Dimensionality Reduction by Kernel CCA in Reproducing Kernel Hilbert Spaces

Zhu Xiaofeng

NATIONAL UNIVERSITY OF SINGAPROE

2009

Dimensionality Reduction by Kernel CCA in Reproducing Kernel Hilbert Spaces

Zhu Xiaofeng

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE
2009

Acknowledgements

The thesis would never have been without the help, support and encouragement from a number of people. Here, I would like to express my sincere gratitude to them.

First of all, I would like to thank my supervisors, Professor Wynne Hsu and Professor Mong Li Lee, for their guidance, advice, patience and help. I am grateful that they have spent so much time with me discussing each problem ranging from complex theoretical issues down to the minor typo details. Their kindness and supports are very important to my work and I will remember them throughout my life.

I would like to thank Patel Dhaval, Zhu Huiquan, Chen Chaohai, Yu Jianxing, Zhou Zenan, Wang Guangsen, Han Zhen and all the other current members in DB 2 lab. Their academic and personal helps are of great value to me. I also want to thank Zheng Manchun and Zhang Jilian for their encouragement and support during the period of difficulties. They are such good and dedicated friends.

Furthermore, I would like to thank the National University of Singapore and School of Computing for giving me the opportunity to pursue advanced knowledge in this wonderful place. I really enjoyed attending the interesting

courses and seminars in SOC. The time when I spent studying in NUS might be one of the most memorable parts in my life.

Finally, I would also like to thank my family, who always trust me and support me in all my decisions. They taught me to be thankful and made me understand that experience is much more important than the end result.

Contents

Summary

In the thesis, we employ a multi-modal method (i.e., kernel canonical correlation analysis) named RKCCA to implement dimensionality reduction for high dimensional data.

Our RKCCA method first maps the original data into the Reproducing Kernel Hilbert Space (RKHS) by explicit kernel functions, whereas the traditional KCCA (referred to as spectrum KCCA) method projects the input into high dimensional Hilbert space by implicit kernel functions. This makes the RKCCA method more suitable for theoretical development. Furthermore, we prove the equivalence between our RKCCA and spectrum KCCA. In RKHS, we prove that RKCCA method can be decomposed into two separate steps, i.e., principal component analysis (PCA) followed by canonical correlation analysis (CCA). We also prove that the rule can be preserved for implementing dimensionality reduction in RKHS. Experimental results on real-world datasets show the presented method yields better performance than the sate-of-the-art algorithms in terms of classification accuracy and the effect of dimensionality reduction.

List of Tables

List of Figures

List of Symbols

Ω : metric spaces

H: Hilbert Spaces

X: matrix

X^T : the superscript T denote the transposed of matrix X

W: the directions of matrix X projected

ρ : Correlation coefficient

$\sum$: covariance matrix

k: kernel function

K: kernel matrix

$\mathbb{N}$:Natural number

$\mathbb{R}$:Real number

$k(.,x)$: a function of dot which is called a literal, and x is a parameter.

f(x): a real valued function

$\psi(x)$: a map from the original space into spectrum feature spaces

$\phi(x)$: a map from the original space into reproducing kernel Hilbert spaces

$\aleph$: the number of dimensions in a RKHS

Chapter 1

Introduction

1.1 Background

Recent applications, such as text categorization, computer vision, image retrieval, microarray technology and visual recognition, all involve high dimensional data [1, 2]. With the prevalence of high dimensional data in real life applications, the definition of “high dimensional" is also changing from tens of features to hundreds or even tens of thousands of features.

In principle, a learning algorithm is expected to perform more accurately given more information. In other words, we should utilize as many features as possible that are available in our data. However, in practice, although we have seen some cases with large amounts of high dimensional data that have been analyzed with high-performance contemporary computers, several problems occur when dealing with such high dimensional data. First, high dimensional data leads to an explosion in execution time. This is always a fundamental problem

when dealing with such datasets. The second problem is that some attributes in the datasets often are just “noise" or irrelevant to the learning objective, and thus do not contribute to (sometimes even degrade) the learning process. Third, high dimensional data suffer from the problem of “curse of dimensionality”. Hence, designing efficient solutions to deal with high dimensional data is both interesting and challenging.

The underlying assumption for dimensionality reduction is that data points do not lie randomly in the high dimensional space, and thus useful information in high dimensional data can be summarized by a small number of attributes. The main idea of dimensionality reduction is to solve a problem defined over a high dimensional geometric space Ω^d, by mapping that space onto Ω^k where k is “low” (usually, k << d) without losing much information in the original data, then solve the problem in the latent space. Most existing algorithms follow the theorem by Johnson and Lindenstrauss [3] which states that there exists a randomized mapping $\boldsymbol{A}$: $\Omega^d \to \Omega^k$, $k = O(long(1/P)/\varepsilon^2)$ such for any $x \in \Omega^d$, have

$$\Pr_A(\|Ax\|_2 = (1 \pm \varepsilon)\|x\|_2) \geq 1 - P \qquad \text{(Eq.1.1)}$$

where $P = \frac{1}{n^{O(1)}}$, n is the sample size and ε is a scalar approximate to zero. The equation means the probability of the difference between the original dataset and the dataset reduced with projection $\boldsymbol{A}$ always almost approaches 1, i.e., there is a little information loss after dimensionality reduction. Often Eq.1.1 may denote

the minimum classification error that a user is willing to accept, or some principles based on mutual information [4], such as, maximum statistical dependency ($\max\{I(\{x_i, i=1,...,m\};c)\}$), maximum relevance ($\max \frac{1}{|S|}\sum_{x_i \in S} I(x_i;c)$), and minimum redundancy ($\max \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i;x_j)$), where feature set S has m features x_i and class feature c, and $I(x_i;x_j)$ is the mutual information between feature x_i and feature x_j .

In order to satisfy the above rule, dimensionality reduction techniques should be designed to search efficiently for a mapping $\boldsymbol{A}$ such that satisfying Eq.1.1 for the given dataset. A naïve search algorithm performs an exhaustive search among all combinations of 2^d subspaces and finds the best subspace. Clearly this is exponential and not scalable. Alternate methods typically employ some heuristic sequential-search-based methods, such as best individual features and sequential forward (floating) search [4].

Dimensionality reduction can solve the problem of high dimensional data by reducing the number of attributes in the dataset, thus saving both storage space and CPU time required to process the smaller dataset. In addition, interpreting the learned models is easier with a smaller number of attributes. Furthermore, by transforming the high dimensional data into low dimensional data (say 2D or 3D), it is much simpler to visualize and obtain a deeper understanding of the data characteristics. Hence, dimensionality reduction techniques have been regarded as one of the efficient methods for dealing with the high dimensional data.

However, dimensionality reduction can result in certain degree of information loss. Inappropriate reduction can cause useful and relevant information to be filtered out. To overcome this, researchers found some solutions. For example, naive Bayes classifier can classify high dimensional data sets accurately for certain application, and some regularized classifiers (such as support vector machine) can be designed to achieve good performance for high dimensional text datasets [9]. Furthermore, some learning algorithms, such as, boosting methods or mixture models, can build separate models for each attribute and combine these models, rather than performing dimensionality reduction. Despite the apparent robustness of the methods mentioned above, dimensionality reduction is still useful as a first step in data preparation. That is because noise/irrelevant attributes can degrade the learning performance, and this issue can be eliminated as much as possible by effectively performing dimensionality reduction [5]. Furthermore, taking into consideration the savings in time and storage requirement of a learning model, the suggestion for dimensionality reduction is reasonable. However, how to more effective perform dimensionality reduction still is an interesting or challengeable issue. Hence, in this thesis, we will focus on the issue of dimensionality reduction.

1.2 Motivations and Contributions

Many learning frameworks for dimensionality reduction have been proposed in [6-8, 77] as well as survey papers on dimensionality reduction can be found in [1,

9-11]. The details can be found in Chapter 2 of the thesis. In the thesis, we focus on implementing dimensionality reduction with canonical correlation measures, i.e., kernel canonical correlation analysis (KCCA). Canonical correlations are invariant with respect to affine transformations of the variables. This is the most important difference between CCA and the other ordinary correlation analysis (such as, Pearson correlation coefficient, Kendall τ and Spearman ρ) which highly depend on the representations in which the variables are described [40]. To the best of our knowledge, there is no literature focused on implementing dimensionality reduction with KCCA method. Traditional KCCA method (referred to as spectrum KCCA in the thesis) maps the original feature space to a higher dimensional Hilbert space of real valued functions. However, the approach suffers from at least two main limitations. First, the mapping used in spectrum KCCA method is often implicit which is not conducive to theoretical development [46]. Second, the regularization step employed by spectrum KCCA method requires the setting of many parameters. Moreover, to obtain the optimal parameter setting requires prior knowledge on the datasets.

In this thesis, we first survey the existing literatures on dimensionality reduction techniques. Then we propose a method named RKCCA (Kernel Canonical Correlation Analysis in RKHS) in which we map the original data into reproducing kernel Hilbert spaces (RKHS). In the RKHS, we perform dimensionality reduction with kernel canonical correlation analysis (KCCA) measure by two separate steps, i.e., principal component analysis (PCA) followed

by canonical correlation analysis (CCA). Furthermore, we apply for RKCCA into the learning models in all kinds of learning models, such as, supervised learning model, unsupervised learning model, and transfer learning model. Our contributions are summarized as follows:

- Propose an efficient algorithm to implement dimensionality reduction by Kernel canonical correlation analysis in reproducing kernel Hilbert spaces.
- Prove that the equivalence between the traditional KCCA (referred to as spectrum KCCA in this thesis) and our KCCA in RKHS (i.e., RKCCA).
- Prove that RKCCA can be decomposed into two separate processes, i.e., PCA followed by CCA in RKHS, also proved that the rule is preserved for implementing dimensionality reduction by RKCCA in RKHS.
- Test the effect of dimensionality reduction with KCCA measures in all kinds of learning models, such as, supervised learning model, unsupervised learning model and transfer learning model.

1.3 Organization

The thesis is organized as follows. We give an overview of the existing literatures on dimensionality reduction techniques in Chapter 2 and present some preliminary theory about CCA and KCCA in Chapter 3. In Chapter 4, we propose the RKCCA approach; and we evaluate the proposed approach on real-world datasets in Chapter 5. We conclude our work and proposed future research work in Chapter 6.

Chapter 2

Related Work

In this section, we provide an overview of the existing dimensionality reduction techniques from three aspects:

1) linear versus nonlinear techniques based on the relationships between independent variables and dependent variable, the details can be found in section 2.1;
2) means by which low dimensional data are formed: feature selection, feature extraction, feature grouping techniques; details are given in section 2.2;
3) learning models: supervised learning techniques, unsupervised learning techniques, semi-supervised learning techniques, multi-view techniques and transfer learning techniques; details are described in section 2.3.

2.1 Linear Versus Nonlinear Techniques

Traditional linear dimensionality reduction techniques include principal component analysis (PCA), factor analysis (FA), projection pursuit (PP), singular value decomposition (SVD), independent component analysis (ICA).

Recently, researchers in [11] argued that data in real-life applications are often too complex to be captured by the simple linear models. Instead, kernel methods can be applied to provide a non-linear analysis. For example, Kernel PCA (KPCA) method can (implicitly) construct a higher (even indefinite) dimensional space, in which a large number of linear relations between the independent variables and dependent variable can be easily built in high dimensional spaces. Subsequently, the low-dimensional data is obtained by applying traditional PCA in the higher dimensional spaces.

Other popular nonlinear dimensionality reduction techniques (e.g., [11-13]) include principal curves, random projection, locally linear embedding etc. In this thesis, we are interested in nonlinear dimensionality reduction techniques.

2.2 Techniques for Forming Low Dimensional Data

Based on the techniques for forming low dimensional data, dimensionality reduction techniques can be broadly divided into several categories [9]: (i) feature selection techniques, (ii) feature extraction techniques, and (iii) feature grouping techniques.

