Dimensionality reduction by Mixed Kernel Canonical Correlation Analysis

Xiaofeng Zhu, Zi Huang, Heng Tao Shen, Jian Cheng, Changsheng Xu

Abstract

In this paper, we propose a novel method named Mixed Kernel CCA (MKCCA) to achieve easy yet accurate implementation of dimensionality reduction. MKCCA consists of two major steps. First, the high dimensional data space is mapped into the reproducing kernel Hilbert space (RKHS) rather than the Hilbert space, with a mixture of kernels, i.e. a linear combination between a local kernel and a global kernel. Meanwhile, a uniform design for experiments with mixtures is also introduced for model selection. Second, in the new RKHS, Kernel CCA is further improved by performing Principal Component Analysis (PCA) followed by CCA for effective dimensionality reduction. We prove that MKCCA can actually be decomposed into two separate components, i.e. PCA and CCA, which can be used to better remove noises and tackle the issue of trivial learning existing in CCA or traditional Kernel CCA. After this, the proposed MKCCA can be implemented in multiple types of learning, such as multi-view learning, supervised learning, semi-supervised learning, and transfer learning, with the reduced data. We show its superiority over existing methods in different types of learning by extensive experimental results.

Keywords: Dimensionality reduction, Mixed kernel, Canonical Correlation Analysis, Model selection

1. Introduction

Recent applications, such as text categorization, computer vision, image retrieval, microarray technology and visual recognition, all involve high dimensional data [41,59,49,22,35,43]. In practice, although high dimensional data can be analyzed with high-performance contemporary computers, several problems still occur while dealing with high dimensional data. First, high dimensional data lead to an explosion in execution time, which is called “curse of dimensionality”. Second, some attributes in the datasets are often “noises” or irrelevant to the learning from data, and thus do not contribute to (sometimes even degrade) the learning process. Last but not least, the number of the “intrinsic” dimensions in high dimensional datasets is typically low [51,34,33]. Hence, designing efficient and effective solutions to deal with high dimensional data is both interesting and challenging.

Dimensionality reduction, aiming to reduce the number of dimensions (or attributes) of high dimensional data, is regarded as the primary way to understand the data in the high dimensional space for various applications.

Many frameworks [28,20,57] or survey papers [27,48] on dimensionality reduction have been presented. Most existing dimensionality reduction methods are only designed for one particular type of learning. For example, PCA is mainly used for unsupervised learning, and linear discriminant analysis (LDA) is designed for supervised learning. In this paper, we propose a new dimensionality reduction method named Mixed Kernel CCA (MKCCA) which employs CCA [19] in the reproducing kernel Hilbert space (RKHS) with a mixture of kernels model to implement dimensionality reduction. In the RKHS, we implement dimensionality reduction by two sequential processes, i.e. PCA followed by CCA. MKCCA is a generalized method for multiple types of learning. That is, the reduced data can be used for effective multi-view learning, supervised learning, semi-supervised learning and transfer learning. Moreover, the proposed MKCCA method is easy to be implemented with less parameters. Experimental results on real-life datasets show that MKCCA is more accurate than existing methods corresponding to different types of learning.

The main contributions of this paper include:

- We propose an effective and efficient new dimensionality reduction method called MKCCA. Different from traditional...
Kernel CCA methods, MKCCA utilizes a mixture of kernels (rather than a single kernel) to map original data into the high dimensional space (i.e. the RKHS rather than the Hilbert space). This mapping is more beneficial for building a theoretical framework for different types of learning. In theory, we prove that there is a one-to-one linear transformation between the mapping in the new RKHS with a mixture of kernels and the mapping in the Hilbert space for implementing CCA. In implementation, we map the original data into a “small” but sufficient space (i.e. RKHS) to capture the phenomena of interest [39]. In the RKHS, dimensionality reduction is performed in two sequential processes, i.e. PCA followed by CCA, to better remove noises and handle the issue of trivial learning. Due to introducing the mixture of kernels into the RKHS, the proposed MKCCA method can achieve both interpolation ability and extrapolation ability during the learning process. Furthermore, our method is easy to be implemented with less parameters.

- We introduce a new model selection method which can reduce the time complexity and achieve minimum-discrepancy comparing to the traditional model selection methods, such as exhaustive grid search method, cross-validation method and uniform design method.
- We discuss the issue that one dimensionality reduction method can be applied for multiple types of learning. This can be broadly applied in the real applications. Although CCA method (or Kernel CCA method) can also be applied for implementing dimensionality reduction in different types of learning, to the best of our knowledge, no literature has discussed such an advantage.
- Experimental results confirm the efficiency and effectiveness of our method for different types of learning in the extensive performance study.

In the rest of the paper, we review related work on dimensionality reduction in Section 2, and give a basic introduction on CCA and Kernel CCA methods in Section 3. In Section 4, we present our MKCCA method. Experimental evaluation is reported and discussed in Section 5. We conclude the paper with future work in Section 6.

## 2. Related work

Existing dimensionality reduction methods can be partitioned into different categories according to different perspectives. For example, there are linear methods and nonlinear methods according to the relationships between condition attributes and decision attributes (i.e. class labels) [20], feature selection methods, feature extraction methods, and feature grouping methods respectively [26,48] according to the means by which low dimensional data are formed, and global dimensionality reduction and local dimensionality reduction in the domain of similarity search [41,30]. Here a brief introduction on existing dimensionality reduction methods is described according to all types of learning, such as supervised learning, unsupervised learning, semi-supervised learning, multi-view learning, and transfer learning respectively.

Unsupervised methods perform dimensionality reduction with only the condition attributes without considering the information on class labels. For example, Sanguinetti [38] proposed a latent variable model to perform dimensionality reduction in image datasets. The method in [53] uniquely preserves the feature of global coordinate by a compatible mapping. Among traditional unsupervised methods, such as, PCA [36], independent component analysis (ICA), locally linear embedding (LLE) and random projection [24], random projection method is the promising one as it is not as computationally expensive as the others [48]. PCA is the most popular one in the domain of both machine learning and data mining [51,47,16]. Recently unsupervised dimensionality reduction methods have been carried out as a pre-processing step to select the subspace dimensions before the clustering process. For example, the adaptive technique in [29] adjusts the subspace adaptively to form clusters which are best separated or well defined. And the method in [8] preserves the separability by using the weighted displacement vectors.

Supervised methods on dimensionality reduction are designed to find a low dimensional transformation by considering class labels. In fact, class labels can be used with the condition attributes to extract relevant attributes. For example, discriminant analysis methods [40] can find the effective projection directions by maximizing the ratio of between-class variance to within-class variance. Recently supervised dimensionality reduction methods aim to minimize the loss on before and after the process of dimensionality reduction [37]. This loss can be measured in terms of cost function, degree of discrepancy, degree of dependence, class information distance, k nearest neighbor classification error, penalty function, or partial least squares (PLS) [56]. For example, Zeng and Trussel [56] introduced a novel penalty function to implement dimensionality reduction by allowing a trade-off between dimensionality number and system performance.

