0, 1, \ldots 0), \ldots (0, 0, \ldots 1)$ are used for $\mathbf{Y}$ to specify the classes. We evaluate the classification errors by the kNN classifier (k=5) (*hrKCCA + kNN*) and linear SVM

($hrKCCA + SVM_L$) with the nonlinear features of the data. We use only one or two canonical features for the classification. The sample size and the dimensionality of the datasets are summarized in Table 5.2.

Wine dataset, D_5. The explanatory variable $\mathbf{X}$ is 13 dimensional continuous chemical measurements, and the response variable $\mathbf{Y}$ consists of three dimensional binary vectors corresponding to the three types of wine. The sample size is 178. To apply the 10-fold cross-validation, we have drawn a random sample of size 170 out of 178. For the cross-validation in the proposed method, we used six values of the inverse bandwidths $s \in \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06\}$ and five regularization coefficients $\lambda \in \{0.5, 0.75, 0.90, 1.05, 1.25\}$. The selected parameters are ($s = 0.05, \lambda = 1.05$) applied to the whole dataset. The first canonical correlation of the proposed method is 0.94. The two dimensional plots of first canonical variates and the first two canonical variates of $\mathbf{X}$ are shown in Figures 5.7 (a(i) and a(ii)), from which we can observe that there is strong dependence between the first canonical variates (a(i)), and the first two features for $\mathbf{X}$ are able to extract a clear cluster structure of the dataset (a(ii)).

For comparison, we have also applied the standard kernel CCA with heuristic three choices of the bandwidth, $s_j = \frac{1}{2\sigma_j^2}$, ($j = 1, 2, 3$): $\sigma_1, \sigma_2, \sigma_3$ are the median(Gretton et al., 2008), the minimum(Hardoon and Shawe-Taylor, 2009) and $\sqrt{10}$(Huang et al., 2009a) of the pairwise distances of the standardized $\mathbf{X}$. The values are $s_1 = 0.02$, $s_2 = 0.37$ and $s_3 = 0.05$. With these bandwidths, the first canonical variates and the first two canonical variates of $\mathbf{X}$ given by the standard kernel CCA are shown in Figs. 5.7 (b-d), in which the heuristic choice of bandwidth can extract data structure, but the shape of the clusters is less clear than the result of the proposed method. Also, the correlation of the first canonical variates are not so high (0.80, 0.39, 0.85).

We evaluate the classification errors by the kNN classifier (k=5) with the canonical variates for the data. We split 178 data into 118 for training and 60 for testing (Bache and Lichman, 2013). The classification error of the proposed method and the standard kernel CCA with three bandwidths (s_1, s_2, s_3) using only the first canonical variates of $\mathbf{X}$ are 13% and 36.66%, 0%, and 20%, respectively. With the first two canonical variates, the classification errors are 0 and 0, 0, and 1.66, respectively for the proposed method and kernel CCA. The results indicate that the canonical variates found by the proposed method

Table 5.5: Classification errors (%) for *wine*, *BUPA liver disorders* and *diabetes*. One or two dimensional features are used with the proposed method (hrKCCA+kNN and hrKCCA+SVM) and the kernel CCA.

		Wine		*BUPA*		*Diabeties*	
	ELD	1	2	1	2	1	2
hrKCCA	$+kNN$	14.04	**0**	0.58	**0**	**0**	**0**
	$+SVM_L$	14.04	**0**	0.58	**0**	**0**	**0**
KCCA+kNN	s_1	34.83	2.81	27.54	24.64	37.24	33.79
	s_2	0	2.81	**0**	**0**	2.07	2.07
	s_3	29.77	2.81	45.79	42.89	20	20
KCCA+SVM$_L$	s_1	29.21	2.24	53.33	41.45	33.79	17.93
	s_2	2921	2.24	**0**	**0**	2.07	**0**
	s_3	28.65	2.81	42.03	53.04	20.00	15.86
Full dimensions	*LDA*	1.10		30.10		11.00	
	QDA	0.60		40.60		9.70	
	SVM_G	1.69		25.22		2.14	

have stronger ability for classification.

BUPA liver disorders dataset, D_6 and **diabetes dataset,** D_7. For the cross-validation, the inverse bandwidth s and the regularization coefficient λ are selected from {0.02, 0.03, 0.04, 0.05, 0.06} and {0.09, 0.10, 0.25}, respectively, for D_6; {0.009, 0.01, 0.02, 0.03, 0.04, 0.05} and {0.75, 0.90, 1.05} for D_7. The first canonical correlation are 0.94 for D_6 and 0.98 for D_7.

Using the low dimensional canonical features (only 1 and 2) obtained by the proposed method, we evaluate the leave-one-out cross-validation of the misclassification rates for kNN and linear SVM classifiers ($hrKCCA + kNN$ and $hrKCCA + SVM_L$). In comparison, we use the canonical features given by the standard kernel CCA with the same three heuristic choices of the inverse bandwidth as the ones used for *wine* data. The CV errors for D_5, D_6 and D_7 are tabulated in Table 5.5. We also show the leave-one-out misclassification rates with the full dimensions by linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and 10-fold cross-validation errors for the nonlinear SVM (Gaussian kernel, SVM_G), which are taken from (Izenman, 2008). We see from this table that the proposed method is able to give the best results in almost all cases. Note also that the results of standard kernel CCA strongly depend on the choice of bandwidth parameter, which contrasts with the proposed method incorporating the cross-validation.

DBWorld datasets for classification (subjects and bodies)

We have already used the DBWorld email dataset for measuring the relation between subjects and bodies, but use it again for a different purpose. The dataset is used for classification: the task is to classify an email between "announcement of conferences" and "anything else" based on the subjects and bodies. We apply the proposed method as a preprocessing technique for this purpose.

For the cross-validation, six inverse bandwidths $s \in \{0.03, 0.04, 0.05, 0.06, 0.07, 0.08\}$ and five regularization coefficients $\lambda \in \{0.5, 0.75, 0.9, 1.05, 1.25\}$ are used. The chosen values are $s = 0.07$ and 0.07 for the subject and body dataset, respectively, and regularization coefficients $\lambda = 1.25$ and 0.75. For these datasets the first canonical correlations are 0.84 and 0.93, respectively. To evaluate the misclassification rates of the classification based on the canonical features, we split 64 (35, 29) data into 48 (26, 22) and 16 (9, 7) randomly as by Filannino (Filannino, 2011). The average misclassification rates given by the proposed method using the first and first two canonical features are shown in Table 5.6. The results of SVM (linear kernel), SVM-RBF (Gaussian kernel), decision tree (C4.5), and Bayesian network (K2), using the full dimension, are taken from Filannino (Filannino, 2011). The canonical features found by the proposed method show better classification ability than all the other methods. This means the proposed hrKCCA extracts features for classification effectively with an appropriate choice of parameters.

The scatter plots and index plots (number of data points $1-64$ in x-axis and first canonical variate of $\mathbf{X}$ in y-axis) of the first canonical variates and the first canonical variate of the explanatory variable are shown in Figures 5.8 (Subjects, Bodies, BUPA and Diabetes). As can be seen in the figures, we can extract the data structure properly with only one canonical variate.