Feature selection approaches try to find a subset of the original attributes such that the information in that subset can approximately represent the whole data set. It includes filter approaches (e.g. information gain, mutual information), wrapper approaches (e.g. genetic algorithm), and embedding approaches. Many feature selection methods belong to the supervised learning methods presented in section 2.3.

Feature extraction methods apply a projection of the multidimensional space to a low dimensional space. This projection may involve all the attributes in the dataset. Feature extraction measures (e.g., [12, 14]) are very popular in data mining and machine learning techniques, such as, PCA, semi-definite embedding method, multifactor dimensionality reduction method, Isomap method, latent semantic analysis method, wavelet compression method, semantic mapping method and the others methods. The proposed method in this thesis partially belongs to this domain because one of dimensionality reduction techniques in the thesis is principal component analysis (PCA).

Feature grouping techniques reduce the dimensions by combining several existing features to build one or more new features. The most direct way for feature grouping method is to cluster the features (rather than the objects) of a data set. For example, to cluster a similarity matrix of different features by applying the clustering method (e.g., hierarchical clustering method) [2], then evaluate the result of the cluster with Pearson's correlation coefficient. Another example in [9], instead of clustering the traditional clustering methods, we can also cluster together for both the attributes and the objects, e.g., co-clustering method. Feature grouping can indirectly achieve some similar coefficients by combining ridge regression with LASSO [15] which is a penalized least squares method imposing an *L1*-penalty on the regression coefficients.

2.3 Techniques Based on Learning Models

Dimensional reduction techniques can be categorized into five types based on the types of learning models built, namely: supervised learning methods, unsupervised learning methods, semi-supervised learning methods, multi-view methods and transfer learning methods.

2.3.1 Unsupervised Learning Techniques

Unsupervised dimensional reduction techniques usually refer to techniques that perform dimensionality reduction based only on the condition attributes without

considering the information from class labels. Among the traditional unsupervised dimensional reduction methods, such as, PCA, ICA and random projection, random projection method is the most promising as it is not as computationally expensive as the others.

Recently, Weinberger et al., [16] proposed a nonlinear supervised dimensional reduction method. The method first learns a kernel matrix by preserving local distances for k nearest neighbors of each point to satisfy the maximum variance unfolding (MVU) principle. It then performs PCA in the high dimensional space after using the kernel trick to project the original data into a high dimensional space. In essence, the proposed dimensional reduction technique is similar to PCA. However, this method can preserve the local instances in latent spaces after dimensionality reduction while PCA only wants to assure the maximum separation rather than preserving the geometric distances.

Techniques on dimensionality reduction are also carried out as a pre-processing step to select the subspace dimensions before the clustering process. The most representative of this approach is the adaptive technique presented in [17] which adjusts the subspace adaptively to form clusters are best separated or well defined. Another adaptive technique on dimensionality reduction is presented in [18] which employs K-means clustering to generate class labels and uses linear discriminant analysis (LDA) to select subspaces. The data are then simultaneously clustered while the feature subspaces are selected. This method builds a bridge between the clusters discovered in the subspace and those defined

in the full space by effectively using the cluster membership. This allows clusters that are discovered in the low dimensional subspace to be adaptively re-adjusted for global optimality.

In the unsupervised learning domain, Cevikalp et al., [19] recently proposed a discriminative linear dimensionality reduction method aim at preserving separateability by using the weighted displacement vectors between the training samples and nearby rival class regions to choose the projection directions.

2.3.2 Supervised Learning Techniques

Supervised learning techniques are designed to find a low dimensional transformation by considering class labels. In fact, class labels in supervised dimensionality reduction techniques can be used together with the condition attributes to extract relevant features. For example, both linear discriminant analysis (LDA) methods and multiple discriminant analysis methods can find the effective projection directions by maximizing the ratio of between-class variance to within-class variance. The partial least squares (PLS) method presents the same function as the regression edition of LDA. The Canonical correlation analysis (CCA) method, which finds projection directions by maximizing the correlation between two variables, is also regarded as one of techniques on supervised dimensionality reduction. Some traditional linear supervised

algorithms (e.g., above examples mentioned) can be transformed into nonlinear measure by kernel trick and are presented in [2, 20, 21].

Recent supervised dimensionality reduction techniques aim to minimize loss before and after dimension reduction [4]. This loss may be measured in terms of a cost function, degree of discrepancy, degree of dependence, class information distance [2], k nearest neighbor classification error [20]. For instance, Sajama and Orlitsky in [22] approximated the data distributions to any desired accuracy based on the maximum conditional likelihood estimation of mixture models, while retaining the maximum possible mutual information between feature vectors and class labels in the selected subspace by using the conditional likelihood as the contrast function. Cater et al. [2] employed the information preserving component analysis (IPCA) method to maximize the information distances. Rish et al. [23] combined learning a good predictor with dimensionality reduction but ignoring the "noise" by minimizing the conditional probability of class given the hidden variables.

2.3.3 Semi-supervised Learning Techniques

Semi-supervised dimensionality reduction techniques learn from a combination of both labeled and unlabeled data. In many practical data mining applications, unlabeled data are readily available but labeled data are more expensive to be obtained, therefore techniques on semi-supervised dimensionality reduction are

more practical than the techniques on supervised dimensionality reduction or unsupervised dimensionality reduction techniques. Existing techniques on semi-supervised dimensionality reduction are usually built based on the unsupervised model by combining with prior information, such as, class label, pairwise constraints, side information.

A popular technique is semi-supervised learning algorithm based on graph, which considers a graph over all the samples as prior information to guide learning. The weight matrix, in which the weight of the edge between points in different classes is zero and a positive real value for the points with same classes, is the key to the semi-supervised learning algorithms based graph for classification problems. In the framework presented in [27], a projected subspace can be learnt from the labeled data by supervised learning method. Then, the weight matrix is obtained by combining not only the relationship between the mapped points in the subspace but also the labeled points. In order to obtain the weight matrix, there are two existing techniques. For example, we can assume that points that are near are likely to have the same label. We can also assume that the p-nearest neighbor graph is preserved between the original spaces and the subspaces.

The supervised methods, such as, least square method, or linear discriminant analysis (LDA) algorithm, encounter the ill-posed problems (i.e., within-class scatter matrix is singular) when data size is smaller than the number of the features. By combining the relationship between regularized least-squares and

regularized discriminant analysis, Song et al., [7] added a regularization term to the original criteria of LDA. The regularization term in the eigen problem is based on the prior knowledge coming from both labeled and unlabeled data, and can be constructed to employ graph Laplacian, to avoid the ill-posed problem during the process of dimensionality reduction. This transforms the original supervised model into semi-supervised model. Therefore, under their framework, some classical methods, such as principal component analysis (PCA), linear discriminant analysis (LDA), maximum margin criterion (MMC), locality preserving projections (LPP) and their corresponding kernel versions will be the special cases of the proposed method.

Pairwise constraint is an information pair of instances known as belonging to the same class (must-link constraints) or different classes (cannot-link constraints) rather than knowing the actual class label of the instances, and it arises naturally in many tasks [24], such as, image retrieval. In the real life applications, pairwise constraint is more general than class labels because true labels are difficult to obtain due to lack of prior knowledge, while specifying a pairwise constraint (i.e., whether some pairs of instances belong to the same class or not) is easier. Moreover, the pairwise constraints can be implied from labeled data but not vice versa. What is more, the pairwise constraints can be automatically obtained without human intervention [25]. For example, Bar-Hillel et al. [25] proposed the constrained Fisher's Linear Discriminant (cFLD) for dimensionality reduction from equivalence constraints (only for must-link constraint) as an interim-step for

Relevant Component Analysis (RCA). Tang and Zhong [26] used pairwise constraints to guide dimensionality reduction, which can exploit both must-link constraints and cannot-link constraints but does not consider the usefulness of abundant unlabeled data. Zhang, et al., [24] considered the problem by combining unlabeled data with pairwise constraints.

Recently Zhang et al., [28] effectively used the information from class labels and the information learnt with online method from unlabeled data without the assumption of existence of classes to implement dimensionality reduction. The method uses a ranking rule for the class label and does not require an actual class label.

Prior information can be obtained from experts or by performing experiments. Some of these prior information may be exact or inexact. Yang et al. [29] extended the traditional nonlinear unsupervised techniques on dimensionality reduction (such as, Locally Linear Embedding method, ISOMAP method, and Local Tangent Space Alignment (LTSA)) to semi-supervised model by considering the prior information aim at yielding global low dimensional coordinates as well as bearing the same physical meaning deriving from the prior information. Weinberger and Saul [30] first learnt a kernel matrix aim at maximum variance unfolding (MVU) for k nearest neighbor distances of original data, then performed PCA to implement dimensionality reduction after projecting the original data into high dimensions by kernel matrix learnt. The proposed method also belongs to nonlinear technique. Based on the maximum variance

unfolding (MVU), Song et al., [31] learned a kernel matrix to preserve the local distance of data points as well as add the side information in the process, then built a semi-supervised model.

All above methods on semi-supervised dimensionality reduction models are designed based on unsupervised model. To the best of our knowledge, there is no literature focusing on the supervised model.

2.3.4 Multi-view methods

All the above techniques (such as, unsupervised learning techniques, supervised learning techniques, or semi-supervised learning techniques) are designed for dealing with the data in one dataset. For the case with multiple views (there are multiple views and one feature for class label in one dataset, and each view can correctly separate the class label without the help from the other views) in one dataset, we call the dimensionality reduction methods as multi-view methods. For example, Foster et al., [32] presented a nonlinear unsupervised technique on dimensionality reduction with canonical correlation analysis. In the proposed algorithm, the algorithm first performs CCA technique in unlabeled data $\{(X^{(1)}, X^{(2)})\}$. Then it constructs a projection Π that projects $(X^{(1)}, X^{(2)})$ to the most correlated lower dimensional subspace by selecting a (or several) maximal correlation coefficients. Finally, with a labeled dataset $\{(X^{(1)}, X^{(2)}, Y)\}$, a least squares regression is performed in this low dimensional subspace.

2.3.5 Transfer learning methods

Most of the former methods, i.e., supervised dimensionality reduction methods, unsupervised methods and semi-supervised methods, are focused on one dataset to implement dimensionality reduction. Given the limited information in the dataset, for example, only one class label in the dataset, previous methods are unable to build an effective classifier. To overcome this, external datasets may be employed and this is the motivation in transfer learning. Transfer learning [33-35] is to learn a new task through the transfer of knowledge from a related task which has already been learned or easily to be learned a model (we also call the related task as outer information or source dataset due to it is not in the target dataset). The objective of transfer learning is to improve learning performance in the target task by the help from the source task. This can present significant improve while there is a little information in the target task or the useful information is too expensive to obtain.

Dimensionality reduction techniques on transfer learning model are first put forward in [36, 37]. Intuitively, dimensionality reduction techniques in transfer learning model are more practical and general than the traditional techniques on dimensionality reduction, so it will be the research topic in this thesis.

Compared to dimensionality reduction with linear discriminant analysis (LDA), transferred dimensionality reduction (TDR) method [36] has two improvements. First, transferred dimensionality reduction method revises the measure of the between-class information of LDA. The second improvement is

the revision of the composite adjacency matrix of neighborhood graphs. In the TDR algorithm, given initial k classes for target data, the algorithm is iteratively computed till the algorithm converges. Then it is designed applying traditional LDA to do dimensionality reduction for receiving optimal result. The paper also presented nonlinear transferred dimensionality reduction (TDR) by kernel functions.