Semi-supervised methods on dimensionality reduction perform dimensionality reduction by combining labeled data with unlabeled data. In practical applications, unlabeled data are readily available but labeled data are more expensive to be obtained. Hence, existing semi-supervised dimensionality reduction methods are more practical than supervised methods or unsupervised methods. The framework of existing semi-supervised methods is usually built by combining the unsupervised framework with prior information, including class label [55], pairwise constraints [2] and side information [44]. Recently, some semi-supervised dimensionality reduction methods are constructed according to the framework of supervised learning. For example, Song et al. [45] built a semi-supervised framework by adding a regularization term to the original LDA. The new semi-supervised framework makes some classical methods, such as PCA, maximum margin criterion, locality preserving projections and their corresponding kernel versions, as its special cases.

All aforementioned three types of dimensionality reduction methods (i.e. unsupervised methods, supervised methods, and semi-supervised methods) are designed for dealing with the data in one dataset. Thus they are regarded as one-modal methods. Given the limited information in one dataset, outer sources (i.e. coming from the other databases) can be employed to strength the ability of dimensionality reduction. We call such a technique for dimensionality reduction as a multi-modal method. For instance, multi-view learning methods [16,52,54] use one of the views as target source (i.e. the dataset we need to implement dimensionality reduction) and other views as outer sources (i.e. the datasets which are employed to strength the performance of dimensionality reduction) to implement dimensionality reduction. For example, Foster et al. [13] employed CCA method for dimensionality reduction in multi-view learning. Due to implementing dimensionality reduction without class labels, multi-view methods are usually categorized into the unsupervised learning method.

---

2 Multi-view learning means there are multiple views (feature spaces) and one feature for class labels in one dataset. Each view can correctly separate the class labels without the help from the other views. All views are with same data distribution.
Some existing multi-modal dimensionality reduction methods, in which the outer sources are with different data distributions to the target source, are called as transfer learning based dimensionality reduction methods. Transfer learning [32,42] is to learn a new task (i.e. target source) through the transfer of knowledge from a related task (i.e. the outer source), which has already been learned or easily to be learned. Intuitively, transfer learning model is more practical and general than the aforementioned four models, such as, unsupervised learning model, supervised learning model, semi-supervised learning model and multi-view learning model. Both the method in [31] and the method in [50] are transfer learning based dimensionality reduction methods. The method in [50] modifies the LDA method (i.e. a supervised learning model) into a transfer model by summing the basic information (the information of condition attributes in two datasets) and prior information (class label in target dataset). Whereas the method in [31] combines the basic (i.e. source) information with prior information into high dimensional spaces by a learnt kernel trick, then performs dimensionality reduction in an unsupervised learning model. To the best of our knowledge, no literature has been focused on implementing transfer learning based dimensionality reduction by employing CCA-based methods. The proposed MKCCA method is focused on the topic, and can also be applied for other types of dimensionality reduction, including multi-view learning, supervised learning and semi-supervised learning in this paper.

3. Preliminary work

Table 1 lists some important notations used in this paper.

3.1. Basic theory on CCA and KCCA

Assuming two random variables: \( X^{(1)} \in \Omega^1 \) and \( X^{(2)} \in \Omega^2 \), we consider the relationship between \( X^{(1)} \) and \( X^{(2)} \) by choosing appropriate directions \( W_c^{(1)} \) and \( W_c^{(2)} \) as

\[
\rho = \max_{W_c^{(1)},W_c^{(2)}} \text{corr}(W_c^{(1)T}TX^{(1)}, W_c^{(2)T}TX^{(2)})
\]

Denoting the covariance matrix between \( X^{(1)} \) and \( X^{(2)} \) as

\[
\Sigma = \mathbb{E} \left[ \begin{pmatrix} X^{(1)} \\ X^{(2)} \end{pmatrix} \begin{pmatrix} X^{(1)} \\ X^{(2)} \end{pmatrix}^T \right] = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}
\]

Then the maximal canonical correlation between \( X^{(1)} \) and \( X^{(2)} \) can be changed into

\[
\rho = \max_{W_c^{(1)},W_c^{(2)}} \frac{W_c^{(1)T}\Sigma_{12}W_c^{(2)}}{\sqrt{W_c^{(1)T}\Sigma_{11}W_c^{(1)}W_c^{(2)T}\Sigma_{22}W_c^{(2)}}}
\]

Assuming \( \Sigma_{22} \) is invertible, the optimization problem in the corresponding Lagrangian of Eq. (3) is transferred into

\[
\begin{pmatrix} X^{(2)}X^{(1)T} \\ X^{(2)}X^{(1)T} \end{pmatrix} \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}^{-1} \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}^{-1} \begin{pmatrix} X^{(2)}X^{(1)T} \\ X^{(2)}X^{(1)T} \end{pmatrix}
\]

Finally, the problem in Eq. (4) is represented as

\[
\begin{align*}
W^{(2)} & = \frac{1}{2} \Sigma_{22}^{-1} \Sigma_{21} W^{(1)} \\
\Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} W^{(1)} & = \lambda \Sigma_{11} \Sigma_{21} W^{(1)}
\end{align*}
\]

We firstly solve the first equation in Eq. (5) to obtain \( W^{(1)} \), then plug the result into the first term of Eq. (5) for receiving \( W^{(2)} \).

Although we can obtain the optimization result of \( \rho \) (i.e. the correlation coefficient) by solving the eigenproblem in Eq. (5), it is quite difficult for CCA to extract useful representations of the data in real applications when the original data do not follow Gaussian distribution or are not linearly distributed.

Hence, CCA is further extended into nonlinear CCA in which the relationship between two variables can be dealt with by nonlinear relationship. Existing methods include neural network method [25] and Kernel CCA [18]. Since neural network method often suffers from some intrinsic problems such as long-time training, slow convergence and local minima [25], we focus on Kernel CCA in this paper.

Basically, Kernel CCA (KCCA) maps the data into the high dimensional space (a.k.a., the Hilbert space) for linear separation. We review the traditional KCC method following the idea in [18].