KTH human actions dataset, D_8. The human actions video database of *KTH* dataset (Schüldt et al., 2004) is used to show the superiority of the proposed method over kernel CCA as well as other classification methods. This dataset has six types of human actions: boxing, hand clapping, hand waving, jogging, running, and walking, performed by 25 subjects (training set 1-8, validation set 9-16 and test set 17-25) with four different scenarios: outdoors, outdoors with scale variations, outdoors with different clothes, and indoors. The resized video sequence for the experiment is $120 \times 160 \times 20$, i.e. the dimension of $\mathbf{X}$ is

Table 5.6: Classification errors (%) using one dimensional estimated subspace of *DBWorld subject and bodies* datasets by the proposed method (hrKCCA+kNN and hrKCCA+SVM_L) other exiting methods.

		Subjects		*Bodies*	
	ELD	1	2	1	2
hrKCCA	$+kNN$	**1.25**	**1.25**	1.25	1.25
	$+SVM_L$	2.5	1.875	1.875	**0.625**
Full dimensions	SVM:linear k.	2.3437		2.3437	
	SVM-RBF: Gaussian k.	2.3437		4.6875	
	Decision tree: C4.5	7.8125		3.1250	
	Bayesian Network: K2	1.5625		4.6875	

384000.

First, we extract a two dimensional subspace using both the standard kernel CCA and the proposed method to recognize the six human actions only in the outdoor scenario. For this purpose, we take six heuristic inverse bandwidths for the kernel CCA: mean, median, minimum, maximum, 0.05 and 3 $\times$ median based on the pairwise distance of standardized $\mathbf{X}$. The scatter plots of the first canonical variates (upper row) and the first two canonical variates (lower row) of $\mathbf{X}$ are shown in Figures 5.9. From the figures, we can conclude that the heuristic choice of bandwidths are not able to extract high dependence features or effective low dimensional subspaces for this recognition task.

For the proposed method, we select the parameters by 10-fold CV. We consider six inverse bandwidths $s \in \{0.01, 0.05, 0.10.0.15, 0.20, 0.25\}$ and three regularization coefficients $\lambda \in \{0.90, 1.05, 1.25\}$. The appropriate inverse bandwidth and regularization coefficient are $s = 0.20$ and $\lambda = 1.05$, respectively. We visualize the scatter plots of the first canonical variate (upper row) and first two canonical variates of $\mathbf{X}$ (lower row) using all six inverse bandwidths with fixed regularized coefficient $\lambda = 1.05$ in Figures 5.10. This visualization ensures that, the proposed method is able to extract high dependence features as well as an effective, low dimensional subspace for recognition of human actions. Using this subspace, both of the classification methods: kNN classifier (k=5) and linear SVM can recognize all six human actions perfectly (leave-one-out recognition rate is 100%).

Finally, we show the performance of the proposed method in comparison with some exiting human action recognition methods (Danafar et al., 2010). By the proposed method, we extract two dimensional subspace using training and test set for all scenarios i.e., $\mathbf{X} \in$

Table 5.7: Recognition rate (%) for *KTH* dataset (all scenarios) by the proposed method (hrKCCA+kNN and hrKCCA+SVM_L) and other methods.

Methods		Recognition rate (%)
hrKCCA+kNN	s_1	99.1
	s_2	99.5
	s_3	96.8
hrKCCA+SVM_L	s_1	99.8
	s_2	99.5
	s_3	96.3
Full dimensions (Danafar et al., 2010)	Lin *et al.*	93.4
	Danafar *et al.*	93.1
	Schindler and Van Gool	92.7
	Schüldt *et al.*	71.7

$\mathbb{R}^{408\times384000}$. We use the proposed method for a fixed regularized coefficient, $\lambda = 1.05$ and three inverse bandwidths, $s \in \{0.01, 0.10, 0.20\}$ (10-fold CV errors using validation set and only outdoor scenario video are small). The scatter plots of the first canonical variates (upper row), first two canonical variates of $\mathbf{X}$ (middle row) and confusion matrices (lower row) are shown in Figure. 5.11. In view of the visualization, we can observe that there is a strong dependence between the first canonical variates and the first two features of $\mathbf{X}$ are able to extract a clear cluster structure of the human actions.

We also evaluate recognition rates for the test set using the estimated subspace by the kNN classifier (k=5) and SVM. The results of the proposed method along with other methods are tabulated in Table 5.7. It is remarkable to see that the canonical variates found by the proposed method have stronger ability for recognition than all the other methods. The results for the full dimensions are taken from Danafar *et al.* (Danafar et al., 2010).

The *UMD* sushi making data, D_9[1]: In recent, Teo *et al.* have been used this dataset as supervised and unsupervised settings with adding language but accuracy rate, 91.67 stile need to improve (Teo et al., 2012). In the dataset four actors are performed to make sushi, consist of 12 actions: cleaning (A), cutting (B), drinking (C), flipping (D), peeling (E), picking-up (F), pouring (G), pressing (H), sprinkling (I), stirring (J) tossing (K), turning (L), based on different kitchen tool tools. The 48 video sequences are around 30 seconds. The resized video sequence for our experiment is $480 \times 640 \times 20$, i.e. $\mathbf{X} \in \mathbb{R}^{48\times6144000}$.

[1]http://www.umiacs.umd.edu/research/POETICON/umd_sushi/

Table 5.8: Recognition rate (%) for *UMD* dataset by the proposed method (hrKCCA+kNN and hrKCCA+SVM_L) and some of the best stat-of-the-art methods of this dataset.

Methods		Recognition rate (%)
hrKCCA	kNN	100
$s \in \{0.1, 0.5, 1, 10, 50\}$	$+SVM_L$	100
STIP+ Bag of Words	SVM_P	77.08
Action Features+Language	SVM_P	91.67
	Semi-supervised EM	9167

We extract a low dimensional subspace with first two canonical variates of $\mathbf{X}$ using eight inverse bandwidths $s \in \{0.01.0.05, 0.1, 0.5, 1, 5, 10, 50\}$. We then split first three actors for training and fourth actor for test. Finally, we evaluate recognition rates for the test set using the estimated subspace by the hrKCCA+kNN classifier (k=5) and hrKCCA+SVM. The low dimensional subspace can successfully recognize all 12 actions. The results of the proposed method and some of the best stat-of-the-art methods are tabulated in Table 5.8. It is remarkable to see that the canonical variates found by the proposed method have stronger ability for recognition than all the other methods. The rest of results (STIP+ Bag of Words and Action Features + Language) are taken from Teo *et al.*(2012).