Dimensionality reduction method with transfer learning model presented in [37] is based on the nonlinear supervised techniques on dimensionality reduction methods presented in [30, 38]. There are two steps in the framework. First, the algorithm extracts the common latent spaces between source and target datasets based on the maximal mean dependency embedded (MMDE) principle. In the common latent space extracted, the prior information is added into the learning process of kernel matrix. The objective is to maximize the dependence on the matrix which includes the side information and original information. In the second step of the proposed algorithm, the classifier built from source data in latent spaces is employed to classify target dataset in latent spaces. The whole algorithm is a KPCA-style method and extended from [30]. The last method in [30] receives the distances by kernel function with Hilbert-Schmidt Independence Criterion (HSIC) as well as considers side information, and it is regarded as a technique on semi-supervised methods.

Comparing the method in [36] with the method in [37], all two papers transfer prior information (i.e., class label) under the semi-supervised framework. The

difference is: Wang et al., [36] transfer information by summing the basic information (the information of independent variables in two datasets) and prior information (class label in target dataset, for strength the ability of dimensionality reduction; whereas Pan, et al. [37] compose the basic information with prior information into high dimensional spaces by kernel trick, then perform learning in the traditional semi-supervised learning model.

2.4 The proposed method

In this thesis, the proposed the algorithm RKCCA: 1) belongs to a nonlinear dimensionality reduction technique as it employs kernel methods; 2) can be categorized into feature extraction method for it uses PCA method as one of its two process; 3) can be applied to many kinds of datasets in the supervised learning model, unsupervised model (i.e., multi-view method) and transfer learning model.

Chapter 3

Preliminary Work

Some measures of relationship between two sets of variables have been popular in machine learning domains because they can reduce noise by correlation analysis. These methods include Pearson correlation coefficient, Kendall τ and Spearman ρ [39], mutual information [4] and canonical correlation analysis [40].

Canonical correlation analysis (CCA) method, which searches for two diagonal representations with maximal correlations of the two original variables, is a way of measuring the linear relationship between two variables. An interesting characteristic of canonical correlations on CCA is that they are invariant with respect to affine transformations of the variables. This is the most important difference between CCA and the other ordinary correlation analysis which highly depend on the representations in which the variables are described. Therefore, initially proposed as a multivariate analysis method by Hotelling [41], CCA and its variants have been widely applied to all kinds of domains, such as,

image processing [40, 42], pattern recognition [43], computer vision [44], wireless network[45] and the other domains.

3.1 Basic theory on CCA

Assuming two random variables: $X^{(1)} \in \Omega^p$ and $X^{(2)} \in \Omega^q$, we can consider the relationship between $X^{(1)}$ and $X^{(2)}$ by choosing appropriate directions $W_{CCA}^{(1)}$ (and $W_{CCA}^{(2)}$) of $X^{(1)}$ (and $X^{(2)}$) to let: $S^{(1)} = W_{CCA}^{(1)T} X^{(1)}$, and $S^{(2)} = W_{CCA}^{(2)T} X^{(2)}$, then we can find the relationship between $X^{(1)}$ and $X^{(2)}$ and let

$$\rho = \max_{W_{CCA}^{(1)}, W_{CCA}^{(2)}} corr(S^{(1)}, S^{(2)}) \qquad \text{(Eq.3.1)}$$

After receiving the covariance matrix of the observed sample, i.e.,

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \qquad \text{(Eq.3.2)}$$

Then the maximum canonical correlation between $X^{(1)}$ and $X^{(2)}$ can be changed into:

$$\rho = \max_{W_{CCA}^{(1)}, W_{CCA}^{(2)}} \frac{W_{CCA}^{(1)\,T} \Sigma_{12} W_{CCA}^{(2)}}{\sqrt{W_{CCA}^{(1)\,T} \Sigma_{11} W_{CCA}^{(1)}} \sqrt{W_{CCA}^{(2)\,T} \Sigma_{22} W_{CCA}^{(2)}}} \qquad \text{(Eq.3.3)}$$

We use A^T to denote the transpose of matrix A throughout this thesis.

Due to the arbitrary of scale, the optimization problem in Eq.3.3 can be equaled to maximizing the numerator in Eq.3.3 subject to:

$$W_{CCA}^{(1)T} \sum_{11} W_{CCA}^{(1)} = 1, \quad W_{CCA}^{(2)T} \sum_{22} W_{CCA}^{(2)} = 1 \tag{Eq. 3.4}$$

Thus, its corresponding Lagrangian is

$$L(\lambda^{(1)}, \lambda^{(2)}, W_{CCA}^{(1)}, W_{CCA}^{(2)}) = W_{CCA}^{(1)T} \sum_{12} W_{CCA}^{(2)} - \sum_{i=1}^{2} \frac{\lambda^{(i)}}{2} (W_{CCA}^{(i)T} \sum_{11} W_{CCA}^{(i)} - 1) \tag{Eq.3.5}$$

After derivatives in respective to $W_{CCA}^{(1)}$ and $W_{CCA}^{(2)}$, we can obtain

$$\begin{cases} \dfrac{\partial L}{\partial W_{CCA}^{(1)}} = \sum_{12} W_{CCA}^{(2)} - \lambda^{(1)} \sum_{11} W_{CCA}^{(1)} = 0 \\ \dfrac{\partial L}{\partial W_{CCA}^{(2)}} = \sum_{21} W_{CCA}^{(1)} - \lambda^{(2)} \sum_{22} W_{CCA}^{(2)} = 0 \end{cases} \tag{Eq. 3.6}$$

Multiplying with $W_{CCA}^{(1)T}$ (and $W_{CCA}^{(2)T}$) to the two equations in Eq. 3.6 and subtracting their results, we can easily know

$$\begin{aligned} 0 &= W_{CCA}^{(1)T} \Sigma_{12} W_{CCA}^{(2)} - \lambda^{(1)} W_{CCA}^{(1)T} \Sigma_{11} W_{CCA}^{(1)} - \lambda^{(2)} W_{CCA}^{(2)T} \Sigma_{22} W_{CCA}^{(2)} + \lambda^{(2)} W_{CCA}^{(1)T} \Sigma_{11} W_{CCA}^{(1)} \\ &= \lambda^{(2)} W_{CCA}^{(1)T} \Sigma_{11} W_{CCA}^{(1)} - \lambda^{(1)} W_{CCA}^{(1)T} \Sigma_{11} W_{CCA}^{(1)} \\ &= \lambda^{(2)} - \lambda^{(1)} \end{aligned} \tag{Eq. 3.7}$$

and we let $\lambda^{(1)} = \lambda^{(2)} = \lambda$.

Assuming $\sum_{22}$ is invertible, then the optimization problem in Eq. 3.3 is transferred into an eigenproblem as:

$$\begin{cases} \sum_{12}\sum_{22}^{-1}\sum_{21}W_{CCA}^{(1)} - \lambda^2\sum_{12}W_{CCA}^{(1)} = 0 \\ \sum_{21}\sum_{11}^{-1}\sum_{12}W_{CCA}^{(2)} - \lambda^2\sum_{21}W_{CCA}^{(2)} = 0 \end{cases} \quad \text{(Eq. 3.8)}$$

Or

$$\begin{pmatrix} & X^{(1)}X^{(2)^T} \\ X^{(2)}X^{(1)^T} & \end{pmatrix}\begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix} = \lambda\begin{pmatrix} X^{(1)}X^{(1)^T} & \\ & X^{(2)}X^{(2)^T} \end{pmatrix}\begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix} \quad \text{(Eq. 3.9)}$$

Although we can obtain the optimization result of ρ (the correlation coefficient) by solving the eigenproblem in Eq. 3.8 (or Eq. 3.9), the CCA method difficultly extract useful representations of the data in real application. That is because, 1) CCA method assumes the two original variables following Gaussian distribution; 2) its linearity.

Hence, researchers extended the linear CCA into nonlinear CCA in which the relationship between two variables can be dealt with by nonlinear relationship. Popular nonlinear CCA methods have statistical methods (i.e., step function method, B-splines) [47] and the methods on machine learning, such as, neural network methods based on CCA[48, 49] and kernel function methods based on CCA (i.e., KCCA) [40, 50]. In this thesis, we focus on the methods in machine learning. Unfortunately, in real applications, neural networks based on CCA method suffer from some intrinsic problems such as long-time training, slow convergence and local minima [44]. KCCA is a good alternative because it can perform linear separation of the data simply via mapping the original spaces to the high (or infinite) dimensional spaces.

3.2 Basic theory on KCCA

Researchers consider to replacing CCA with KCCA in which the data will be projected into high dimensional data for linearly separating, and we will introduce the traditional KCCA method following the idea in [40] but with a little improvement.

Given two input data $X^{(1)} \in \Omega^p$ and $X^{(2)} \in \Omega^q$ with sample size *n*. We map both $X^{(1)}$ and $X^{(2)}$ into high (even infinite) dimensional spaces Ω^P and Ω^Q ($P \geq p$, $Q \geq q$), via the implicit mappings

$$\psi^{(1)} : X^{(1)} \mapsto \psi^{(1)}(X^{(1)}) = (\psi_1^{(1)}(X^{(1)}),...,\psi_P^{(1)}(X^{(1)})) \quad \text{(Eq. 3.10)}$$

and

$$\psi^{(2)} : X^{(2)} \mapsto \psi^{(2)}(X^{(2)}) = (\psi_1^{(2)}(X^{(2)}),...,\psi_Q^{(2)}(X^{(2)})) \quad \text{(Eq. 3.11)}$$

where $\psi^{(i)}(X^{(i)})$(i=1, 2) is the kernel spectrum for a certain positive definite kernel, i.e.,

$$k_i(x_j^{(i)}, x_l^{(i)}) = \psi^{(i)}(x_j^{(i)})^T \psi^{(i)}(x_l^{(i)}), \ (x_j^{(i)}, x_l^{(i)} \in X^{(i)}, \text{i=1, 2 and } j,l = 1,...,n) \quad \text{(Eq. 3.12)}$$

and the corresponding kernel matrix is

$$K_i = \left\{k_i(x_j^{(i)}, x_l^{(i)})\right\}_{j,l=1}^{n} \text{ (i=1, 2)} \quad \text{(Eq. 3.13)}$$

After the original data $X^{(i)}$ are projected into kernel matrix K_i (i=1, 2) by a kernel function, based on the Eq. 3.3, we assume the projection direction on $X^{(1)}$ or $X^{(2)}$ is $W_{KCCA}^{(1)}$, or $W_{KCCA}^{(2)}$ respectively, the linear relationship between $W_{KCCA}^{(1)\ T}K_1$ and $W_{KCCA}^{(2)\ T}K_2$ (i.e., the nonlinear relationship between $X^{(1)}$ and $X^{(2)}$) can be substituted as:

$$\rho = \max_{W_{KCCA}^{(1)}, W_{KCCA}^{(2)}} \frac{W_{KCCA}^{(1)\ T}K_1K_2W_{KCCA}^{(2)}}{\sqrt{W_{KCCA}^{(1)\ T}K_1K_1W_{KCCA}^{(1)}}\sqrt{W_{KCCA}^{(2)\ T}K_2K_2W_{KCCA}^{(2)}}} \quad \text{(Eq. 3.14)}$$

Due to the arbitrary of scale, the optimization problem in Eq. 3.14 can be equaled to maximize the numerator in Eq. 3.14 subject to:

$$W_{KCCA}^{(1)\ T}K_1K_1W_{KCCA}^{(1)} = 1, \text{ and } W_{KCCA}^{(2)\ T}K_2K_2W_{KCCA}^{(2)} = 1 \quad \text{(Eq. 3. 15)}$$

Thus, its corresponding Lagrangian is

$$L(\lambda^{(1)}, \lambda^{(2)}, W_{KCCA}^{(1)}, W_{KCCA}^{(2)}) = W_{KCCA}^{(1)\ T}K_1K_2W_{KCCA}^{(2)} - \sum_{i=1}^{2}\frac{\lambda^{(i)}}{2}(W_{KCCA}^{(i)\ T}K_iK_iW_{KCCA}^{(i)} - 1)$$

(Eq.16)

After derivatives in respective to $W_{KCCA}^{(1)}$ and $W_{KCCA}^{(2)}$, we can obtain

$$\begin{cases} K_1K_2W_{KCCA}^{(2)} - \lambda^{(1)}K_1K_1W_{KCCA}^{(1)} = 0 \\ K_2K_1W_{KCCA}^{(1)} - \lambda^{(2)}K_2K_2W_{KCCA}^{(2)} = 0 \end{cases} \quad \text{(Eq. 3.17)}$$

The traditional methods (e.g., [40, 46]) always directly assume $\lambda^{(1)} = \lambda^{(2)}$ without explaining anything. In fact, the assumption $\lambda^{(1)} = \lambda^{(2)}$

is true, and we will prove $\lambda^{(1)} = \lambda^{(2)}$ instead of assuming it, and the process is presented Lemma 3.1.