Given two input \( X^{(1)} \in \Omega^1 \) and \( X^{(2)} \in \Omega^2 \) with sample size \( n \), KCCA method maps them into high (even infinite) dimensional Hilbert spaces \( \Omega^1 \) and \( \Omega^2 \) (\( p \geq p, \text{and} q \geq q \)), via implicit mappings \( \psi^{(1)}(x^{(1)}) \rightarrow \psi^{(1)}(x^{(1)}) = (\psi^{(1)}(x^{(1)}), \ldots, \psi^{(1)}(x^{(1)})) \)

and

\[
\psi^{(2)}(x^{(2)}) = (\psi^{(2)}(x^{(2)}), \ldots, \psi^{(2)}(x^{(2)}))
\]

Here \( \psi^{(i)}(X^{(i)}) \) (\( i = 1, 2 \)) is called as a kernel spectrum [51] for a certain positive definite kernel, i.e. the kernel function

\[
k(x, y) = \psi(x) \psi(y)^T
\]

where \( x \in X^{(1)} \) (or \( y \in X^{(2)} \)). Its corresponding kernel matrix is

\[
K_{ij} = K_{ij} = \psi(x^{(i)}) \psi(x^{(j)})^T
\]

Denoting the projection direction on \( X^{(i)} \) as \( W_{k}^{(i)} \), according to Eq. (3), the linear relationship between \( W_{k}^{(1)}TK_1 \) and \( W_{k}^{(2)}TK_2 \) (i.e. the nonlinear relationship between \( x^{(1)} \) and \( x^{(2)} \)) can be substituted as

\[
\rho = \max_{W_{k}^{(1)},W_{k}^{(2)}} \frac{W_{k}^{(1)T}TK_1K_{ij}^{(2)}W_{k}^{(2)}}{\sqrt{W_{k}^{(1)T}TK_1K_{ij}^{(2)}W_{k}^{(1)}}}\sqrt{W_{k}^{(2)T}TK_1K_{ij}^{(2)}W_{k}^{(2)}}
\]

Assuming \( K_{ij}K_{ij}^{(2)} \) is invertible, we can receive

\[
\begin{pmatrix} K_{ij}K_{ij}^{(2)} \\ K_{ij}K_{ij}^{(2)} \end{pmatrix} \begin{pmatrix} W_{k}^{(1)} \\ W_{k}^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} W_{k}^{(1)} \\ W_{k}^{(2)} \end{pmatrix}
\]

Finally, the problem in Eq. (10) is transferred into

\[
\begin{align*}
W_{k}^{(2)} & = \frac{1}{2} K_{ij}K_{ij}^{(2)} W_{k}^{(1)} \\
K_{ij}^{-1}W_{k}^{(1)} & = \lambda K_{ij}^{-1}W_{k}^{(2)}
\end{align*}
\]

We firstly solve the first equation in Eq. (12) to obtain \( W_{k}^{(1)} \), then get \( W_{k}^{(2)} \). Both Eqs. (3) and (9) belong to a generalized eigenproblem with the form \( AX = \lambda BX \).

However, if \( K_{ij} \) or \( K_{ij}^{(2)} \) (or \( \Sigma_{11} \) or \( \Sigma_{22} \)) is invertible, the learning by both Eqs. (3) and (10) will be trivial. A trivial learning can cause numerical instability and computational difficulty. The optimization problem in these two equations is then ill-conditioned. To force a nontrivial learning on the correlation the regularization process should be introduced to control the flexibility of the projection mappings [18,21]. For example, Hardoon et al. [18] regularized the optimization problem by partial least squares (or ridge-style regression methods) methods to penalize the norms of the associated weights for avoiding overfitting and ill-conditioned. Hardoon et al.
[18] also dealt with it by Gram–Schmidt orthogonalization method or incomplete Cholesky decomposition method. Gretton et al. [17] stabilized the numerical computation for solving the regularization process by adding a small quantity to the diagonals. Huang et al. [21] proposed a random basis subset method to deal with this issue.

In our observation, CCA does not provide matched subspaces in the following two cases, i.e. the case of CCA without regularization, or the case of CCA without linear dependence between samples. The proposed MKCCA method (i.e. PCA followed by CCA) essentially induces linear dependence, then makes an unregularized CCA step for nontrivial learning. Hence, our MKCCA method does not need to handle the regularization process.

In the left part of this subsection, we give a note which will benefit for the theoretical proof of the proposed MKCCA method in the following section. That is, in the traditional KCCA method, the kernel matrix $K_i$ in Eq. (10) can be represented in terms of its descending order of eigenvectors $v_a$ and eigenvalues $\lambda_a$ as

$$K_i = \sum_{a=1}^{m} \lambda_a v_a^T v_a$$  \hspace{1cm} (13)

It is regarded as a spectrum decomposition of $K_i$, thus we call the traditional CCA method (e.g. [18,21]) as the spectrum CCA method throughout this paper.

### 3.2. Feasibility analysis

In this subsection, we explain the following issues: (1) Why to select CCA for multiple types of learning. (2) Why to modify the existing KCCA method. The way to modify the traditional KCCA method and the advantages of the MKCCA method over the spectrum KCCA method will be presented in the next section.

As one of the multi-modal methods, CCA can build more effective learning by involving outer useful information (one modal is called target source, outer sources otherwise). If the outer sources contain same data distribution and share the same class labels with the target source, the type of learning belongs to a multi-view learning [10]. While both outer sources and the target source present different data distribution, CCA can be used in transfer learning. If class labels are regarded as a modal, CCA can be implemented in supervised learning. CCA can also be implemented in semi-supervised learning while considering prior information as the outer source or another model [2,58]. Therefore, it is feasible for utilizing CCA for multiple types of learning.

We have previously explained why CCA is replaced by KCCA in real applications. However, traditional CCA methods still have some drawbacks to be improved. On the one hand, the traditional KCCA method should simultaneously set many parameters, such as precision parameter, regularization parameter and the others. Prior knowledge is often needed for correctly setting these parameters. A new method which can be easily operated by general users is still expected. On the other hand, KCCA maps the original data into the high dimensional space by an implicit function. That is hard for theoretical development [21] as we usually need to know its concrete presentation for theoretical proof. Hence, it is necessary to extend the traditional KCCA method with clear theoretical foundations.

In this paper, we propose a new dimensionality reduction method named MKCCA which can overcome aforementioned limitations of the traditional KCCA.

### 4. MKCCA

#### 4.1. The general idea

There are two key steps in the proposed MKCCA method. First, MKCCA maps the original data into a high (or even infinite) dimensional space (details in Section 4.2), Second, MKCCA implements dimensionality reduction by two sequential processes (i.e. PCA followed by CCA) in the mapped space (its theory is presented in Section 4.3). After these two key steps, MKCCA can be used to implement all kinds of learning assignments or scenarios (such as classification, regression or others) in the reduced data space for different types of learning, such as multi-view learning, supervised learning, semi-supervised learning and transfer learning.