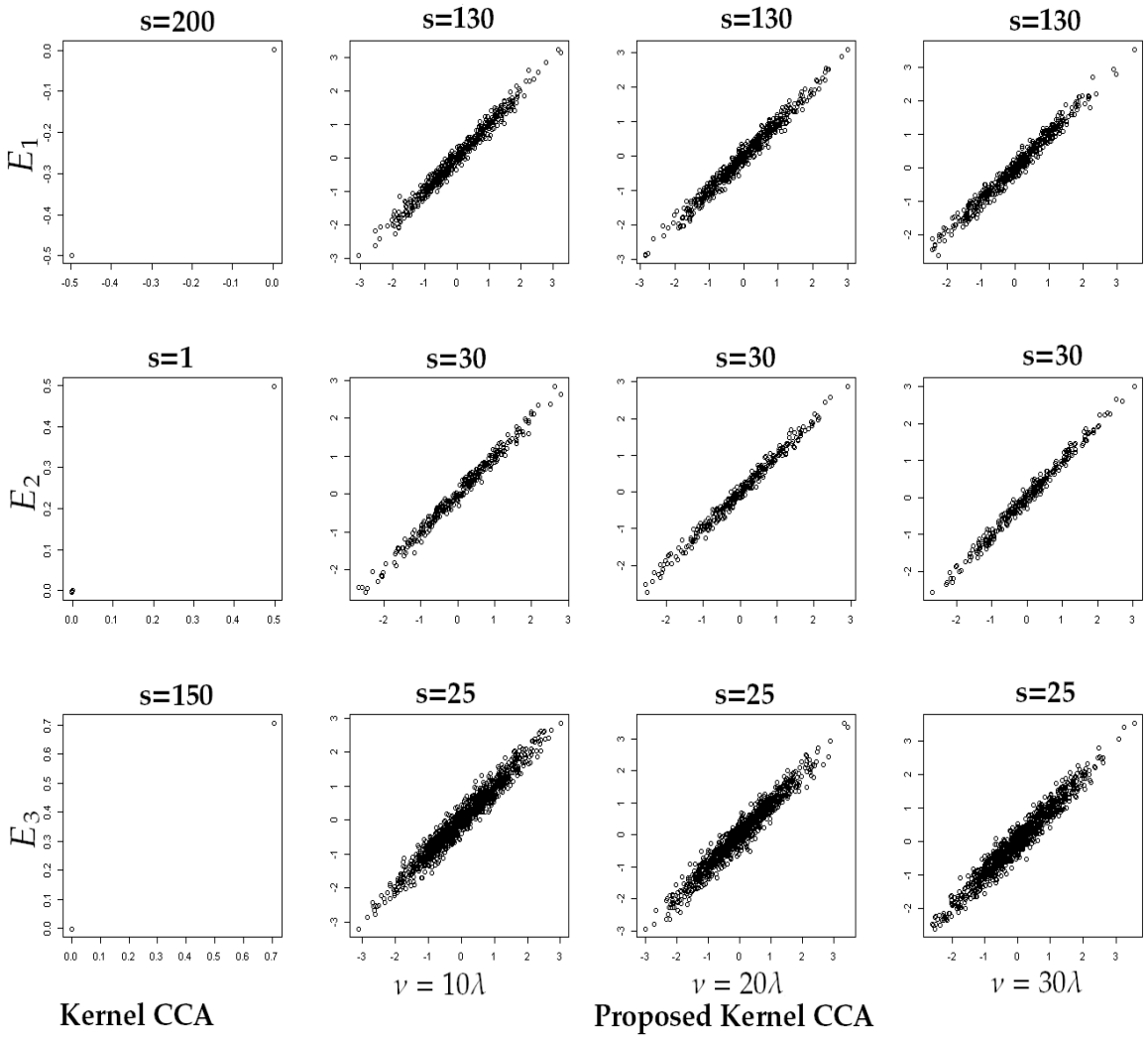

Figure 5.3: Scatter plots of 1st kernel canonical variates for the examples ($E_1 - E_3$). The first column for the standard kernel CCA. The final three columns for the hrKCCA, using different trade-off c for the regularization parameters: $\nu = c\lambda$. The inverse bandwidth s and the 4th moment regularization coefficient λ are chosen by the CV.

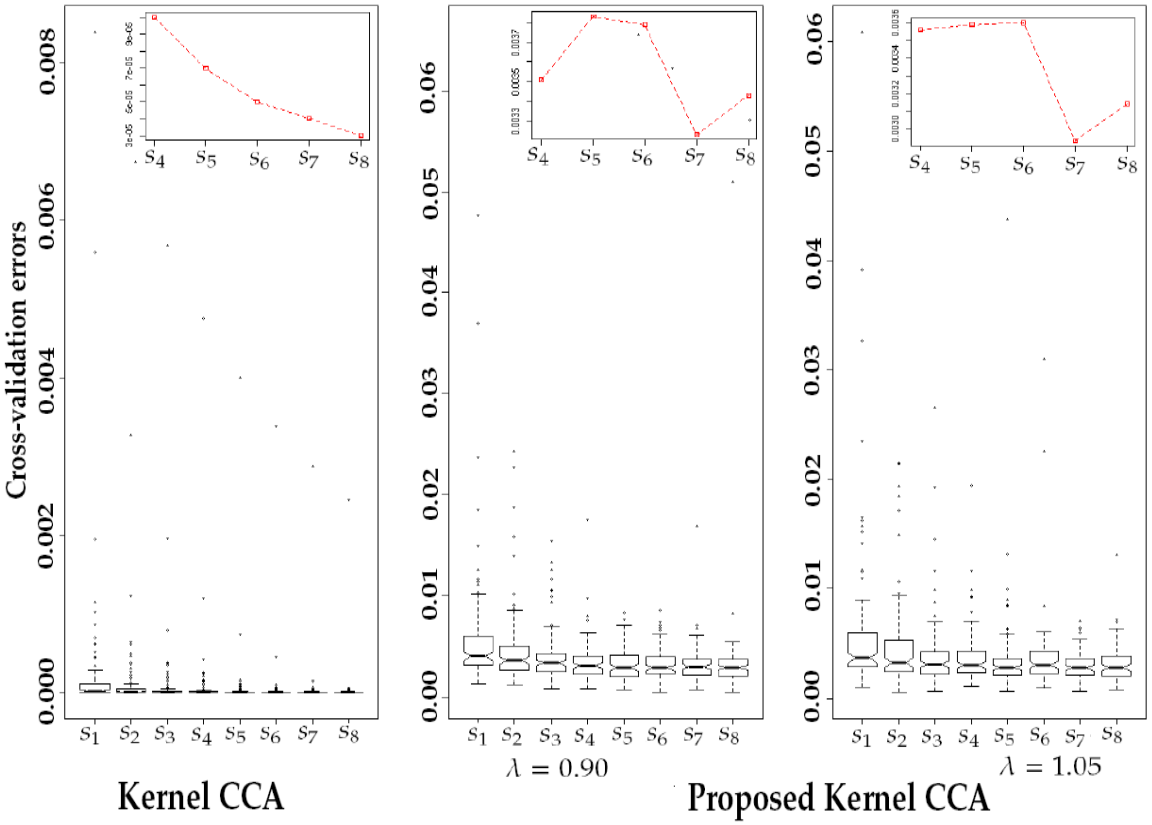

Figure 5.4: Box plots and line plots (inset) using mean values of cross-validation errors of 100 samples for example 3 (bandwidths, $s_1 = 225, s_2 = 250, s_3 = 275, s_4 = 300, s_5 = 325, s_6 = 350, s_7 = 375, s_8 = 400$).