Lemma 3.1 ***If equations*** $\begin{cases} K_1K_2W_{KCCA}^{(2)} - \lambda^{(1)}K_1K_1W_{KCCA}^{(1)} = 0 \\ K_2K_1W_{KCCA}^{(1)} - \lambda^{(2)}K_2K_2W_{KCCA}^{(2)} = 0 \end{cases}$ ***are consistent, then*** $\lambda^{(1)} = \lambda^{(2)}$**.**

Proof: we employ the pseudo inverse method to change Eq. 3.17 into:

$$\begin{cases} (K_1K_1)^{-}K_1K_2W_{KCCA}^{(2)} = \lambda^{(1)}W_{KCCA}^{(1)} \\ (K_2K_2)^{-}K_2K_1W_{KCCA}^{(1)} = \lambda^{(2)}W_{KCCA}^{(2)} \end{cases} \quad \text{(Eq. 3.18)}$$

where $(K_1K_1)^{-}$ and $(K_2K_2)^{-}$ is the pseudo inverse of matrix K_1K_1 and K_2K_2 respectively. Based on the definition of the pseudo inverse, we know

$$(K_1K_1)(K_1K_1)^{-}K_1K_2 = K_1K_2, \text{ and } (K_2K_2)(K_2K_2)^{-}K_2K_1 = K_2K_1 \quad \text{(Eq. 3. 19)}$$

Based on Eq. 3.15,

$$\lambda^{(1)} = \lambda^{(1)}W_{KCCA}^{(1)\ T}K_1K_1W_{KCCA}^{(1)},\ \lambda^{(2)} = \lambda^{(2)}W_{KCCA}^{(2)\ T}K_2K_2W_{KCCA}^{(2)} \quad \text{(Eq. 3. 20)}$$

Based on Eq. 3.18, we can get:

$$\begin{aligned} \lambda^{(1)} &= W_{KCCA}^{(1)\ T}K_1K_1\lambda^{(1)}W_{KCCA}^{(1)} \overset{Eq.3.18}{=} W_{KCCA}^{(1)\ T}K_1K_1((K_1K_1)^{-})^{T}K_1K_2W_{KCCA}^{(2)} \\ &= W_{KCCA}^{(1)\ T}K_1K_2W_{KCCA}^{(2)} \end{aligned} \quad \text{(Eq. 3.21)}$$

$$\begin{aligned} \lambda^{(2)} &= W_{KCCA}^{(2)\ T}K_2K_2\lambda^{(2)}W_{KCCA}^{(2)} \overset{Eq.3.18}{=} W_{KCCA}^{(2)\ T}K_2K_2((K_2K_2)^{-})^{T}K_2K_1W_{KCCA}^{(1)} \\ &= W_{KCCA}^{(2)\ T}K_2K_1W_{KCCA}^{(1)} \end{aligned} \quad \text{(Eq. 3.22)}$$

Obviously, the maximal relationship between K_1 and K_2 in Eq. 3.21 is equivalent to the maximal relationship between K_2 and K_1 in Eq. 3.22.

Hence, $\lambda^{(1)} = \lambda^{(2)}$, and we let $\lambda^{(1)} = \lambda^{(2)} = \lambda$. □

Based the Lemma 3.1, we can get the eigenproblem based on kernel matrix:

$$\begin{cases} K_1 K_1 W_{CCA}^{(1)} - \lambda^2 K_1 K_1 W_{CCA}^{(1)} = 0 \\ K_2 K_2 W_{CCA}^{(2)} - \lambda^2 K_2 K_2 W_{CCA}^{(2)} = 0 \end{cases} \quad \text{(Eq. 3.23)}$$

Or

$$\begin{pmatrix} & K_1 K_2 \\ K_2 K_1 & \end{pmatrix} \begin{pmatrix} W_{KCCA}^{(1)} \\ W_{KCCA}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} K_1 K_1 & \\ & K_2 K_2 \end{pmatrix} \begin{pmatrix} W_{KCCA}^{(1)} \\ W_{KCCA}^{(2)} \end{pmatrix} \quad \text{(Eq. 3.24)}$$

Both Eq. 3.3 and Eq. 3.14 belong to a generalized eigenproblem with the form $AX = \lambda BX$. However, the eigenproblem in either CCA method or KCCA method suffer the singular problem. That is to say, both K_1 and K_2 ($\sum_{11}$ and $\sum_{22}$) maybe be singular or near singular when the dimensions on $X^{(1)}$ and $X^{(2)}$ are larger than the sample size. This can cause numerical instability and computational efficiency. So the optimization in Eq.3.3 and Eq. 3.14 will be ill-posed. In order to solving these issues, some regularization methods are employed. For example, 1) regularizing with partial least squares (or ridge-style regression methods) to penalize the norms of the associated weights for avoiding overfitting and ill-conditioned; 2) to stabilize the numerical computation for solving problem by adding a small quantity to the diagonals, or 3) to perform dimensionality reduction with Gram-Schmidt orthogonalization method or

incomplete Cholesky decomposition method for reducing complexity of the algorithm, and among others.

After these preprocesses, Eq. 3.14 in [4] can be changed into:

$$S^{-1}Z_{12}(Z_{22}+\kappa I)^{-1}Z_{21}(S^{-1})^T\hat{\alpha}=\lambda^2\hat{\alpha} \qquad \text{(Eq. 3.25)}$$

where $K_1 \triangleq R_1R_1^T = Z_{11} = SS^T$, $K_2 \triangleq R_2R_2^T = Z_{22}$, $R_1^TR_2 = Z_{12}$, $R_2^TR_1 = Z_{21}$, $\hat{\alpha} = S^TW_{CCA}^{(1)}$, and κ is a scalar, symbol $\triangleq$ means approximate equivalent.

The solution to Eq. 3.14 can also provides a set of eigenvectors $v_{1,j}^{(i)},...,v_{d,j}^{(i)}$ and the corresponding eigenvalues $\lambda_1^{(i)},...,\lambda_d^{(i)}$, i=1, 2. We sort these corresponding eigenvalues from the largest to the smallest based on Scholkopf and Smola in [51], the d-dimensional embedding that best preserves inner products in high dimensional spaces is obtained by the mappings ($\psi^{(i)}: X^{(i)} \to (\sqrt{\lambda_1^{(i)}}v_{1,j}^{(i)},...,\sqrt{\lambda_d^{(i)}}v_{d,j}^{(i)})$ ($d > \max(p,q)$) and the kernels k_i can be expressed in terms of its eigenvectors v_α and eigenvalues λ_α as

$$k_i = \sum_\alpha \lambda_\alpha^{(i)} v_\alpha^{(i)} v_\alpha^{(i)T} \qquad \text{(Eq. 3.26)}$$

This can be regarded as a spectrum decomposition of k_i, thus we call this method of KCCA (e.g., [40, 50]) as the spectrum KCCA throughout this thesis.

Recently, KCCA method [74] is one of popular research areas. In the theory of KCCA, Kuss and Raepel [52] explained how the canonical correlation between configurations of points mapped into kernel feature spaces can be

determined and preserved the geometry of the original points. Yamada et al., [53] studied the relationship between spectrum KCCA (as an unsupervised model) and KFDA (as a supervised model). As in statistical methods, many parameters are usually estimated from finite samples, the convergence of the estimated functions should be considered to justify the estimation method. Since the objective of spectrum KCCA is to estimate the relationship of a pair functions, it is necessary to evaluate its convergence. Hence, Fukumizu, et al., [54] rigorously proved the statistical consistency of spectrum KCCA and the consistency of a pair of functions for expressing the nonlinear dependence in two variables. Yamaish et al., [55] extended the spectrum KCCA application for two datasets into multiple datasets. And Blaschko and Lampert in [56] explained why using paired data can reduce the effects of noise by considering the covariance matrix of paired data with independent additive noise.

Except the application domains mentioned in section 3.1 on CCA, as traditional unsupervised learning model, spectrum KCCA has been successfully applied in all kinds of learning models, such as, supervised learning model [57], multi-view model [58-59], and semi-supervised model [60, 61] for some real assignments, such as, classification [56], regression [58], clustering [61], and testing for independence [46, 62] in all kinds of practical domains, such as, chaotic time series [63], Climate forecasting [75], media information retrieval [64], analysis of fMRI data [40], text mining [65], extraction of gene clusters [55, 73], and independent component analysis [66].

However, on the one hand, many parameters must be simultaneously set in spectrum KCCA method, such as, the precision parameter, regularization parameter and the others. Moreover, it is difficult to correctly set them by manual, i.e., to obtain the optimal parameters setting needs prior knowledge. For example, the Gram-Schmidt orthogonalization algorithm in [40] will not be regularized well if setting a larger precision parameter even for 0.1 in our comparison experiments. In this case, we wish the designed algorithm can be easily operated by the researchers in all kinds of domains even if the researcher are unfamiliar machine learning, such as, statisticians or the other practitioners.

On the other hand, spectrum KCCA method maps the original data into the high dimensional space by an implicit representation which is not convenient for theoretical development. For example, assuming a mapping $\psi : X \mapsto \psi\ (X)$, where the feature map $\psi(x, x') = (x^2, xx', x'^2)$ maps data in Ω^2 into Ω^3, then

$$\begin{aligned} k(\alpha,\beta) &= <\psi(\alpha),\psi(\beta)> \\ &= \alpha^2\beta^2 + 2\alpha\alpha'\beta\beta' + \beta^2\beta'^2 \\ &= (\alpha\beta + \alpha'\beta')^2 \\ &= (<\alpha,\beta>)^2 \end{aligned} \qquad \text{(Eq. 3.27)}$$

Based on the above example, we only need to compute the inner product (i.e., kernel matrix) such that the original data in Ω^2 can be projected into Ω^3, and we even do not care what the representation of ψ is. This does not convenient to theoretical development [46] due to the implicit kernel function.

Chapter 4

KCCA in RKHS

In this chapter, we propose a novel approach, called RKCCA method, which can overcome the limitations of spectrum KCCA described in the end of Chapter 3 and is equivalence to spectrum KCCA method. Instead of projecting the original data into the Hilbert space (or spectrum feature spaces), our RKCCA algorithm maps the original data into the Reproducing Kernel Hilbert Space (RKHS) of continuous values function based on some positive definite kernels (details presented in section 4.1). The RKHS, in which we aim to construct a theoretical framework for implementing dimensionality reduction, are smaller than Hilbert spaces of smooth functions but sufficient to capture non-parametric phenomena of interest [67]. To eliminate the regularization for coding by users, we first prove that KCCA in RKHS is the same as PCA followed by CCA in RKHS. Then we transform the regularization process of our RKCCA algorithm into CCA whose

regularization process has been embedded in many existing software, such as, Matlab, SPSS. This can reduce run time and lessen programming.

Before we describe the details of our algorithm RKCCA, we need to prove that the mapping of input to a RKHS is unique as well as feasible. Then we show that KCCA in RKHS is equivalent to spectrum KCCA. With this, we proceed to establish that KCCA in RKHS is the same as Kernel PCA followed by CCA in RKHS. Finally, we prove that performing dimensionality reduction using KCCA in RKHS is equivalent to dimensionality reduction with PCA in RKHS followed by a further reduction with CCA in RKHS. The details are provided in lemmas 4.2 to lemma 4.5.