In the first step, MKCCA projects the original data into the RKHS of continuous value function by an explicit positive definite mixture of kernels to replace the spectrum KCCA method which projects the original data into the Hilbert space of real value function by an implicit positive single kernel function. The new RKHS is “smaller” than the Hilbert spaces of smooth functions but sufficient to capture interesting phenomena. Moreover, the new mapping is beneficial for constructing a theoretical framework by the explicit mapping function. Due to employing a mixture of kernels (details in Section 4.4.1), the new RKHS contains a larger hypothesis space than the traditional RKHS such that MKCA can simultaneously receive interpolation ability and extrapolation ability during the learning process. Furthermore, the new mapping in the proposed MKCCA method is proved to be a linear transformation from the mapping in the Hilbert space for implementing the CCA.

In the second step, to reduce the parameter setting burden for users and effectively implement dimensionality reduction in the new RKHS, we prove that the dimensionality reduction in our MKCCA method can be further decomposed into two processes, i.e. PCA followed by CCA. Then the proposed MKCCA method induces a linear dependence among variables, and makes an unregularized CCA step for nontrivial learning. The PCA process is more effective for removing noises. Thus the MKCCA method is also more efficient and effective.

Besides above these, in the first step, we propose a new model selection method named uniform design for experiments with mixtures (UDEM) method, for efficient implementation of dimensionality reduction in the new RKHS. The complexity of UDEM method is usually linear to the levels per parameter, i.e. $O(k)$ ($k$ is the number of parameters’ level) in the mixture of kernels model. However, the popular methods, such as cross-validation method, are typically at least $O(k^i)$ ($i$ is the number of parameters) in the mixture of kernels model or $O(k)$ in the single kernel model.

#### 4.2. Mapping input into the new RKHS

Following the literature [39], we firstly explain how MKCCA maps the original data into the new RKHS by an explicit projection.

Given a positive definite mixture of kernels function (details in Section 4.4.1) and a centered variable $X$ (i.e. zero-mean and univariate), we firstly define an explicit mapping $\Phi : X \rightarrow \Phi(X) = k(x, X)$

where $k(x, X) = k(x_1, X_1), \ldots, k(x_n, X_n)$, $x, x_i \in X$. Term $k(x, X)$ means a function of the expression “dot” which is called a literal in mathematics or logic.

Next, we construct an inner product space (denoted as $\langle \cdot, \cdot \rangle$) containing the input under $\Phi$ by two steps. In the first step, we form the vector space containing all linear combinations, i.e. for any $m, m' \in N$, $x, x_i' \in X$, and $x_i, x_i' \in R$

$$f() = \sum_{i=1}^{m} \alpha_i k(x_i, X), \hspace{1cm} g() = \sum_{j=1}^{m'} \beta_j k(x_i', X')$$

(15)
In the second step, we define an inner product between \( f(x) \) and \( g(x) \):

\[
\langle f(x), g(x) \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i \beta_j k(x_i \cdot x_j)
\]

(16)

This inner product can be proved satisfying the symmetry, bilinearity and positive definiteness conditions [39]. So such an inner product space under a Hilbert space based on the aforementioned steps is called a reproducing kernel Hilbert space (RKHS), and has the following properties:

\[
\langle f(x), g(x) \rangle = \| f \|^2 = \sum_{i=1}^{m} x_i \beta_i k(x_i \cdot x_i)
\]

(17)

\[
\langle k(x, x), k(x, x) \rangle = k(x, x) \quad \text{or} \quad f(x) = \langle f, k(x) \rangle
\]

(18)

where \( k(x, x) = k(x_1, x_1), \ldots, k(x_m, x_m) \in X, x \in X \), and \( \| \cdot \|^2 \) means the Euclidean norm. Term \( k(x, x) \) means a function of the expression “dot” which is called a literal in mathematics or logic. The kernel \( k(x, x) \) is called the reproducing kernel satisfying the reproducing property in Eqs. (17) and (18).

Comparing the Hilbert space to the new RKHS: (1) the feature space in the RKHS is constructed by continuous functions, i.e. the RKHS is filled with a set of linear and bound continuous functions, whereas a Hilbert space is filled with real points. Therefore, we can informally think that the RKHS is “smaller” than the Hilbert space; (2) the mapping function in the new RKHS is explicitly represented (i.e. \( \Phi(x) = k(x, x) \)), whereas it is implicit in the Hilbert space.

Different from the traditional kernel methods with a single kernel function, the MKCCA method replaces the single kernel function with a mixture of kernels function in the new RKHS. The proposed method with mixtures of kernels can simultaneously provide interolation ability and extrapolation ability in a larger hypothesis space (deferred in Section 4.4.1).

4.3. Reducing dimensionality in the new RKHS

After showing that it is feasible for mapping the original data into the new RKHS, we prove that the above mapping is unique. Then we show that there is a one-to-one linear transformation between the mapping in the new RKHS and the one in the Hilbert space for performing CCA. With this, we proceed to prove that Kernel CCA in the new RKHS can be decomposed into two sequential steps: PCA and CCA in the new RKHS. Thus dimensionality reduction by the MKCCA method in the new RKHS can be implemented with two sequential processes, i.e. PCA followed by CCA in the new RKHS.

According to the proof in [39], given a Mercer kernel \( k \), there exists an RKHS \( H \), such that \( x \rightarrow \Phi(x) = k(x, x) \), where \( \langle \Phi(x), \Phi(x') \rangle = k(x, x') \in X, x \in X \), and the reproducing kernel \( k(x, x) \) is uniquely determined by the space \( H \).

Thus any data can be mapped into a smooth space by kernel functions in the RKHS. Hence, it is feasible to map the input data into an RKHS.

After projecting the input into the RKHS, we proceed to prove there is a one-to-one linear transformation between the new mapping in the RKHS and the one in the Hilbert space for implementing CCA. This is achieved by showing that the isomorphic characteristic in the Hilbert space is preserved in the new RKHS.

**Theorem 1.** There exists a one-to-one linear transformation between the mapping function \( \Psi(x) \) in the Hilbert space and the one \( \Phi(x) \) in the new RKHS for implementing the CCA.

**Proof.** We first prove “\( \Phi(x) \Rightarrow \Psi(x) \)”.

Based on the Mercer’s theorem, since the continuous positive definite kernel \( k(x, x) \) is symmetric and positive definite, it is orthogonally diagonalizable as in the case with finite dimensions. Thus \( k(x, x) \) can be represented as \( k(x, x) = \sum_{i=1}^{N} \lambda_i \Phi(x) \Phi(x)^T \) by its ordered eigenvectors series \( \Phi(x) \) and corresponding eigenvalues series \( \lambda_i \). According to [11], the nonlinear CCA can be approximated as \( k(x, x) = \sum_{i=1}^{N} \lambda_i \Phi(x) \Phi(x)^T \) (\( n \in N \)), in terms of uniform convergence of a certain underlying sequence. Hence, CCA in the new RKHS can be implemented as spectrum decompositions similar to Eq. (13).