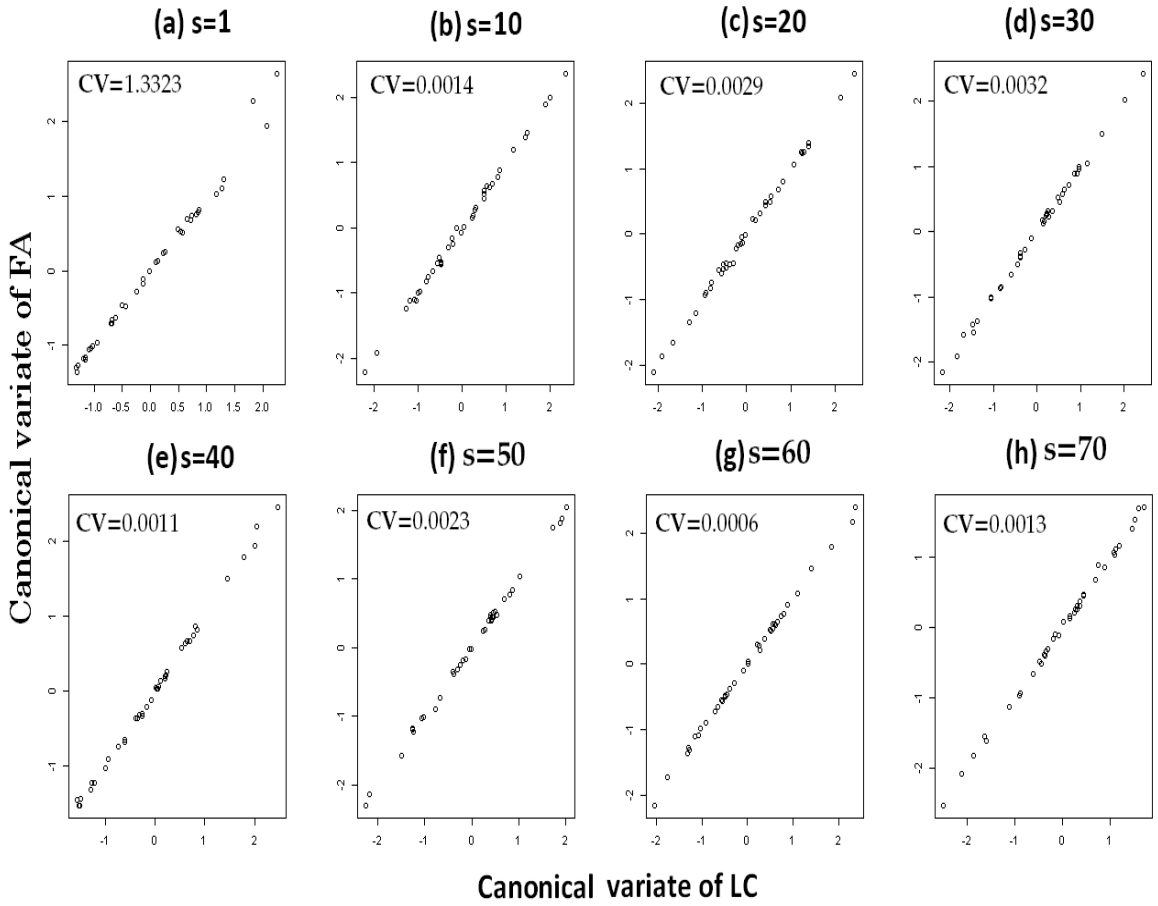

Figure 5.5: Scatter plots of the 1st kernel canonical variates given by the proposed method for the nutrimouse dataset (liver cells and hepatic fatty acids) using the Gaussian RBF kernel with eight inverse bandwidths s and fixed regularization coefficient $\lambda = 0.75$. The 10-fold cross-validation error is also embedded (see also Table 5.4).

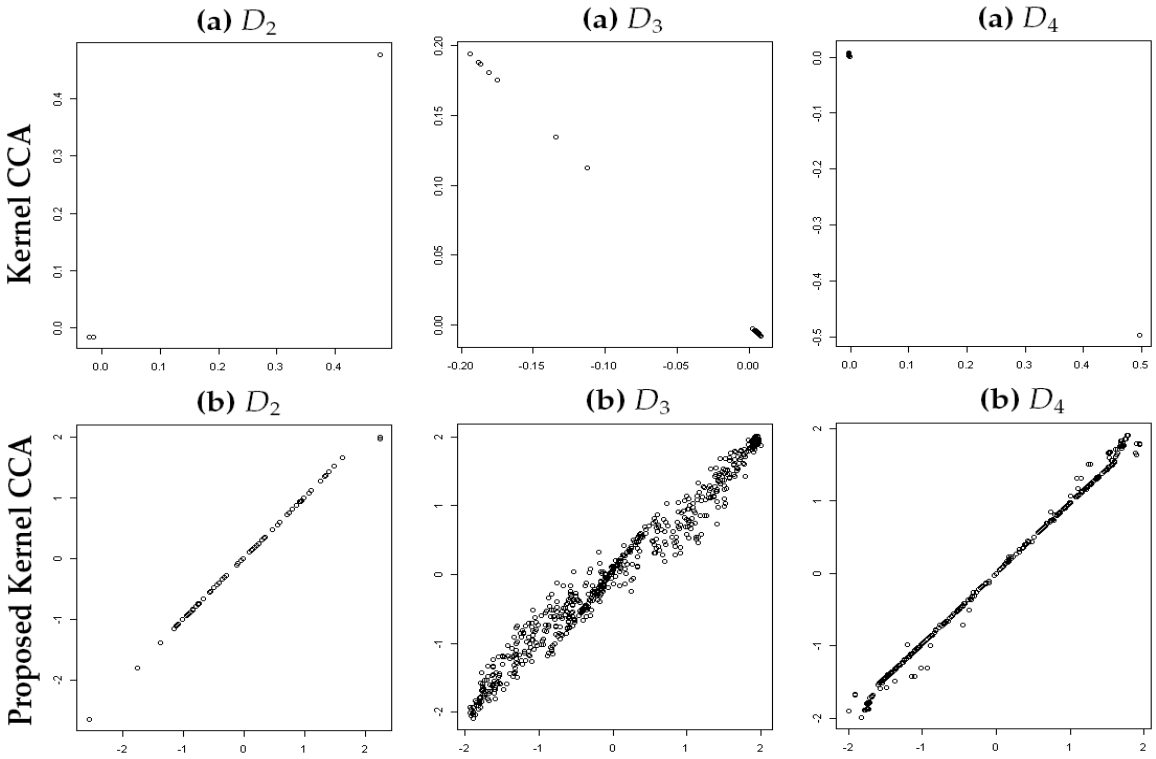

Figure 5.6: Scatter plots of the first canonical variates of real datasets (Email (D_2), Psychological (D_3) and Carbig (D_4)) using the parameters chosen by CV for the kernel CCA (a) and the proposed method (b).