4.1 Mapping Input into RKHS

Given a positive definite kernel function and a variable $\boldsymbol{X}$ with zero-mean and unit-variance, we define an explicit mapping

$$\phi : x \mapsto \phi(x) = k_i(.,x) \qquad \text{(Eq. 4.1)}$$

where $k(.,x) = (k(x_1,x),...,k(x_n,x))$, $x, x_i \in X$. Note that, $k(.,x)$ means a function of the expression '*dot*' which is called a literal in mathematics or logic, and x is a parameter.

Next, we construct a dot product space (denoted as $< \cdot, \cdot >$) containing the input under ϕ in two steps. First, we form the vector space containing all linear combinations, e.g.,

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \qquad \text{(Eq. 4.2)}$$

Second, we define a dot product between $f(\cdot)$ and another vector space $g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x_j)$ as follows:

$$< f, g >= \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(\cdot, x_i) k(\cdot, x_j) \qquad \text{(Eq. 4.3)}$$

This dot product can be proved to satisfy the symmetry, bilinearity and positive definiteness conditions based on Lemma 4.1.

Lemma 4.1 The dot product space constructed by the order steps presented in Eq. 4.2 and Eq. 4.3 satisfy the symmetry, bilinearity and positive definiteness conditions.

Proof:

1. Symmetry:

$$\begin{aligned} < f, g >= \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(\cdot, x_i) k(\cdot, x_j^{'}) &= \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x_j^{'}) \\ = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_j^{'}, x_i) &= \sum_{i=1}^{m'} \sum_{j=1}^{m} \beta_j \alpha_i k(\cdot, x_i) k(\cdot, x_j^{'}) \\ =< g, f > & \end{aligned} \qquad \text{(Eq. 4.4)}$$

2. Bilinearity:

$$< f, g >= \sum_{i=1}^{m} \alpha_i g(x_i) = \sum_{j=1}^{m'} \beta_j g(x_j^{'}) \quad \text{(Eq. 4.5)}$$

3. Positive definiteness:

$$< f, f >= \alpha^T K \alpha \geq 0 \text{ with equality if only if f = 0} \quad \text{(Eq. 4.6)}$$

□

Such a dot product space under a Hilbert space is called a reproducing kernel Hilbert space (RKHS). A RKHS is a Hilbert space of continuous valued functions (i.e., bounded and linear) with an explicit expression (i.e., $\phi(x) = k_i(., x)$). Hence, a RKHS has the following properties based on [22]:

$$< f, f >= \| f \|^2 = \left\| \sum_{i=1}^{m} \alpha_i k(x_i, x_i) \right\|^2 = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \quad \text{(Eq. 4.7)}$$

$$< k(\cdot, x), k(\cdot, x') >= k(x, x') \text{ or } f(x) =< f, k(\cdot, x) >, \quad x, x' \in X \quad \text{(Eq. 4.8)}$$

The kernel $k(x, x')$ is called the reproducing kernel satisfying the reproducing property presented in Eq. 4.8.

4.2 Theorem for RKCCA

Lemma 4.2 ***Given a Mercer kernel k, there exists a RKHS space H, such that*** $x \to \phi(x) = k(.,x)$ ***, where*** $< \phi(x), \phi(x') >= k(x,x') \quad x, x' \in X$ ***, and the reproducing kernel*** $k(x,x')$ ***is uniquely determined by the space H.***

Proof.

First, we show that there exists a RKHS in *H* for each Mercer kernel *k*.

Based on Mercer theory in [22] and Eq. 3.14:

$$f(x) \overset{Eq.3.14}{=} \sum_{i=1}^{\infty} \alpha_i k(x, x_i) = \sum_{i=1}^{\infty} \alpha_i \sum_{j=1}^{\aleph} \lambda_j \phi_j(x) \lambda_j \phi_j(x_i) \qquad \text{(Eq. 4.9)}$$

Where $\aleph$ is the number of dimensions in a RKHS.

By the linearity of inner product, the Eq. 4.8 is transformed into

$$\langle f, k(,x) \rangle = \sum_{i=1}^{\infty} \alpha_i \sum_{j,n=1}^{\aleph} \lambda_j \phi_j(x_i) \langle \phi_j, \phi_n \rangle \lambda_n \phi_n(x) \qquad \text{(Eq. 4.10)}$$

Since *k* is a Mercer kernel, the ϕ_j (j=1,…, $\aleph$) can be chosen to orthogonal. Hence, based on [22], let

$$\langle \phi_j, \phi_n \rangle = C / \lambda_j \qquad \text{(Eq. 4.11)}$$

where C is the Kronecker symbol in [22].

Based on Eq.4.10 and Eq. 4.11, the reproducing property in Eq. 4.8 is preserved.

Next, we prove that the reproducing kernel is unique.

Let $k(x,x')$ be a reproducing kernel of H. Assuming there exists another different reproducing kernel $k'(x,x')$ of H. Then for all $x \in X$, applying the reproducing property for k and k', we get

$$\begin{aligned} \left\| k_x - k_x^{'} \right\|^2 &\overset{Eq.4.7}{=} \left\langle k_x - k_x^{'}, k_x - k_x^{'} \right\rangle = \left\langle k_x - k_x^{'}, k_x \right\rangle - \left\langle k_x - k_x^{'}, k_x^{'} \right\rangle \\ &\overset{Eq.4.8}{=} (k_x - k_x^{'})(x) - (k_x - k_x^{'})(x) = 0 \end{aligned} \qquad \text{(Eq. 4.12)}$$

Hence, $k_x = k_x^{'}$, that is, $k_x(x') = k_x^{'}(x')$ for all $x' \in X$. This is means $k(x,x') = k'(x,x')$ for $x, x' \in X$. □

Based on Lemma 4.2, any space can be mapped into a smooth space by a unique kernel in RKHS. Hence, it is feasible for us to map the input data into a RKHS.

After projecting the input into RKHS, we proceed to prove the equivalence between KCCA in RKHS (i.e., RKCCA) and spectrum KCCA. This is achieved by showing that the isomorphic characteristic in spectrum KCCA is preserved in RKHS, i.e., there is a one-to-one mapping between them.

Lemma 4.3 There is a one-to-one projection between $\psi(x)$ on spectrum KCCA and the mapping $\phi(x)$ on RKCCA.

Proof: we first prove that "$\phi(x) \Rightarrow \psi(x)$".

Based on the Mercer's theorem, for the continuous positive definite kernel $k(x,x')$ in RKHS, there exists an integral operator $I:\Omega^X \to \Omega^X$, and $(If)(x) = \int I(x,x')f(x)dx'$, where $x, x' \in X$. Since $k(x,x')$ is symmetric and positive definite, it is orthogonally diagonalizable as in the case with finite dimensions. Thus, $k(x,x')$ can be expressed as $k(x,x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(x')$, by its ordered eigenvectors series $\phi_i(x)$ and corresponding eigenvalues series λ_i. Based on Dauxois and Nkiet in [47], the nonlinear CCA can be approximated as, $k(x,x') = \sum_{i=1}^{n} \lambda_i \phi_i(x)\phi_i(x')$, in terms of uniform convergence of a certain underlying sequence. Hence, RKCCA can be implemented spectrum decompositions similar to Eq. 3.26.

Next, we prove "$\psi(x) \Rightarrow \phi(x)$".

By combining Eq. 4.1 and Eq. 4.7 with Eq. 4.8, for any $x \in X$, we have

$$\|\psi(x)\|_{\psi}^2 = k(x,x) = < k(x,.), k_i(x,.) >_{\phi} = \|\phi(x)\|_{\phi}^2 \qquad \text{(Eq. 4.13)} \quad \square$$

Lemma 2 shows that KCCA in RKHS is equivalent to spectrum KCCA.

Lemma 4.4 *KCCA in RKHS can be decomposed into PCA followed by CCA in RKHS.*

Proof: Given positive definite kernel functions k_1, k_2 and two centered variables (i.e., zero-mean and unit-variance) $X^{(1)} \in \Omega^p$, $X^{(2)} \in \Omega^q$, and a mapping: $\phi : x^{(i)} \to \phi(x^{(i)}) = k_i(., x^{(i)})$ in RKHS. After performing PCA in RKHS, the original data $X^{(i)}$ becomes $\tilde{X}^{(i)} = W_{PCA}^{(i)\,T} \phi(\cdot, X^{(i)})$, where $W_{PCA}^{(i)}$ is the projected directions of $X^{(i)}$. Then for two variables $X^{(i)} and\ X^{(j)}$, based on the reproducing property presented in Eq. 4.8, we can get

$$\tilde{X}^{(i)} \tilde{X}^{(j)T} = W_{PCA}^{(i)\,T} \phi(\cdot, X^{(i)}) \phi(\cdot, X^{(j)})^T W_{PCA}^{(j)} = W_{PCA}^{(i)\,T} K_{i,j} W_{PCA}^{(j)} \qquad \text{(Eq. 4.14)}$$

After performing CCA by solving Eq. 3.3, the result can be denoted as a generalized eigenproblem, i.e.,

$$\begin{pmatrix} & X^{(1)} X^{(2)^T} \\ X^{(2)} X^{(1)^T} & \end{pmatrix} \begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} X^{(1)} X^{(1)^T} & \\ & X^{(2)} X^{(2)^T} \end{pmatrix} \begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix}$$

(Eq. 4.15)

where $W_{CCA}^{(i)}$ is the projected directions of $X^{(i)}$ by CCA. We prove if we apply the result of PCA to the input data of CCA in RKHS, the result will be equivalent to directly implementing KCCA in RKHS where $W_{RKCCA}^{(i)}$ is the projected directions of $X^{(i)}$ in RKHS. Based on Eq. 4.14,

$$Eq.15 \Leftrightarrow \begin{pmatrix} & W_{PCA}^{(1)\,T} K_1 K_2 W_{PCA}^{(2)} \\ W_{PCA}^{(2)} K_2 K_1 W_{PCA}^{(1)\,T} & \end{pmatrix} \begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} W_{PCA}^{(1)\,T} K_1 K_1 W_{PCA}^{(1)} & \\ & W_{PCA}^{(2)\,T} K_2 K_2 W_{PCA}^{(2)} \end{pmatrix} \begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} W_{PCA}^{(1)\,T} & \\ & W_{PCA}^{(2)\,T} \end{pmatrix} \begin{pmatrix} & K_1K_2 \\ K_2K_1 & \end{pmatrix} \begin{pmatrix} W_{PCA}^{(1)} & \\ & W_{PCA}^{(2)} \end{pmatrix} \begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix}$$
$$= \lambda \begin{pmatrix} W_{PCA}^{(1)\,T} & \\ & W_{PCA}^{(2)\,T} \end{pmatrix} \begin{pmatrix} K_1K_1 & \\ & K_2K_2 \end{pmatrix} \begin{pmatrix} W_{PCA}^{(1)} & \\ & W_{PCA}^{(2)} \end{pmatrix} \begin{pmatrix} W_{CCA}^{(1)} \\ W_{CCA}^{(2)} \end{pmatrix}$$

$$\Leftrightarrow \quad \begin{pmatrix} & K_1K_2 \\ K_2K_1 & \end{pmatrix} \begin{pmatrix} W_{PCA}^{(1)}W_{CCA}^{(1)} \\ W_{PCA}^{(2)}W_{CCA}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} K_1K_1 & \\ & K_2K_2 \end{pmatrix} \begin{pmatrix} W_{PCA}^{(1)}W_{CCA}^{(1)} \\ W_{PCA}^{(2)}W_{CCA}^{(2)} \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} & K_1K_2 \\ K_2K_1 & \end{pmatrix} \begin{pmatrix} W_{RKCCA}^{(1)} \\ W_{RKCCA}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} K_1K_1 & \\ & K_2K_2 \end{pmatrix} \begin{pmatrix} W_{RKCCA}^{(1)} \\ W_{RKCCA}^{(2)} \end{pmatrix}$$

where $W_{RKCCA}^{(i)} = W_{PCA}^{(i)} W_{CCA}^{(i)}$, i=1, 2. (Eq. 4.16)

□

Lemma 4.5 Dimensionality reduction in RKHS is equivalent to dimensionality reduction by PCA followed by CCA in RKHS.