Next, we prove “\( \Psi(x) \Rightarrow \Phi(x) \)”.

By combining Eq. (14) with both Eqs. (17) and (18), for any \( x \in X \), we have

\[
\| \Psi(x) \|^2 = k(x, x) = \langle k(x, x), k(x, x) \rangle = \| \Phi(x) \|^2
\]

where “\( \| \Psi(x) \|^2 \)” (or “\( \| \Phi(x) \|^2 \)” means a \( L_2 \) norm operator under the Hilbert space (or the RKHS).

**Theorem 1** shows CCA in the new RKHS can be linearly transferred into the spectrum CCA method in the Hilbert space, and vice versa.

**Theorem 2.** MKCCA (i.e. Kernel CCA in the new RKHS) can be decomposed into two components, i.e., PCA and CCA.

**Proof.** Given positive definite kernel functions \( k_1 \) and \( k_2 \), two centered variables \( X^{(1)} \in \Omega^1, X^{(2)} \in \Omega^2 \), and two mappings: \( \Phi : X^{(1)} \rightarrow \Phi(X^{(1)}) = k(x, x'), x, x' \in X \), in the new RKHS, after performing PCA in the new RKHS, we denote the original data \( X^{(1)} \) as \( X^{(1)} = \hat{W}^{(1)}_1 \Phi(X^{(1)}) \), where \( \hat{W}^{(1)}_1 \) is the projected directions of \( X^{(1)} \).

Then according to the reproducing property presented in Eq. (18), two variables \( X^{(1)} \) and \( X^{(2)} \) are represented as

\[
\begin{align*}
X^{(1)} & = \hat{W}^{(1)}_1 \Phi(X^{(1)}) \Phi(X^{(1)})^T \hat{W}^{(1)}_1 = \hat{W}^{(1)}_1 \hat{K}^{(1)}_1 \hat{W}^{(1)}_1 \\
X^{(2)} & = \hat{W}^{(2)}_2 \Phi(X^{(2)}) \Phi(X^{(2)})^T \hat{W}^{(2)}_2 = \hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \hat{W}^{(2)}_2
\end{align*}
\]

Plugging Eq. (20) into Eq. (4) and according to [16,47], we have

\[
\begin{align*}
&\lambda \left( \begin{array}{c}
\hat{W}^{(1)}_1 \\
\hat{W}^{(1)}_1 \hat{K}^{(1)}_1 \hat{W}^{(1)}_1 \\
\hat{W}^{(2)}_2 \\
\hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \hat{W}^{(2)}_2
\end{array} \right)
\end{align*}
\]

\[
\begin{align*}
&\lambda \left( \begin{array}{c}
\hat{W}^{(1)}_1 \\
\hat{W}^{(1)}_1 \hat{K}^{(1)}_1 \hat{W}^{(1)}_1 \\
\hat{W}^{(2)}_2 \\
\hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \hat{W}^{(2)}_2
\end{array} \right)
\end{align*}
\]

\[
\text{with}
\begin{align*}
&\lambda = \left( \begin{array}{c}
\hat{K}^{(1)}_1 \\
\hat{K}^{(1)}_1 \hat{K}^{(1)}_1 \\
\hat{K}^{(2)}_2 \\
\hat{K}^{(2)}_2 \hat{K}^{(2)}_2
\end{array} \right)
\end{align*}
\]

\[
\begin{align*}
&\lambda = \left( \begin{array}{c}
\hat{K}^{(1)}_1 \\
\hat{K}^{(1)}_1 \hat{K}^{(1)}_1 \\
\hat{K}^{(2)}_2 \\
\hat{K}^{(2)}_2 \hat{K}^{(2)}_2
\end{array} \right)
\end{align*}
\]

where

\[
\begin{align*}
&\lambda = \left( \begin{array}{c}
\hat{W}^{(1)}_1 \\
\hat{W}^{(1)}_1 \hat{K}^{(1)}_1 \\
\hat{W}^{(1)}_1 \\
\hat{W}^{(2)}_2 \\
\hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \\
\hat{W}^{(2)}_2 \\
\hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \\
\hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \\
\hat{W}^{(2)}_2 \hat{K}^{(2)}_2 \\
\end{array} \right)
\end{align*}
\]

According to **Theorem 1**, we know that there is a linear transformation between the mappings in the new RKHS and Hilbert space. Hence, we denote the projected direction of \( X^{(1)} \) in
the new RKHS as $W_{M}^{(i)}$, i.e.
\[
\begin{pmatrix}
K_{1}K_{2}^T \\
K_{2}K_{1}
\end{pmatrix}
\begin{pmatrix}
W_{1}^{(i)} \\
W_{2}^{(i)}
\end{pmatrix}
= \lambda
\begin{pmatrix}
K_{1}K_{1}^T X \\
K_{2}K_{2}^T X
\end{pmatrix}
\begin{pmatrix}
W_{1}^{(i)} \\
W_{2}^{(i)}
\end{pmatrix}
\] (22)

Denoting $W_{M}^{(i)}$ as
\[
W_{M}^{(i)} = \sum_{j=1}^{N} W_{j}^{(i)} C_{j} 
\] i = 1, 2
Then we can find that Eq. (22) is same as Eq. (21).

Theorem 2 shows that PCA can be performed before CCA in the new RKHS for correlation analysis. In our new MKCCA method, we regard the results of PCA as the input of CCA for more effective dimensionality reduction. The reasons behind are further explained in Section 5.4.

4.4. Choice of kernel and model selection

In this subsection, we discuss the issues of kernel choice and model selection, before we present the overall MKCCA algorithm.

4.4.1. Choice of kernel

To choose the kernel function in kernel methods is important because different kernels reveal different types of low dimensional structure [51]. Moreover, the learning quality for a test point is not only determined by its ability to learn from its neighborhoods (i.e. interpolation ability) but also its ability to predict or affect unseen data far way from itself (i.e. extrapolation ability). Jordaan [23] pointed out that the choice of the kernel function is usually determined by two factors, i.e. predefining the type of kernels and tuning the kernel parameters. There are two types of kernels, namely local kernels (e.g., Gaussian kernel) and global kernels (e.g., polynomial kernel). It is showed [23,33] that a local kernel can present good interpolation abilities, but fails to provide longer range extrapolation (i.e. extrapolation ability). A global kernel contains interpolation ability as well as extrapolation ability, but we cannot obtain them simultaneously.

The behavior of the two types of kernels is shown in Fig. 1. We can see that only the neighborhood of the test point has an influence on its kernel value because the kernel values in many points far way from the test point level off to zero on the local kernel. However, all points have an influence on the test point because they have nonzero kernel value in a global kernel. Moreover, the larger degree the polynomial function, the lager its kernel value is. That is, better extrapolation ability can be found at lower orders, while at higher order with better interpolation ability.