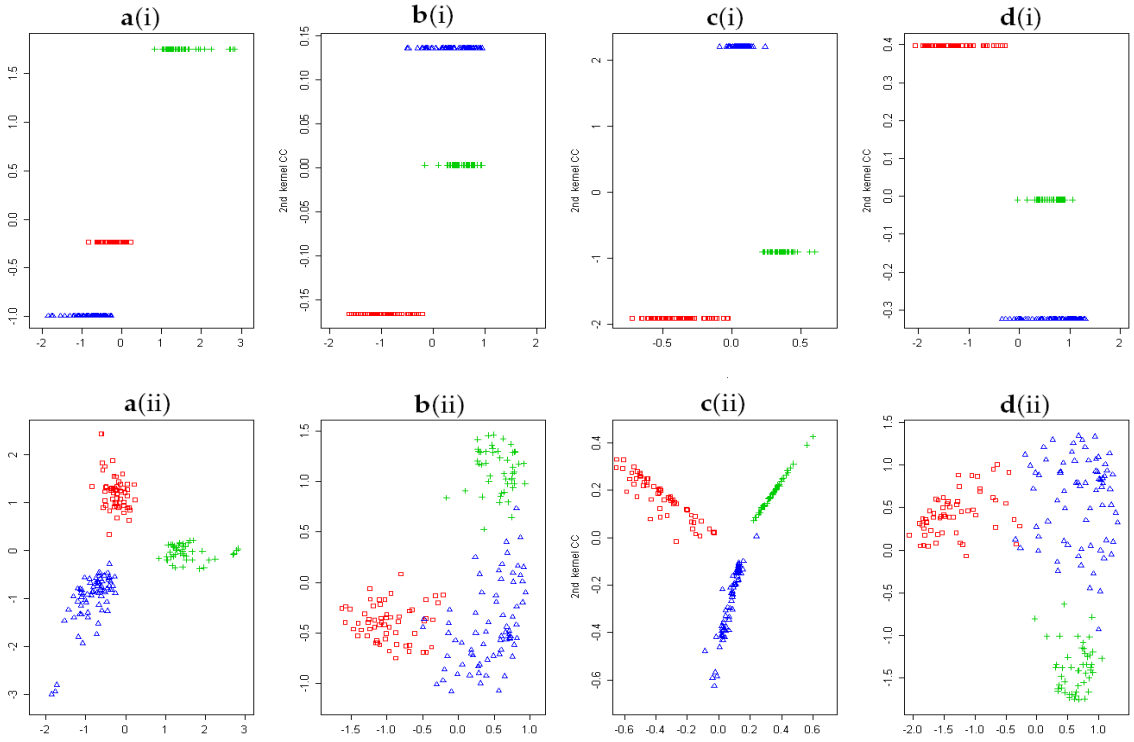

Figure 5.7: Scatter plots of the first canonical variates ($a(i)$) -($d(i)$) and the first two canonical variates of the exploratory variables ($a(ii)$ -$d(ii)$) for the *wine* dataset. The proposed method ($s = 0.05, \lambda = 0.1$) in (a) and kennel CCA using three heuristic bandwidths ($s_1 = 0.02, s_2 = 0.073, s_3 = 0.05$) in (b - d) are shown.

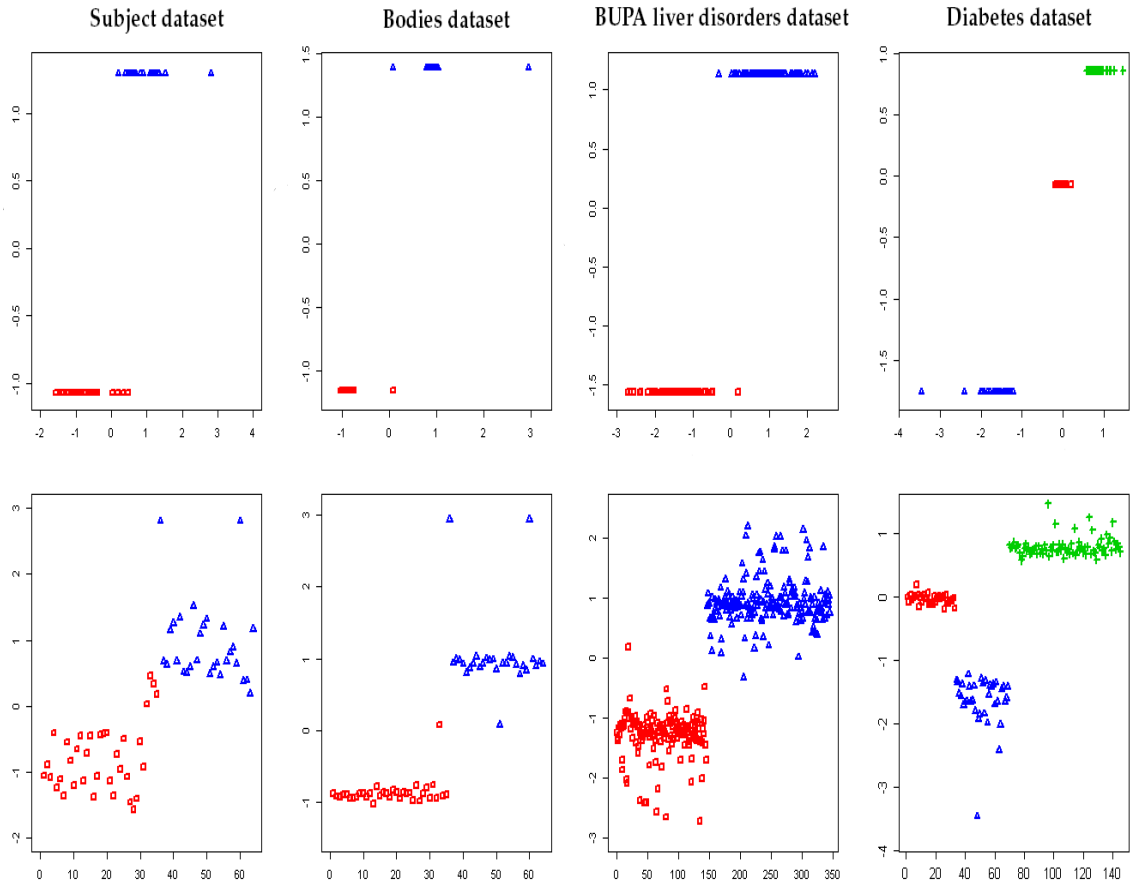

Figure 5.8: Scatter plots of the first canonical variates (upper row) and one dimensional index plots (lower row) given by the proposed method for *DBWorld subject, bodies, BUPA liver disorders*, and *diabetes.*

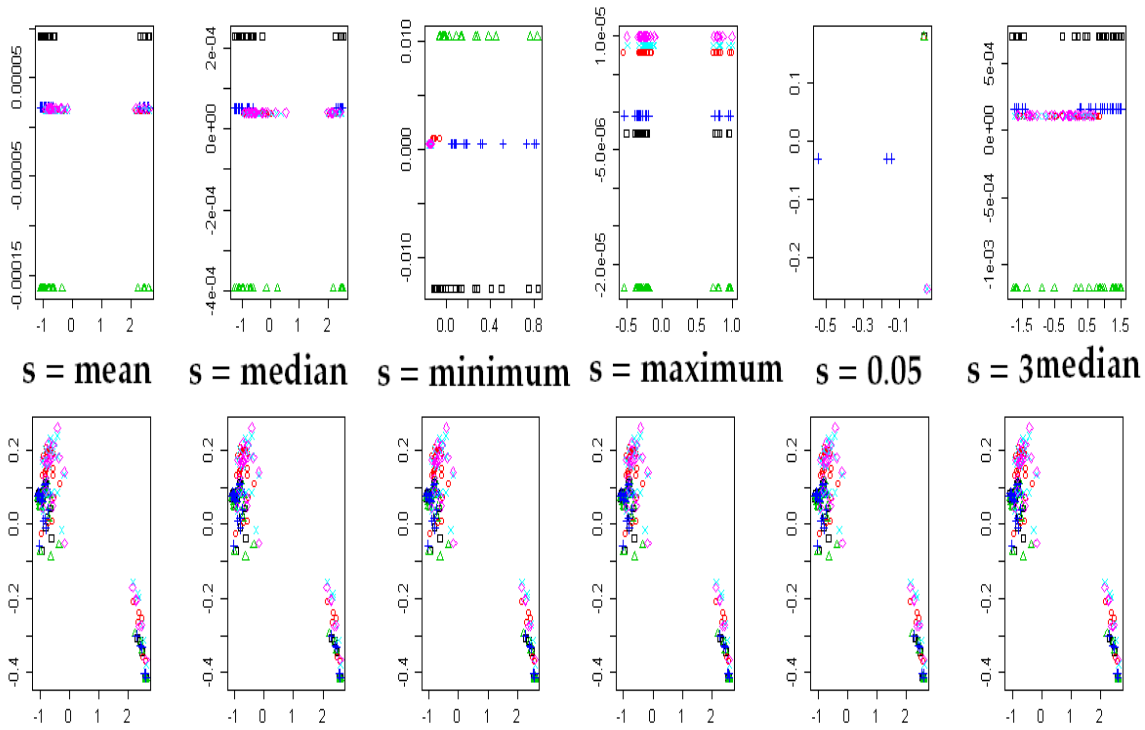

Figure 5.9: Scatter plots of the first canonical variates (upper row) and the first two canonical variates of **X** (lower row) using *KTH* dataset (outdoor scenario only) for the kernel CCA.