Proof: Bach and Jordan [66] showed that PCA, CCA and KCCA can be expressed as generalized eigenproblems. Given a dataset $\phi(X)$, the projected directions by PCA, CCA, and KCCA in RKHS are denoted by $W_{PCA}^{(1)}$, $W_{CCA}^{(1)}$, and $W_{RKCCA}^{(1)}$ respectively. So the corresponding result of dimensionality reduction with the three methods is denoted as $\tilde{\phi}_{PCA}(X) = W_{PCA}^{(1)\,T}\phi(X)$, $\tilde{\phi}_{CCA}(X) = W_{CCA}^{(1)\,T}\phi(X)$, and $\tilde{\phi}_{RKCCA}(X) = W_{RKCCA}^{(1)\,T}\phi(X)$ respectively.

Since the result of dimensionality reduction by PCA (i.e., $W_{PCA}^{(1)\,T}\phi(X)$) is regarded as the input on implementing CCA based on Lemma 4.4, the result of dimensionality reduction with CCA will be

$$\tilde{\phi}_{CCA}(X) = W_{CCA}^{(1)\,T}\tilde{\phi}_{PCA}(X) = W_{CCA}^{(1)\,T}(W_{PCA}^{(1)\,T}\phi(X)) \qquad \text{(Eq. 4.17)}$$

Then

$$\tilde{\phi}_{CCA}(X) = (W_{PCA}^{(1)}W_{CCA}^{(1)})^{T}\phi(X) \qquad \text{(Eq. 4.18)}$$

Based on Eq. 4.16, this result can be expressed as

$$(W_{PCA}^{(1)}W_{CCA}^{(1)})^{T}\phi(X) = W_{RKCCA}^{(1)\,T}\phi(X) = \tilde{\phi}_{RKCCA}(X)\,. \qquad \text{(Eq. 4.19)}$$ □

4.3 Extending to Mixture of Kernels

The quality of a non-parametric learning method is not only determined by its ability to learn from the data (i.e., interpolation) but also its ability to predict unseen data (i.e., extrapolation). Jordaan [68] argued that the two characteristics are largely determined by the choice of kernel in kernel methods. Jordaan [68] and Zheng et al. [69] showed a global kernel (such as the polynomial kernel) can present better extrapolation abilities at lower-order degrees, but lack of good interpolation even if with high-order degree. And a local kernel (such as Gaussian kernel) has good interpolation abilities, but fails to provide longer range extrapolation. Based on analysis, we may receive the better interpolation and extrapolation by combining the local kernel and the global kernel.

In this thesis, our RKCCA algorithm can replace the single kernel function by mixture of kernels defined as

$$k_{mix} = \omega k_p + (1-\omega)k_g . \quad \text{(Eq. 4.20)}$$

where $k_p = (< x, x_i > +1)^q$ is polynomial kernel, and $k_g = \exp(-(x-x_i)^2 / \sigma^2)$ is Gaussian kernel, q ($q \in \mathbb{N}$) and σ ($\sigma \in \mathbb{R}$) is the corresponding bandwidth in kernel functions, the weight ω ($0 \leq \omega \leq 1$).

The issue of choosing the optimal parameters settings for q, σ, ω to achieve a better generalization performance in a learning task is called model selection. Existing methods for model selection include grid search methods, cross-validation methods, uniform design method, and among others [70]. In the thesis, we will propose two strategies to solve the issue of model selection.

First, based on [22], the polynomial kernel should be set with a lower-order degree q, and the Gaussian kernel should have a smaller σ value. In our experiments, we set q <=10, and 0< σ <5. Our second strategy is to uniformly select the optimal parameters by uniform design for experiments with mixtures method (referred to as UDEM method in the thesis), which is designed to seek the design points to be uniformly scattered on the experimental domain.

The algorithm for model selection of RKCCA algorithm (i.e., UMED method) is presented as follows, and the details of setting parameters will be explained in the end of this chapter.

1. Choose parameters search ranges (the number of parameters is denoted as *s*), determine a suitable levels for each parameter based on the first strategy, and the number of level is denoted as *n*. (Note that, in this UD-web, the authors assumed all parameters containing some levels, otherwise, the different levels for parameters will be change into same

level).

2. Choose a suitable UD table to accommodate the number of parameters and levels for UD-web.
3. From the UD table, randomly determine the run order of experiments and conduct the performance evaluation of each parameter combination in the UD, and denoted the element of the UD table as $\{q_{ik}\}$, k (or i) is the number of parameters (or level).
4. Receiving $\{x_{ki}\}$ based on Eq. (4-21), then $\{x_{ki}\}$ is fed into the step.3 in RKCCA algorithm.

The uniform experimental design is one kind of space filling designs that have been used for all kinds of experiments, such as, computer domain, industrial domain and the others.

Suppose there are s parameters in a domain Ω^S, and we want to choose a set of points $P_m = \{p_1, p_2, ..., p_m\} \subset \Omega^S$ which are uniformly scattered over the domain Ω^S. Let $F(\theta)$ (or $F_m(\theta)$) be the cumulative uniform distribution function over Ω^S (or the empirical cumulative distribution function of P_m). Let L_2- discrepancy of non-uniformity of P_m be

$$D(\Omega^S, P_m) = \left[\int_{\Omega^S} \left| F_m(\theta) - F(\theta) \right|^2 d\theta \right]^{1/2} \quad \text{(Eq. 4.21)}$$

The search for uniform designs with minimum L_2- discrepancy is an NP-hard problem [71]. Thus approximated methods are designed to find low

discrepancy (i.e., closing the theoretical minimum discrepancy), such as, centered L_2- discrepancy in [71]. A complete list of the uniform design (UD) tables can be found in UD-web (http://www.math.hkbu.edu.hk/ UniformDesign) based on the centered L_2 –discrepancy principle.

The UDEM method can uniformly set experimental plans by considering the recipe (i.e., the parameter ω) of the parameters (q and σ) into UD method.

Assuming the element of the UD table is denoted as $\{q_{ik}\}$, k (or i) is the number of parameters (or level). We define an intermediate variable c_{ki}, and let

$$c_{ki} = \frac{2q_{ki} - 1}{2n}, \text{ k=1,…, n} \qquad \text{(Eq. 4.22)}$$

Then the weight of x_{ki} for s parameters with n levels is uniformly set based on UDEM method as:

$$\begin{cases} x_{ki} = (1 - c_{ki}^{1/s-i}) \prod_{j=1}^{i-1} c_{ki}^{1/s-j}, \text{ i=1,...,s-1} \\ x_{ks} = \prod_{j=1}^{s-1} c_{ki}^{1/s-j}, \text{ k=1,...,n} \end{cases} \qquad \text{(Eq. 4.23)}$$

Based on the UDEM method, all the test points are uniformly selected in the experimental plan. However, the method does not consider the border points. However, the optimal results are often found in the border of the test range. One simple remedy is to add the border points into the experimental plans. In Eq. 4.20, the border point is the pair (0, 1) and (1, 0) respectively for the weight ω.

4.4 RKCCA Algorithm

Many real-life datasets contain thousands of features and result in the generation of singular kernel matrix. However, the 'intrinsic' dimensions in these datasets are typically low. In general, a pre-processing step is first performed to obtain the datasets consisting of only the 'intrinsic' dimensions. For this purpose, we utilize the random projection method due to its linear complexity and high accuracy. Next we perform the mapping to a RKHS by explicit representations after centering the data at mean 0 with unit variance. Then we perform covariance analysis using the Kernel PCA method to remove noise and redundancy before the data are fed into a CCA tool. Finally, after CCA, we perform dimensionality reduction on the result of CCA. Details of the RKCCA algorithm are presented as follows:

Function [OutputData]= RKCCA ($X^{(1)}$, $X^{(2)}$, *r*, *k*, *c*)

% input: $X^{(1)}$ *and* $X^{(2)}$ *are two original data, r (or k or c) is number of dimensionality reduction with random projection (or PCA or CCA)*

% output: ***OutputData*** *is the data with reduced dimensions*

---

1. $X_{PR}^{(i)} = X^{(i)} * PR(r)$; *% Random Projection on* $X^{(i)}$

2. $X_{PR,centered}^{(i)}$; *% Centering data* $X_{PR}^{(i)}$ *, i=1, 2;*

3. $X_{RKHS}^{(i)}$ = K($X_{PR,centered}^{(i)}$, options);

% K() is kernel method, options: single or mixture of kernels

% the experimental plan comes from model selection presented in the last section.

4. [p s l t]= $X_{RKHS}^{(i)}$ ***princomp**(* $X_{RKHS}^{(i)}$) *% PCA( ): dimensional reduction with PCA*

---

5. [a b r u v] = **canoncorr** ($X_{RKHS}^{(1)} * p^{(1)}(:,1:k)$, $X_{RKHS}^{(2)} * p^{(2)}(:,1:k)$); % *CCA on* $X_{RKHS}^{(i)} * p^{(i)}(:,1:k)$
Output: v(:,1:*c*) or u(:,1:*c*)

Note that: the value of r, k, and c will be decided by users or the expertise; the parameters in function ***princomp*** *and* ***canoncorr*** *are same to the representation in* ***Matlab*** *software, and the details are presented in "****HELP****" part of Matlab.*

Comparing to the spectrum KCCA, the proposed method RKCCA presents some features as follows:

- The proposed method projects the original data into reproducing kernel Hilbert space which is smaller than Hilbert space but sufficient to find the linear functions for linearly separating the data in high dimensional spaces. Moreover, our RKCCA algorithm defines explicit kernel functions for convenient to the theoretical development, and spectrum KCCA defines implicit kernel functions. Furthermore, we also prove the equivalence between spectrum KCCA and our RKCCA.
- In RKHS, we prove RKCCA can be decomposed into two separately steps, i.e., PCA followed by CCA in RKHS. We also prove the dimensionality reduction by RKCCA can be decomposed into two processes of dimensionality reduction, i.e., PCA followed by CCA in the high dimensional space. There are at least two advantages. Firstly, this can increase the effect of dimensionality reduction by efficiently removing noise and redundancy. In fact, in PCA, its diagonal terms are

ordered in non-increase ordering, and all the off-diagonal terms are zero, so PCA can effective remove noise and redundancy by selecting parts of principal components. Secondly, our algorithm performs PCA to extract noise and redundancy in RKHS before implementing CCA because some noise or redundancy can easily be detected in high dimensional spaces rather than in original spaces.

- Our algorithm directly performs CCA in RKHS without the regularization process, which can reduce running time of the algorithm and lessening programming. In fact, our algorithm also needs to regularization, but we transfer the process into the CCA process which has been programmed well in the popular software, such as, Matlab. So the algorithm can reduce running time and lessening programming. And it can also be easily implemented and understood even if the users are with little knowledge or unfamiliar to the machine learning domain because the whole framework can be coded by tens line of codes and many codes can be employed the existing functions in software Matlab.
- The key theoretic advantage of the UDEM model selection over the other methods (such as, grid search) for model selection in our mixture of kernel is that the UDEM points are “far more uniform” and “far more space filling” than lattice grid points [70]. Moreover, basically the UDEM method can find good representative points uniformly scattered over the parameter domain to replace the lattice grid points for a much more efficient parameter search. Furthermore, the single kernel methods

become a special case of our proposed method. Therefore, although the mixture of kernels in RKCCA need to set three parameters, it can be designed with less running time and better performance than any single kernel methods only with a little discrepancy.