Hence, we can combine the good feature of the two kinds of kernels to simultaneously achieve interpolation ability and extrapolation ability, i.e. replacing single kernel (i.e. local kernel or global kernel) with a linear combination between a local kernel and a global kernel (this combination is called as mixture of kernels in [23]), i.e.
\[
k_{mix} = \omega k_{p} + (1-\omega)k_{g}
\] (24)

Here $k_{g} = \exp(-(x-x_{i})^2/2\sigma^2)$, and $k_{p} = (x_{i}+1)^q$ are a Gaussian kernel and a polynomial kernel respectively $q$ or $\sigma$ ($\sigma \in \mathbb{R}$ and $q \in \mathbb{N}$) are the corresponding width in kernel functions, the weight $\omega (\omega \in [0,1])$. Obviously, since both polynomial kernel and Gaussian kernel are positive definite kernel, their linear combination is also positive definite. So the mixture of kernels in Eq. (24) is a positive definite kernel. Moreover, traditional single kernel method is a special case of mixture of kernels. That is, mixture of kernels is a Gaussian kernel means while $\omega = 0$, and a polynomial kernel while $\omega = 1$.

Fig. 2 shows that the mixture of kernels model has not only a local effect, but also a global effect, while tuning the value of $\omega$.

Different from the single kernel model, the mixture of kernels model can simultaneously receive interpolation ability and extrapolation ability, whereas the single kernel model only presents one of them once. Moreover, the mixture of kernels model can potentially give a larger hypothesis space which tends to be more expressive than the single kernel model. That is, a mixture of kernels model can better approximate target functions of practical problems. Although the multiple Gaussian kernels model can approximate limit to any continuous functions, it should set multiple parameters as well as usually give very poor approximations for many target functions [4]. However, the mixture of kernels model only needs to set three parameters together, and we later propose a novel method named UDEM with lower complexity to efficiently set parameters.

Both the mixture of kernels and multiple kernel learning (MKL)\(^3\) [1,46] can give a large hypothesis space by employing a

---

\(^3\) Given $p$ kernel functions $k_{1},...,k_{p}$ that potentially fit for a given problem, the MKL is designed to search for a linear combination of these kernels (or part of these kernels) such that the derived kernel $k = \sum_{p} \omega_{p} k_{p}$ is optimal according to some a criteria.
linear combination of kernel functions. However, they are different. First, the mixture of kernels consists of two kernel functions (i.e., a global kernel function and a local kernel function), but the MKL can be composed by two or more than two kernel functions. Second, the mixture of kernels generates the kernel matrix by performing the convex optimization instead of predefining it. The MKL generates the optimal kernel matrix by performing the convex optimization instead of predefining it [46]. Hence the mixture of kernels model is easier and faster to compute the kernel values than the MKL. Third, the goal of the mixture of kernels is to obtain better extrapolation ability and interpolation ability by integrating the advantages of two different kernel functions. The MKL focuses on learning a better combination on the weights of kernel functions for achieving some goals, such as classification accuracy. In essence, the mixture of kernels model belongs to the domain of the single kernel model but with the form of multiple kernel learning. This is because each kernel function in the mixture of kernels is assigned an independent weight rather than achieving the weights by solving a considerable complex optimization problem in the MKL.

### 4.4.2. Model selection

The issue of choosing the optimal parameter setting for $q, \sigma$, and $\omega$ to achieve a better generalization performance is called model selection. Existing methods for model selection include grid search method, gradient-based method, cross-validation method, uniform design method, and the others [12]. However, exhaustive grid search method can not implement effectively due to high computational cost. In the domain of machine learning, cross-validation method is very popular, but its complexity is at least $O(k^l)$ ($k$ (or $i$) is the number of level per parameter), i.e., $O(k^3)$ for three parameters. Uniform design (UD) method [12] overcomes these drawbacks by finding uniformly scattered as well as good representative points over the parameters’ domain. That is, UD method uses less feature points to present minimal discrepancy rather than exhaustive searching the whole domain of the parameters, and its complexity is usually $O(k)$ for multiple parameters.

In the left part of this section, we briefly introduce the uniform design method, and propose our uniform design for experiments with mixtures method (referred to as UDEM method) for model selection in our MKCCA method.

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

Supposing there are $s$ parameters in a domain $\Omega^s$, and we want to choose a set of points $P_m = \{p_1, \ldots, p_m\} \subset \Omega^s$ which are uniformly scattered over the domain $\Omega^s$. Let $F(\theta)$ (or $F_m(\theta)$) be the cumulative uniform distribution function over $\Omega^s$ (or the empirical cumulative distribution function of $P_m$). The $L_2$-discrepancy of nonuniformity of $P_m$ can be defined as

$$D(\Omega^s, P_m) = \left[ \int_{\Omega^s} (F_m(\theta) - F(\theta))^2 d\theta \right]^{1/2}$$

The search for uniform designs with minimum $L_2$-discrepancy is an NP-hard problem [12]. Thus approximated methods are designed to find low discrepancy (i.e., closing the theoretical minimum discrepancy), such as centered $L_2$-discrepancy method [12]. A complete list of the uniform design (UD) tables, which is based on the centered $L_2$-discrepancy principle, can be found in UD-web (http://www.math.hkbu.edu.hk/UniformDesign).

Assume the element of the UD table is denoted as $q_{i,k}$, $i$ (or $k$) is the number of parameters (or experimental levels per parameter). We define an intermediate variable $c_{i,k}$, and let

$$c_{i,k} = \frac{2q_{i,k} - 1}{2n}, \quad k = 1, \ldots, n$$

where $j$ is the index on $i$, and $j = 1, \ldots, l-1$. Then the weight of $q_{i,k}$ for $s$ parameters with $n$ levels is uniformly set based on

$$x_{i,k} = \left(1-c_{i,k}\right)^{\frac{1}{s-j}} \prod_{j=1}^{l-1} c_{i,k}^{(s-j)/s}, \quad i = 1, \ldots, s-1$$

$$x_{k,d} = \prod_{j=1}^{l-1} c_{i,k}^{(s-j)/s}, \quad k = 1, \ldots, n$$

(27)

Based on the UD theory in [12], all the test points are uniformly selected in the experimental plan. Hence, the points $x_{i,k}$ generated by Eq. (27) are uniform design to parameters $s$ and $n$. However, there are at least two drawbacks in the existing UD method. First, it is expensive for computing Eq. (27). Second, the UD method does not consider the border points into the UD table. However, we can mathematically know that optimal results are often found in the border of parameters’ domain.