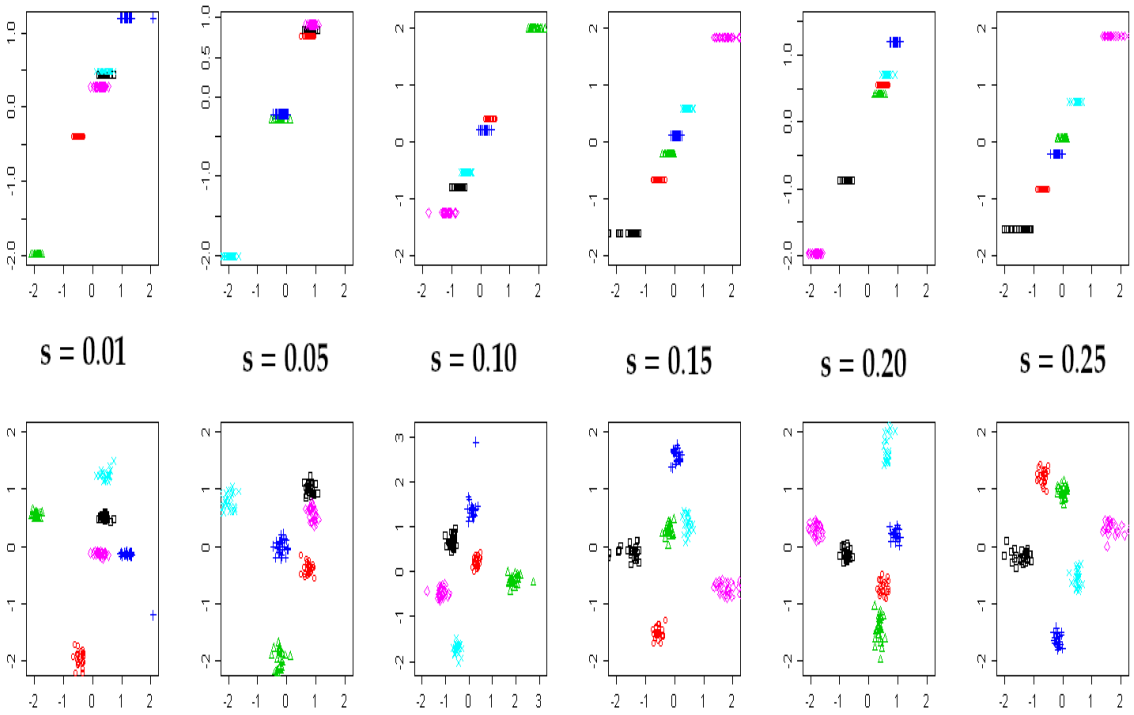

Figure 5.10: Scatter plots of the first canonical variates (upper row) and first two canonical variates of **X** (lower row) using *KTH* dataset (outdoor scenario only) for the proposed hrKCCA.

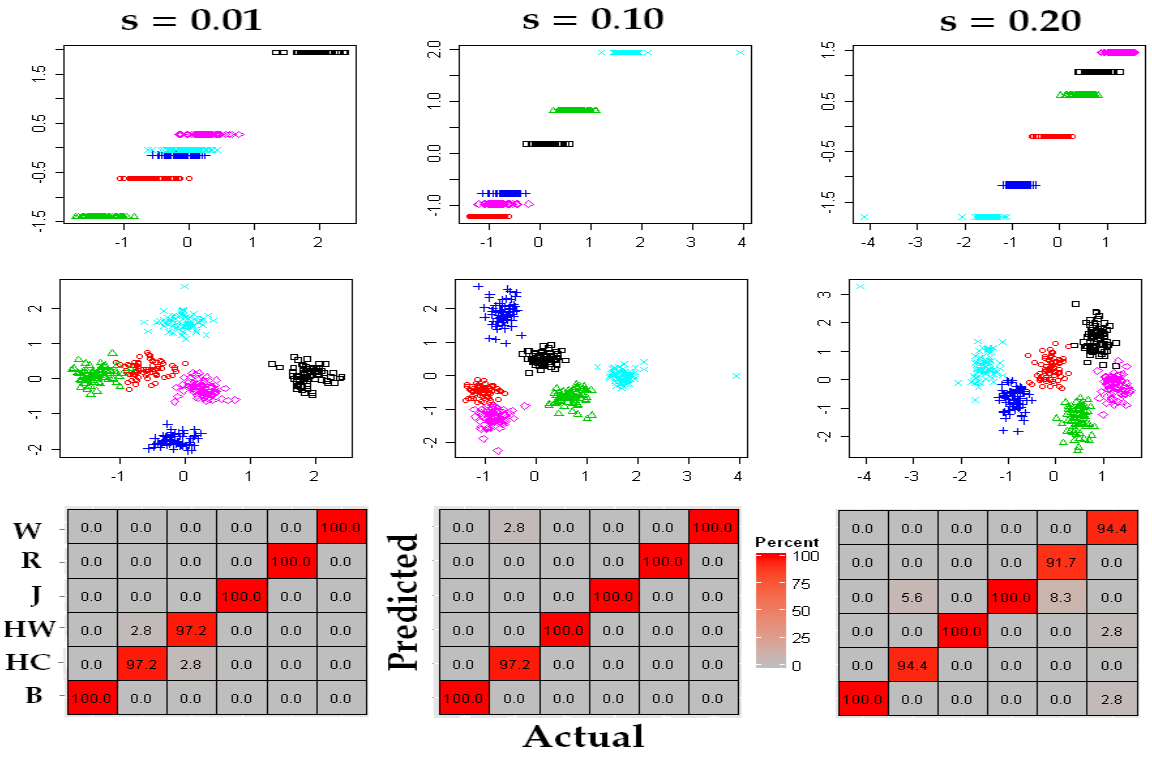

Figure 5.11: Scatter plots of the first canonical variates (upper row), first two canonical variates of **X** (middle row) and confusion matrices (lower row) using *KTH* dataset for all scenarios (boxing (B), hand clapping (HC), hand waving (HW), jogging (J), running (R), and walking (W)) for the for the proposed hrKCCA.

Chapter 6

Conclusion and Future Research

6.1 Conclusion

First, we discussed the drawbacks of kernel principal component analysis (kernel PCA), and proposed a method for choosing hyperparameters, optimal kernel (parameters in a kernel) and the number of kernel principal components, through the LOOCV for the reconstruction errors of pre-images. We made empirical studies using synthesized examples and real-world datasets. For evaluation of the proposed method, in addition to visualization, we used classification errors for the projected data onto the subspace chosen by the method, if the data set is provided for a classification task. We observed that for all the datasets classification performances of the kernel PCA chosen by the proposed method is the best or close to the best among the candidates of hyperparameters. The experimental results imply that the proposed method successfully provides an automatic way of finding such hyperparameters that give appropriate low-dimensional representation of dataset.