Chapter 5

Experimental Analysis

We evaluate the proposed RKCCA algorithm in terms of classification accuracy (or error rate) and its effectiveness in dimensionality reduction in this chapter. We compare the CCA algorithm [13], KCCA (KCCA algorithm in [13]), Kernel PCA (KPCA) [11] (or Kernel Fisher Discriminant Analysis (KDA) [11] for supervised learning models) with our RKCCA algorithm.

In our experiments, we first implement dimensionality reduction in the original data sets with these algorithms, such as, CCA, KCCA, KPCA (or KDA) and RKCCA. After reducing the dimensions (setting parameters on the number of dimensionality reduction can be found in Chapter 4), we use 10-fold cross validation method, in which k nearest neighbor (k = 8) classifier is employed, to get the classification accuracy of these algorithms in the reduced space. Each experiment is repeated 10 times and we record the average value.

For setting the parameters in mixture of kernels (such as, q, σ and ω), we employ uniform design for experiments with mixtures (UDEM) method presented in section 4.3 to uniformly design the experimental plans with 10 levels for each parameter and select the optimal parameter values by cross-validation method. The procedure is implemented with MATLAB (R2009b edition) software running in PC (Microsoft Windows XP, Intel Core 2 Duo CPU, 4GB of RAM).

5.1 Performance for Classification Accuracy

As a nonparametric method, KCCA algorithm (or RKCCA algorithm) has been focused on detecting the relationship between two variables in different learning models, for example, multi-view method [37, 61] (it can be regarded as an unsupervised model), supervised learning model [57]. However, no research has focused on transfer learning model with KCCA or RKCCA measure. In the section, we investigate the application on real-life datasets for dimensionality reduction by all methods presented above in three models, i.e., unsupervised learning model, supervised model and transfer learning model.

5.1.1 Unsupervised Learning Models

We first examine the performance for KCCA in dimensionality reduction under unsupervised learning model. We use the real world *ads* dataset for this set of

experiments. There are 3279 instances and 1558 features in the dataset with 5 views. We extract three views (i.e., *url* (457 features), *origurl* (495 features), and *ancurl* (472 features)) for our experiments and combine them to form 3 datasets as shown in Table 5.1. Each instance in the dataset corresponds to an image on the web, and the task is to predict whether an image is used for advertisement. In the preprocessing step, the dimensions are designed to 400 for each view in the Random Projection method. Table 5.1 gives the results for the various methods. The value in bracket is the standard deviation.

Table 5.1: Classification Accuracy in *Ads* Dataset.

	url+origurl	url+ancurl	origurl+ancurl
CCA	0.8722 (0.0187)	0.8618 (0.0104)	0.8607 (0.0132)
KCCA	0.8792 (0.0112)	0.9044 (0.0102)	0.8985 (0.0103)
KPCA	0.8840 (0.0162)	0.9151 (0.0075)	0.9138 (0.0134)
RKCCA	**0.8938 (0.0159)**	**0.9291 (0.0105)**	**0.9240 (0.0082)**

We observe that the RKCCA method consistently outperforms the rest of the methods. Comparing the kernel methods (such as, KCCA, KPCA and RKCCA) algorithms with CCA, we find that the classification accuracy of the kernel methods in all methods yield better performance than CCA method. This is because the relationship between independent variables and dependent variable in

real-life datasets can be better expressed by nonlinear relationship rather than linear one. Comparing kernel correlation analysis algorithms (i.e., RKCCA and KCCA) with KPCA method that performs classification only with the information from one dataset (e.g., *origurl* in the experiment *url+origurl*), the RKCCA gives better performance due to the availability of additional information (e.g., the *url* is regarded as the source data, and *origurl* as the target data in experiment *url+origurl*). Based on the analysis, in the two KCCA methods, our RKCCA algorithm presents better results because RKCCA can efficiently remove noise and redundancy by performing PCA and CCA separately.

5.1.2 Supervised Learning Models

CCA and KCCA (or RKCCA) methods are designed to deal with the relationship between vectors $X^{(1)}$ and $X^{(2)}$. If we regard the class label information as $X^{(2)}$, then CCA-based methods (i.e., CCA, KCCA, and RKCCA) can also serve as a supervised feature extraction method (but PCA is not feasible for this case, so we use KDA to replace it in this section). Existing literatures (such as, [57, 76]) in CCA-based methods usually employ some effective methods to deal with the class labels. In the thesis, we adopt the one-of-c label encoding.

In the supervised experiments, we test the performance of our KCCA algorithms comparing with CCA, KCCA, KDA method on two datasets, i.e., *scene* and *yeast* from LIBSVM data sets

(http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/). Dataset y*east* contains 2417 instances, 103 features, and 14 classes. *Scene* data is 2407 instances, 296 features and 6 classes. The experimental results are presented in Table 5.2, and the value in bracket is the standard deviation.

Table 5.2: Comparison of classification accuracy in dataset *yeast* and dataset

	Yeast	scene
CCA	0.9880 (0.0037)	0.9320 (0.0085)
KCCA	0.9912 (0.0032)	0.9340 (0.0028)
KDA	0.9880 (0)	0.9330 (0)
RKCCA	**0.9920 (0.0024)**	**0.9361(0.0014)**

We observe that the proposed method RKCCA outperforms all the other algorithms.

5.1.3 Transfer Learning Models

We use the *WiFi* dataset [37] and 20 newsgroups [72] (denoted as *news* in this paper) for this set of experiments. The *WiFi* dataset records *WiFi* signal strength in 135 small grids, each of which is about 1.5 *1.5 square meters, and has five domains collected in different time phrase, i.e., d0826 collected in 08:26am, d1112, d1354, d1621 and d1910 respectively. There are 7140 instances and 11

features with 119 classes in each dataset. We construct 2 datasets by combining the domains collected at different time phrase, such as, d0826 means the source dataset and d1910 means the target dataset in dataset "d0826+d1910". Dataset *news* contains approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. In our experiments, we select the domain *comp* as the source dataset and the domain *rec* as the target dataset, and the dimensions are designed as 500 for Random Projection method in the preprocess phrase. Table 5.3 gives the results for the various methods. The value in bracket is the standard deviation. Once again, we observe that RKCCA algorithm yields the best performance in transfer learning models in which the distribution of the source dataset is different from the distribution of the target dataset.

Table 5.3: Comparison of classification accuracy in *WiFi* and dataset *news*

	d0826 + d1910	d1112 + d1621	comp + rec
CCA	0.5006 (0.0227)	0.4970 (0.0213)	0.4989 (0.0178)
KCCA	0.5306 (0.0158)	0.5214 (0.0152)	0.6534 (0.0214)
KPCA	0.5974 (0.0176)	0.6024 (0.0206)	0.5723 (0.0092)
RKCCA	**0.6192 (0.0258)**	**0.6104 (0.0218)**	**0.6671 (0.0327)**

5.2 Performance of Dimensionality Reduction

Finally, we investigate the effect of dimensionality reduction on the error rate (error rate = 1- classification accuracy). We construct the kNN classifiers in the reduced spaces generated by the algorithms mentioned in section 5.1, and we also construct a classifier with the full original dimensions without implementing dimensionality reduction, named *Original*. Figure 5.1 shows that the proposed RKCCA method yields the best performance after implementing dimensionality reduction where the percent of dimensions reduced is 100%.

We also find the results of CCA are worse than the left methods except algorithm *Original*, i.e., the kernel methods, this shows kernel methods can more successfully find a subspace in which the classification can be preserved well even when the dimensionality is significantly reduced. Finally, kernel methods present better effect of dimensionality reduction comparing them with the algorithm original except the data *WiFi* which only contains 11 features. This shows it is necessary to implement dimensionality reduction while suffering high dimensional data.

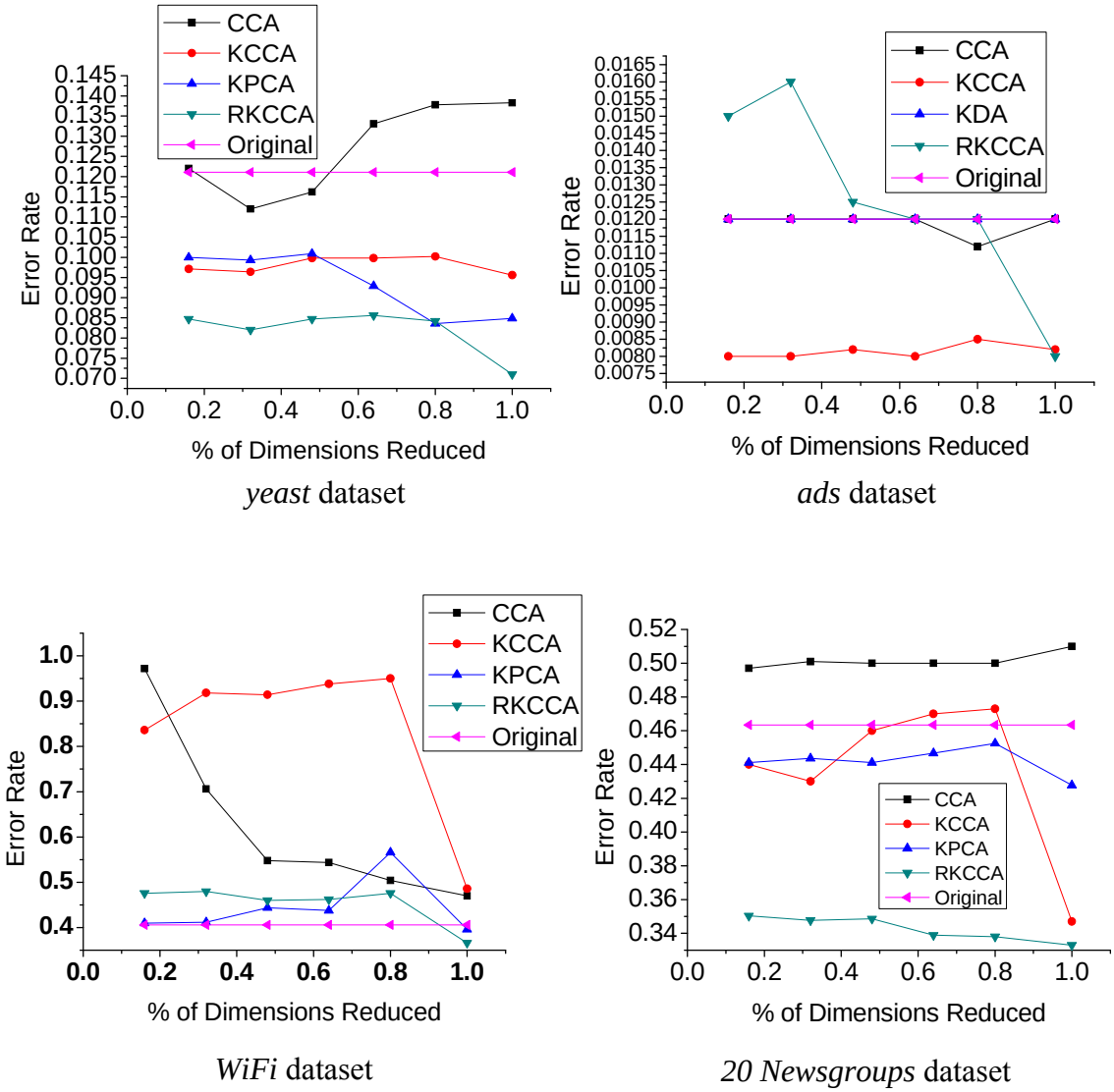

Figure 5.1 Classification Error after Dimensionality Reduction for data set yeast, ads, WiFi, and 20 Newsgroups respectively

Chapter 6

Conclusion

In this thesis, we have reviewed the existing techniques on dimensionality reduction. During the review process, we analyzed the pros and cons of the existing techniques on dimensionality reduction. Then we proposed a correlation analysis algorithm named RKCCA for dimensionality reduction. In the proposed algorithm, we projected two original vectors into RKHS in which to implement dimensionality reduction with KCCA measure is composed into two order steps, i.e., PCA followed by CCA in RKHS. Finally, the experimental results show that RKCCA is better than spectrum KCCA or the others algorithms in terms of classification accuracy and its effectiveness in dimensionality reduction. In summary, we have theoretical proved that the proposed RKCCA algorithm is equivalent to the spectrum KCCA algorithm, i.e., RKCCA = spectrum KCCA, in Chapter 4, and that the proposed RKCCA algorithm can be decomposed into two

orderly processes, i.e., PCA and CCA respectively in RKHS. Furthermore, we have shown in our experiments that RKCCA algorithm outperforms the traditional spectrum KCCA.