For solving the first drawback, UDEM uses a recursion method to decrease the computational cost. The details can be found in the pseudo of Algorithm 1.
For solving the second drawback, UDEM puts forward two improvements. In the first change, UDEM method adds border points (i.e., $\sigma = 1$ or $\sigma = 0$) into the experimental plans. Thus the UD method is a special case of our UDEM method. Moreover, the traditional single kernel model is also included into the mixture of kernels model by the modification. Obviously the UDEM method with mixture of kernels includes the UD method with a single kernel function. Thus the MKCCA method (including UDEM method and mixture of kernels) includes the traditional KCCA method (including UD method and a single kernel function).

The second improvement for solving the second drawback of the existing UD method bases on the literature [23] and our observation. That is, we bound the parameters’ domain into a smaller interval, i.e. $q$ is an integral and $q \leq 10$, $0 < \sigma < 5$ and $\omega \in [0.95, 1]$. From Fig. 2 we can see: the extrapolation ability (receiving by global kernel) is strengthened and the interpolation ability (receiving from local kernel) is weakened while the value of $\omega$ increases (i.e. the weight of global kernel in mixture of kernels model increases) in Fig. 2a. While the value of $\omega$ increases to some point (i.e. about 0.95 according to our observation), we can see that both the extrapolation ability and the interpolation ability will reach to a balance (i.e. appropriate extrapolation ability and the interpolation ability) in Fig. 2b. Now such a setting for parameter $\omega$ is enough for achieving interpolation ability and extrapolation ability for learning from data. That is, only a “pinch” of a Gaussian kernel needs to be added to the polynomial kernel for achieving both interpolation ability and extrapolation ability in the mixture of kernels model. More specifically, the polynomial kernel with low degree can show better extrapolation ability but lack of interpolation ability, thus it needs to be added a little interpolation ability by concatenating a Gaussian kernel which contains interpolation ability. Furthermore, such interpolation ability with low width is enough because the interpolation ability is achieved from its neighborhoods. Either grid search method or cross-validation method in single kernel model for model selection must search for the whole domain of the parameters, i.e. $q \in N$, $\sigma \in R$, and $\omega \in [0.1, 1]$.

According to the two improvements, our UDEM method can overcome the limitations of existing UD method. That is, UDEM uniformly sets experimental plans by considering the recipe (i.e. the parameter $\omega$) of the parameters $(q$ and $\sigma$) into the UD method. Moreover, the proposed UDEM method is proposed according to the UD theory [12], which has been theoretically proved to present minimal discrepancy to the methods, such as exhaustive searching the whole domain of the parameters. Obviously, to use the UDEM method for model selection also retains minimal discrepancy to those with exhaustive searching the whole domain of the parameters. The pseudo of the proposed UDEM method in our MKCCA method is presented in Algorithm 1.

Algorithm 1. UDEM algorithm.

1. Choose the levels for each parameter (the number of parameters is denoted as $s$, and the number of level is denoted as $n$).
2. Let $g_{kj} = 1$, $g_{0k} = 0$, $k = 1, \ldots , n$;
3. Compute recursively $g_{kj} = g_{kj+1} \times c_{kj}^{1/2}$, $j = s-1, \ldots , 1$;
4. $x_{kj} = \sqrt{g_{kj} - g_{kj+1}}$, $j = 1, \ldots , s$; $k = 1, \ldots , n$;
5. $x_{kj}$ is fed into the step 1 in Algorithm 2.

4.5. The MKCCA algorithm

Many real-life datasets contain thousands of features and easily result in many problems. For example, if the number of dimensions is larger than the number of instances, the maximal number of the dimension after dimensionality reduction is $n$. Such a case will not achieve the real objective of dimensionality reduction. Generally, a pre-processing step should be employed. In this paper, random projection method [5] is firstly employed to avoid such an issue because of its linear complexity and high accuracy. Then the reduced data are centered (i.e. zero-mean and unit-variance) before fed into Algorithm 2, which outlines the pseudo of the MKCCA algorithm.

In Algorithm 2, we firstly map the centered data into the new RKHS by a mixture of kernels. Then we perform covariance analysis using PCA to remove noises and redundancy before the centered data in the RKHS are fed into a CCA tool for effective dimensionality reduction. After the above steps, we can construct all kinds of learning assignments (e.g., classification, clustering, and others) in the reduced data (i.e. the return of Algorithm 2).

Algorithm 2. MKCCA algorithm.

Require: (Input:) MKCCA[$X^{(1)}, X^{(2)}$, $r$, $k$, $c$]

1. $X^{(1)}, X^{(2)}$: original data
2. $r$, $k$, $c$: retained dimensionality after random projection, PCA, and CCA respectively

Ensure: (output:) the reduced data

// Perform dimensionality reduction
3. $[p s | t] = \text{princomp}(X^{(1)}_{\text{RKHS}})$;
4. $[a b r u v] = \text{canoncorr}(X^{(1)}_{\text{RKHS}} \times p^{(1)}(, . , 1 : k), X^{(2)}_{\text{RKHS}} \times p^{(2)}(:, . , k))$;
5. return $v(. , . : c)$ or $u(. , . : c)$;

Note that: the values of parameter $k$ and $c$ will be decided by users or the expertise, the arguments in function princomp and canoncorr can be found in “HELP” part of Matlab.

Dimensionality reduction in the proposed MKCCA method can be informally represented as PCA + CCA in the new RKHS. Comparing the MKCCA method with the spectrum KCCA method: (1) In the new RKHS, the MKCCA method projects the original data into a “smaller” feature space but with a larger hypothesis space. (2) The MKCCA method can provide both interpolation ability and extrapolation ability by employing the mixture of kernels model. (3) Although both the MKCCA method and the spectrum KCCA method can be applied for multiple types of learning, no literature has analyzed it. Moreover, the MKCCA method is expected to be more effective and efficient.

The proposed UDEM model selection method can also be used in any kernel-based methods. However, the UD method with the single kernel model is a special case of the UDEM model with mixture of kernels. Both the UD method with the single kernel model and the UDEM method with mixture of kernels have the same complexity (i.e. $O(k)$, $k$ is the number of the levels per parameter) for model selection. However, the UDEM method with mixture of kernels model can achieve both the extrapolation ability and interpolation ability, and the UD method with the single kernel model only achieves one of these two abilities.

5. Experimental study

5.1. Experiment setting

In this section, we evaluate the proposed MKCCA method in terms of classification accuracy and the effectiveness of
dimensionality reduction with existing algorithms, including the CCA method, the spectrum KCCA method in [18], and the popular methods in each type of learning, such as (KPCA method [48]) in multi-view learning, Kernel Fisher Discriminant Analysis (KDA) [7] in supervised learning, and MMDE [31] in transfer learning respectively. We do not compare the MKCCA method to the existing methods in semi-supervised learning because it has been shown in [58,6] that the KCCA method outperforms most existing semi-supervised methods for classification.