We applied the proposed method for synthesized and real datasets in the Section 3.3. The scatter plots of first two kernel principal components for synthesized datasets (with the best hyperparameter) are visualized in the Figure 3.5 (c). Both the plots show that the proposed method is able to extract the hyperparameters that can separate three cluster clearly without using the explicit clusters information. We next applied the proposed method to real datasets. For classification data sets, we can see from the Tables 3.3 (for five real-world datasets) and 3.4 (USPSG dataset) that the hyperparameter gives the best or close to best LOOCV classification error. In all cases, we observe that the chosen hyperparameters are

close to the best parameters for the classification error. For unlabeled dataset, from the Table 3.5 and the Figure 3.6 we can see that the hyperparameter chosen by the proposed method provides the features with a clearer structure than the other two hyperparameters. For this dataset, Izenman (2008) provides detailed analysis on the results of kernel PCA with a hand-tuned bandwidth parameter: a meaningful "curve" structured is observed in the result of two-dimensional kernel PCA. As shown in Figure 3.6, the proposed method automatically chooses such a hyperparameter that accords with the observation in Izenman (2008). These experimental results suggest the effectiveness of the proposed method.

Second, we compared the performances of five (classical, robust and the standard kernel CCA with three functions) estimators of canonical correlation coefficient that are commonly used in the statistical literature. Their performance was investigated through qualitative robustness indices, sensitivity curves and breakdown point in linear, contaminated and nonlinear simulated datasets. It is found that both classical and robust measure fail completely to capture nonlinear relationships. All kernel measures, especially the Gaussian kernel and Laplacian kernel are able to detect nonlinear relationships. The robust measure is found to be the best and followed closely with kernel CCA for contaminated datasets. On the other hand, the classical CCA gives the best performance for multivariate normal data set, but it fails in contaminated and nonlinear datasets. By breakdown plots, we observe that the breakdown is very high for robust estimator in linear data, but in nonlinear data kernel methods are better.

Finally, we proposed a kernel CCA method based on the regularization for the 4th order moments. The proposed method is to overcome the limitations of the standard kernel CCA: choosing the bandwidth and regularization coefficients are not straightforward, and the cross-validation approach give undesired distributions of the canonical variates. By comparing the results of kernel CCA and the proposed method of Figure 5.1, 5.5, 5.5 and 5.6 it is clear that the proposed method provides well-posed solution but standard kernel CCA. From the Table 5.3, 5.4 and the Figure 5.5 it is confirmed that we can optimize all the parameters using cross-validation, which provides more well-shaped distribution of the canonical features of the proposed method. When we apply the proposed method of the classification datasets, the low dimensional canonical variates provide favorable features of the classification. From the visualization of the first two canonical variates in Figure 5.7

and 5.8 we can see clearer data structure of the proposed method. From the Figure 5.9, 5.10, 5.11 and Table 5.7 we see that the proposed method provides better performance over standard kernel CCA for the both human action datasets.

The experimental results confirmed that the propose approach has, in fact, these favorable properties unlike the standard kernel CCA. In the real world datasets, the classification performance with the data projected to the low dimensional features outperforms the results of the state-of-the-art methods of the same task.

6.2 Future research

There are also some possibilities to improve both the proposed methods: kernel PCA and higher order regularized kernel CCA (hrKCCA). For the kernel PCA, first the optimization such as fixed-point and steepest descent method for computing the pre-image has possibility of being trapped by local optimum. Applying other pre-image methods to alleviate the problem will be an important future research. Second, since our method uses the cross-validation with pre-image optimization, it may be time-consuming for large datasets. One possible approach is to use a part of data onto evaluating reconstruction errors, and it is also an interesting future direction to develop a more efficient way of hyperparameter choices of kernel PCA. Third, the reconstruction errors in the proposed method assume that the original space admits a metric, while kernel PCA can be applied to more general data spaces including non-metric spaces. It is also among our future studies to consider hyperparameter choices applicable to kernel PCA for non-metric spaces.

For hrKCCA the gradient based method may converse with local optima. It is important to develop a more sophisticated optimization technique as a future work. Statistical properties such as asymptotic and robustness will be also among important future directions of research for this proposed method.

Developing robust kernel PCA and kernel CCA based on robust covariance and cross-covariance operators are also an interesting direction of future research. A robust kernel PCA has been derived theoretically based on projection residual error (Huang et al., 2009b). They have derived influence function of kernel PCA but not for standard kernel CCA. In the future we will propose the influence function of standard kernel CCA. We are also trying

to develop robust covariance and cross-covariance operators to apply in kernel PCA and kernel CCA, respectively.

Bibliography

S. Akaho. A kernel method for canonical correlation analysis. *International meeting of psychometric Society.*, 35:321–377, 2001.

M. A. Alam and K. Fukumizu. Kernel and feature search in kernel PCA. *IEICE Technical Report, IBISML2011-49*, 111 (275):47–56, 2011.

M. A. Alam and K. Fukumizu. Higher-order regularized kernel CCA. *12th International Conference on Machine Learning and Applications*, pages 374–377, 2013.

M. A. Alam and K. Fukumizu. Hyperparameter selection in kernel principal component analysis. *Journal of Computer Science*, 10(7):1139–1150, 2014.

M. A. Alam, M. Nasser, and K. Fukumizu. A comparative study of kernel and robust canonical correlation analysis. *Journal of Multimedia.*, 5:3–11, 2010.

C. Alzate and J. A. K. Suykens. A regularized kernel CCA contrast function for ICA. *Neural Networks*, 21:170–181, 2008.

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, 2000.

P. Arias, G. Arias, and G. Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. *In IEEE Computer Society conference on computer vision and pattern recognition*, pages 1–8, 2007.

S. Arlot. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

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

K. Bache and M. Lichman. UCI machine learning repository, 2013.

G. Bakir, J. Weston, and B. Schölkopf. Learning to find pre-images. *In S. Thrun, L. Saul, & B. Schölkopf (Eds.), Advances in Neural Information Processing Systems*, 16:449–456, 2004.

A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publishers, London, 2004.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

A. J. Branco, P. Filzmose C. Croux, and M. R. Oliviera. Robust canonical correlations: A comparative study. *Computational Statistics*, 20:203–229, 2005.

L. Breiman and T. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598, 1985.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

S. Danafar, A. Gretton, and J. Schmidhuber. Characteristic kernels on structured domains excel in robotics and human action recognition. *Proc. European conf. Mach. learn. and knowledge discovery in databases: Part I, , Lecture Notes in Arti. Intell*, pages 264–279, 2010.

P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics*, 12(3):793–815, 1984.

L. Dümbgen and P. D. Counte-Zerial. On low-dimensional projections of high-dimensional distributions. *IMS Collections, From Probability and Statistics and Back: High-Dimensional Models and Processes*, 9:91–104, 2013.

G. E. Fasshauer. Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:11–63, 2011.

Y. Feng and Y. Liu. A cellular automata model based on nonlinear kernel principal component analysis for urban growth simulation. *Environment and Planning B: Planning and Design*, 40(1):116–134, 2013.