In this thesis, we have fixed a polynomial kernel (can be any positive semi-definite kernel) or their combination as the kernel function to learning the kernel matrix. Such kernel matrix may not be suitable for real world applications. In our future work, we plan to learn a kernel matrix from the training data rather than from a fixed kernel function.

Bibliography

[1] N. D. Lawrence (2008) Dimensionality Reduction the Probabilistic Way, *ICML2008 Tutorial*.

[2] K. M. Carter, R. Raich, and A. O. Hero III, (2009) An Information Geometric Approach To Supervised Dimensionality Reduction, *in Proceeding ICASSP2009*.

[3] A. Andoni, P. Indyk, and M. Patrascu (2006) On the Optimality of the Dimensionality Reduction Method, *FOCS2006*, 449-458.

[4] H. Peng, F. Long, and C. Ding (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on PAMI*, 27(8): 1226-1238.

[5] B. Krishnapuram, et al. (2004) A Bayesian approach to joint feature selection and classifier design. *IEEE Transactions on PAMI*, 6(9):1105-1111.

[6] F. Li, J. Yang and J. Wang, (2007) A Transductive Framework of Distance Metric Learning by Spectral Dimensionality Reduction, *ICML2007*.

[7] S.Y. Song et al. (2008) A unified framework for semi-supervised dimensionality reduction, *Pattern Recognition*, 2008.

[8] C. Hou, et al., (2009) Stable Local Dimensionality Reduction Approaches, *Pattern Recognition*.

[9] H.C. Law (2006) Clustering, Dimensionality Reduction, and Side Information, *PhD thesis in Michigan State University*.

[10] L.J.P. van der Maaten (2007) An Introduction to Dimensionality Reduction Using Matlab. *Technical Report MICC 07-07. Maastricht University, Maastricht, The Netherlands*.

[11] L.J.P. van et al. (2008) Dimensionality Reduction: A Comparative Review. *Neurocomputing*.

[12] S. Xing et al. (2007) Nonlinear Dimensionality Reduction with Local Spline Embedding, *IEEE TKDE*.

[13] C. Zhang et al.(2008) Nonlinear dimensionality reduction with relative distance comparison, *Neurocomputing*.

[14] J. B. Tenenbaum, V. de Silva and J. C. Langford (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science* 290 (5500): 2319-2323.

[15] H. Zou and T. Hastie (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Methodological),* 67(2): 301–320.

[16] K.Q. Weinberger et al. (2004) Learning a kernel matrix for nonlinear dimensionality reduction. *ICML2004*.

[17] T. Li, S. Ma, and M. Ogihara (2004) Document clustering via adaptive subspace iteration. *Proc. conf. Research and development in IR (SIRGIR),* 218–225.

[18] C. Ding, T. Li (2007) Adaptive Dimension Reduction Using Discriminant Analysis and K-means Clustering, *ICML2007*.

[19] H. Cevikalp, et al. (2008) Margin-Based Discriminant Dimensionality Reduction for Visual Recognition, *CVPR2008*.

[20] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov (2004) Neighborhood component analysis, in *Neural Information Processing Systems*, 17: 513–520.

[21] R. Raich, J. A. Costa, and A. O. Hero (2006) On dimensionality reduction for classification and its applications, in Proc. IEEE Intl. Conference on Acoustic Speech and Signal Processing.

[22] S.A. Orlitsky (2005) Supervised dimensionality reduction Using Mixture Models, *ICML 2005*.

[23] I.Rish, G. Grabarnik, and G. Cecchi (2008) Closed-Form Supervised Dimensionality Reduction with GLMS, *ICML2008*.

[24] D. Zhang, et al. (2007) Semi-Supervised Dimensionality Reduction, *SDM07*.

[25] A. Bar-Hillel, et al. (2005) Learning a Mahalanobis metric from equivalence constraints, *Journal of Machine Learning Research*, 6:937–965.

[26] W. Tang and S. Zhong (2006) Pairwise constraints-guided dimensionality reduction, in *SDM'06 Workshop on Feature Selection for Data Mining*.

[27] W. Yang, et al. (2008) A Graph Based Subspace Semi-supervised Learning Framework for Dimensionality Reduction, *ECCV2008*.

[28] B. Zhang, et al. (2008) Semi-supervised dimensionality reduction for image retrieval, *Visual Communications and Image Processing*.

[29] X. Yang, et al. (2006) Semi-Supervised Nonlinear Dimensionality Reduction, *ICML06*.

[30] K.Q., Weinberger and L.K Saul (2006) An Introduction to Nonlinear Dimensionality Reduction by Maximum Variance Unfolding, *AAAI2006*.

[31]L. Song, et el. (2007) Colored maximum variance unfolding, *NIPS2007*.

[32] D.P. Foster, et al. (2009) Multi-View Dimensionality Reduction, via Canonical Correlation Analysis, *ICML2009*.

[33] J. Jiang (2008) A Literature Survey on Domain Adaptation of Statistical Classifiers, *Technical Report in UIUC*.

[34] S.J. Pan and Q. Yang (2008) A Survey on Transfer Learning, *Technical Report in HKUST*.

[35] L. Torrey and J. Shavlik (2009) Transfer Learning, Handbook of Research on Machine Learning Applications, *IGI Global 2009*.

[36] Z. Wang, Y. Song and C. Zhang (2008) Transferred Dimensionality Reduction, *ECML2008*.

[37] S.J. Pan et al., (2008) Transfer Learning via Dimensionality Reduction, *AAAI2008*.

[38] K.M. Borgwardt, et al. (2006) Integrating Structured Biological Data by Kernel Maximum Mean Discrepancy, *Bioinformatics*, 22:49-57.

[39] A. Gifi (1990) Nonlinear multivariate analysis, *New York, Wiley*.

[40] D.R. Hardoon, et al. (2004) Canonical Correlation Analysis: An Overview with Application to Learning Methods, *Neural Computation*, 16: 2639-2664.

[41] H. Hotelling (1936) Relations between two sets of variates, *Biometrika*, 28: 321-377.

[42] M. Loog, et al. (2005) Dimensionality reduction of image features using the canonical contextual correlation projection, *Pattern Recognition*, 38:2409-2418.

[43] E. Kidron, et al. (2005) Pixels that Sound, *IEEE CVPR2005*, 1:88-95.

[44] T.K. Sun and S.C. Chen (2007) Locality Preserving CCA with Applications to Data Visualization and Pose Estimation, *Image and Vision Computing*, **25**: 531-43.

[45] J. Pan, et al. (2005) Accurate and Low-cost Location Estimation Using Kernels. *IJCAI2005*.

[46] S.Y. Huang, et al., (2007) Nonlinear Measures of association with KCCA and applications, *Journal of Statistical Planning and Inference*, 2007.

[47] J. Dauxois and G.M. Nkiet (1998) Nonlinear canonical analysis and independence test, *Ann. Statist.,* 26:1254-1278.

[48] P.L. Lai (2000) Neural implementations of canonical correlation analysis, *Ph.D thesis, Dept. of computing and information systems, University of Paisley, Scotland*.

[49] Z. Gou, and C. Fyfe (2004) A canonical correlation neural network for multicollinearity and functional data, *Neural Networks*, 17:285–293.

[50] A. Gretton, et al. (2005) kernel methods for measuring independence, *JMLR* 2005.

[51] B. Scholkopf and A. Smola (2001) Learning with Kernels, *MIT Press*.

[52] M. Kuss and T. Graepel (2003) The Geometry Of Kernel Canonical Correlation Analysis, *Technical Report in Max Planck Institute for Biological Cybernetics*.

[53] M. Yamada, et al. (2005) Relation between kernel CCA and Kernel FDA, *IJCNN2005*.

[54] K. Fukumizu, et al. (2007) Statistical Consistency of Kernel Canonical Correlation Analysis, *Journal of Machine Learning Research*, 8:361-383.

[55] Y. Yamanishi, et al. (2003) Extraction of Correlated Gene Clusters from Multiple genomic Data by Generalized KCCA, *Bioinformatics*, 19:1323-1330.

[56] M.B. Blaschko and C.H. Lampert (2008) Correlational spectral clustering, *CVPR2008*.

[57] T.K. Sun, et al. (2008) A Supervised Combined Feature Extraction Method for Recognition. *ICDM 2008*.

[58] M. Sham and P. Dean (2007) Multi-View Regression via Canonical Correlation Analysis, *COLT 2007.*

[59]K. Livescu et al. (2008) Multi-View Clustering via Canonical Correlation Analysis, *NIPS2008*.

[60] Z. Zhou et al. (2007) Semi-supervised learning with very few labeled training examples, *In: Proceedings of AAAI2007*, 675-680.

[61]M.B. Blaschko, et al., (2008) Semi-Supervised Laplacian Regularization of Kernel Canonical Correlation Analysis, *ECML2008*.

[62] K. Fukumizu et al. (2007) Kernel Measures of Conditional Dependence. *In: NIPS2007*.

[63] H. Suetani, et al. (2006) Detecting hidden synchronization of chaotic dynamical systems: A kernel-based approach. *Journal of Physics A: Mathematical and General*, 39:10723–10742.

[64] Y. Li, and J. Shawe-Taylor (2006) Using KCCA for Japanese-English cross-language information retrieval and document classification. *Journal of Intelligent Information Systems*. 27:117-133.

[65] B. Fortuna and J. Shawe-Taylor (2005) The use of machine translation tools for cross-lingual text mining, *ICML2005*.

[66] F.R. Bach and M.I. Jordan (2002) Kernel Independent Component Analysis. *JMLR* 3: 1-48.

[67] K. Fukumizu et al. (2009) Kernel dimension reduction in regression. *Annals of Statistics*.

[68] E.M. Jordaan (2002) Development of Robust Inferential Sensors: Industrial Application of Support Vector Machines for Regression, *EUT, Netherlands PhD thesis*.

[69] S. Zheng, J. Liu, and J. Tian (2005) An efficient star acquisition method based on SVM with mixtures of kernels, *Pattern Recognition Letters,* 26: 147–165.

[70] K. Fang, et al. (2000) Uniform design: Theory and applications, *Technometrics*, 42(3): 237-248.

[71] K. Fang and D. Lin (2003) Uniform experimental designs and their application in industry. *Handbook of Statistics*, 22:131-170.

[72] C.L. Blake and C.J. Merz (1998) *UCI Repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science.

[73] L. Sun et al. (2009) On the equivalence between canonical correlation analysis and orthonormalized partial least squares, *IJCAI2009*.

[74] S. Akaho (2001) A kernel method for canonical correlation analysis. *In Proceedings of International Meeting on Psychometric Society*.

[75] A.J. Cannon and W.W. Hsieh (2008) Robust nonlinear canonical correlation analysis: application to seasonal climate forecasting. *Nonlinear Processes in Geophysics*, 12: 221-232.

[76] T. Sun and S. Chen (2007) Class label versus sample label-based CCA, Applied Mathematics and computation, 185:272-283.

[77] Q. Wang and J. Li (2009) Combining local and global information for nonlinear dimensionality reduction, *Neurocomputing*.