M. Filannino. *DBWorld e-mail classification using a very small corpus*. Project of Machine Learning course, University of Manchester, 2011.

K. Fukumizu and C. Leng. Gradient-based kernel dimension reduction for regression. *Journal of the American Statistical Association*, 109(550):359–370, 2014.

K. Fukumizu, F. R. Bach, and A. Gretton. Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8:361–383, 2007.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37:1871–1905, 2009.

K. Fukumizu, L. Song, and A. Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14:3753–3783, 2013.

T. Gärtner. *Kernels for Structured Data*. World Scientific, New Jersey, 2008.

A. Gretton. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723 – 773, 2012.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. *In Advances in Neural Information Processing Systems*, 20:585–592, 2008.

F. R. Hampel, E. M. Ronchetti, and W. A. Stahel. *Robust Statistics*. John Wiley & Sons, New York, 1986.

D. R. Hardoon and J. Shawe-Taylor. Convergence analysis of kernel canonical correlation analysis: theory and practice. *Machine Learning*, 74:23–38, 2009.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. *Neural Computation*, 16:2639–2664, 2004.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2009.

F. Hille. Introduction to general theory of reproducing kernels. *Rocky Mountain Journal of Mathematics*, 2(3):321–368, 1972.

H. Hofmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40:863–874, 2007.

T. Hofmann, B. Schölkopf, and J. A. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36:1171–1220, 2008.

P. Honeine and C. Richard. A closed-form solution for the pre-image problem in kernel-based machines. *Journal of Signal Processing Systems*, 63(3):289–299, 2011.

H. Hotelling. Relations between two sets of variables. *Biometrika*, 28:321–377, 1936.

S. Y. Huang, M. Lee, and C.K. Hsiao. Nonlinear measures of association with kernel canonical correlation analysis and applications. *Journal of Statistical Planning and Inference*, 139:2162–2174, 2009a.

S. Y. Huang, Y. R. Yeh, and S. Eguchi. Robust kernel principal component analysis. *Neural Computation*, 21(11):3179–3213, 2009b.

P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley & Sons, England, 2009.

A. J. Izenman. *Modern Multivariate Statistical Techniques*. Springer, New York, 2008.

R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall, New Jersey, 2007.

C.T. Kelley. *Iterative Methods for Optimization*. SIAM, Philadelphia, 1999.

D. W. Kim, K. Y. Lee, D. Lee, and K. H. Lee. Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition*, 38(4):607–611, 2005.

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

D. King. *Canonical correlation analysis of functional data*. UMI Dissertation publishing, Arizona State University, 2009.

E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, Canada, 1989.

W. J. Krzanowski. Cross-validation in principal component analysis. *Biometrics*, 43:575–584, 1987.

J. T. Kwok and I. W. Tsang. The pre-image problem in kernel methods. *In Machine learning, proceedings of the twentieth international conference (ICML2003)*, 38:408–415, 2003.

P. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *Computing and Information Systems*, 7:43–49, 2000.

K. H. Liang. K-fold crossvalidation in canonical analysis. *Multivariate Behavioral Research*, 30(4):539–545, 1995.

R. A. Marrona, D. R. Martin, and V. J. Yohai. *Robust Statistics: Theory and Methods*. John Wiley and Sons, New York, 2006.

P. G. Martin, H. Guillou, F. Lasserre, S. Déjean, A. Lan, J. M. Pascussi, M. San Cristobal, P. Legrand, P. Besse, and T. Pineau. Novel aspects of ppar?-mediated regulation of lipid and xenobiotic metabolism revealed through a multrigenomic study. *Hepatology*, 54 (2): 767–777, 2007.

E. Meckes. Quantitative asymptotic of graphical projection pursuit. *Electronic Communications in Probability*, 14:176–185, 2009.

T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extractrion using generalized canonical correlation analysis. *In Proceeding of International Conference on Artificial Neural Networks(ICANN)*, pages 353–360, 2001.

J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, A 209(441458): 415446, 1909.

S. Mika, B. Schölkopf, J. A. Smola, K.-B. Müller, and G. Rätsch. Kernel PCA and denoising in feature spaces. *Advances in Neural Information Processing Systems*, 11:536–542, 1999.

N. Otopal. Restricted kernel canonical correlation analysis. *Linear Algebra and its Applications*, 437:1–13, 2012.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

B. Premanode, J. Vongprasert, N. Sopipan, and C. Toumazou. A novel multiclass support vector machine algorithm using mean reversion and coefficient of variance. *Journal of Mathematics and Statistics*, 9:208–218, 2013.

S. J. Press. *Applied Multivariate Analysis*. Holl, Rinchart and Winston, INC, New York, 1987.

Y. Rathi, S. Dambreville, and A. Tannenbaum. Statistical shape analysis using kernel PCA. *Proc. SPIE 6064, Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, 60641B, 2006.

M. Reed and B Simon. *Methods of Modern Mathematical Physics*. Academic Press, California, 1980.

S. Ren, P. Ling, M. Yang, Y. Ni, and Z. Song. Multi-kernel pca with discriminant manifold for hoist monitoring. *Journal of Applied Sciences*, 13:4195–4200, 2013.

D. Samarov, J.S. Marron, Y. Liu, C. Grulke, and A. Tropsha. Local kernel cannonical correlation analysis with application to virtual drug screening. *The Annals of Applied Statistics*, 5:2169–2196, 2011.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning (ICML1998)*, pages 515–521, 1998.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge MA, 2002.

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

C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. *Proc.* 17*th Int. Conf. Pattern Recognition, ICPR*, 3:32–36, 2004.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge Univ. Press, 2004.

D. Skocaj, A. Leonardis, and S. Fidler. Robust estimation of canonical correlation coefficients. *Proceedings of OAGM04*, 20:15–22, 2004.

A. Smola and B. Schölkopf. A tutorial on support vector regression. *NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK*, 1998.

L. Song, A. Smola, K. Borgwardt, and A. Gretton. Colored maximum variance unfolding. *Advances in Neural Information Processing Systems*, 20:1385–1392, 2008.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, New York, 2008.

I. Steinwart and C. Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and rkhss. *Constructive Approximation*, 35:363–417, 2012.

M. Stone. Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

H. Suetani, Y. Iba, and K. Aihara. Detecting generalized synchronization between chaotic signals : a kernel-based approach. *Journal of Physics A: mathematical and General*, 39: 10723–10742, 2006.

J. A. K. Suykens, C. Alzate, and K. Pelckmans. Primal and dual model representations in kernel-based learning. *Statistics Surveys*, 4:148–183, 2010.

S. Taskinen, C. Croux, A. Kankainen, E. Ollila, and H. Oja. Canonical analysis based on scatter matrices. *Journal of Multivariate Analysis*, 97(2):359–384, 2006.

C. L. Teo, Y. Yang, H. D. III, C. Fermü, and Y. Alömono. Towards a watson that sees: Language-guided action recongnition for robots. *Inte. Conf. Robot. and Auto., ICRA*, pages 14–18, 2012.

A. B. Woen and P. O. Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3(2):564–594, 2009.

S. Wold. Cross-validation estimation of the number of components in factor and principal components models. *Technometrics*, 20:397–405, 1978.

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 (in ISMB)*, 19:i323–i330, 2003.

W-S. Zheng, J-H. Lai, and P. C. Yuen. Penalized pre-image learning in kernel principal component analysis. *Neural Networks*, 21(4):551–570, 2010.