Note that, there exist a lot of excellent algorithms on classification, such as SVM, boosting techniques, and many others. In the following experiments, we only compare the proposed MKCCA method with the aforementioned methods because we also want to show the proposed MKCCA method is a generalized method. That is, it can be applied for multiple types of learning, and are better than existing popular methods in each type of learning.

In the classification assignment, i.e. the first experiment, we implement our MKCCA method by Algorithm 2, and the compared algorithms with their corresponding techniques. After implementing dimensionality reduction by these algorithms, we perform classification assignment by employing k nearest neighbor classifier (k=8) in the reduced data to compare classification accuracy in different types of learning.

In the second experiment, we evaluate the effectiveness of dimensionality reduction in different types of learning. We set the keeping ratio as 20%, 40%, 60%, 80% and 1 ("1" means keeping all the dimensions, and "20%" preserves only 20% original dimensions after implementing dimensionality reduction). We firstly implement dimensionality reduction by these algorithms with the predefined keeping ratio. Then k nearest neighbor classifier is employed to obtain the classification accuracy in the reduced space. We also implement the classification on the original data by k nearest neighbor method, and denoted the algorithm as "original" in our experiments. We want to show whether or not the classification accuracy in the original data is better than the reduced data by all kinds of techniques on dimensionality reduction in all kinds of types of learning.

In all the following experiments, to avoid over-fitting, we follow the method in [31] to select data. That is, we randomly select 60% examples from the original data, and repeat the experiments 10 times. The final result is recorded as the averaged performance of these 10 individual results.

Each individual result is the best result in 10 experimental plans with the UDEM method for model selection. In each experimental plan, we set 10 levels for per parameter (i.e. \( k = 10 \) and \( s = 3 \) in Algorithm 1). The 10 levels are set as \( (0, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1) \) for \( \omega \), \( (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) \) for parameter \( q \) and \( (0.01, 0.05, 0.1, 0.25, 0.5, 1, 2, 3, 4, 5) \) for \( \sigma \) respectively. Then we implement Algorithm 1 and list the parameters setting of each experimental plan in Table 2. In Table 2 each row represents the order of each parameter, and each column means the order of parameters’ setting for implementing experimental plan once by Algorithm 2. For example, the data in first column (6, 7, 1) indicates that the corresponding order for parameter \( \omega, p \) and \( \sigma \) is 6, 7 and 1 respectively. Hence, corresponding to their level values, we set \( \omega = 0.95 \), \( q = 7 \) and \( \sigma = 0.01 \) for model selection in one experimental plan. After implementing all of these 10 columns, i.e. repeating the dimension reduction algorithm for 10 times, i.e. the UDEM method finishes 10 experimental plans. Then the best result in 10 experimental plans will be the final individual result for one individual result.

Hence, we only need to run each algorithm of dimensionality reduction 10 times for one individual result. And the results can present minimum \( L_2 \)-discrepancy [12] to the traditional model selection methods (e.g., cross-validation method under the model of a mixture of kernels) which may run procedure at least 1000 times. Although the single kernel model only needs to run 10 times, it cannot simultaneously obtain both extrapolation ability and interpolation ability.

All compared algorithms in this paper will employ our UDEM method. The parameter \( k \) and \( c \) in the MKCCA method (i.e. PCA and CCA) are set as the default values in Matlab, and the number of retained dimensions for the other algorithms are determined by the corresponding algorithms.

In the preprocessing phrase, for all algorithms, the datasets whose dimensionality is beyond 1000 will firstly be reduced by the random projection method, then to keep 1000 dimensions in our experiments. Our MKCCA method is implemented with MATLAB (R2009b edition) software running in PC (Microsoft Windows XP, Intel Core 2 Duo CPU, 4 GB of RAM).

5.2. Accuracy of classification

5.2.1. The comparison on multi-view learning

We use three real datasets for this set of experiments, including ads, citeeseer and webkb. Dataset ads is an image dataset and represents a set of possible advertisements on internet pages. It includes 3279 instances within two classes, i.e. 2821 instances for class nonads, 458 for class ads. Each instance in the dataset ads contains 1558 features within five views, i.e. 457 features for view url, 495 features for view origurl, 472 features for view anurl, 111 features for view alt terms and 19 features for view caption. We extract three views (i.e. url, origurl, and anurl) for our experiments and combine them to form two experiments on multi-view learning. We use url as outer source and view origurl as target source in the first multi-view learning. We denote it as ads12, i.e. url vs. origurl. We regard the former data source (i.e. url) as source data and the last data source (i.e. origurl) as target data throughout the paper. Another multi-view learning on dataset ads is ads13, i.e. url vs. anurl.

Dataset citeeseer is a text dataset. It contains 3312 instances and 201,960 features within three views. The first two views (i.e. text view and inlink view) contain 100,000 features and the outlink view is with 1960 features. The classification task in dataset citeeseer is to predict six classes, i.e. agents, AI, DB, IR, ML and HCI. We extract the view text view and the view inlink (i.e. text vs. inlink) to construct the third experiment on multi-view learning, and short it for citeeseer.

Dataset webkb is also a text dataset. It contains 4502 instances and 103,810 features within two views, i.e. view page text (100,000 features) and view link text (3810 features) for predicting six classes, i.e. Course (928 instances), Department (174 instances), Faculty (1119 instances), Project (504 instances), Staff (135 instances), Student (1641 instances). We denote webkb (i.e. page text vs. link text) as the fourth experiment on multi-view learning in this paper.

The other details on the datasets are summarized in Table 3.

In multi-view learning, the proposed MKCCA method is compared with CCA, KCCA [18], and RPCA [48]. Table 4 gives the results for the various methods. The value in bracket is the standard deviation. Table 5 shows the running time for MKCCA and KCCA.

We observe that the MKCCA method consistently outperforms other methods. Further, the classification accuracies of the kernel

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
</table>
| Experimental plans for each parameter with 10 levels.
| \( \omega \) | 6 | 2 | 5 | 1 | 7 | 9 | 4 | 10 | 8 | 3 |
| \( q \) | 7 | 8 | 3 | 5 | 1 | 6 | 10 | 4 | 9 | 2 |
| \( \sigma \) | 1 | 8 | 10 | 4 | 7 | 9 | 6 | 5 | 3 | 2 |
methods (KCCA, KPCA and MKCCA) are higher than CCA method. This is because the relationship among features in real-life datasets are typically nonlinear rather than linear.

KPCA method performs classification using only the information from one dataset. Both the MKCCA method and KCCA method use outer source. We note that the MKCCA method outperforms the KPCA method. Note that, in Table 3, the result of the KCCA method is worse than the KPCA method. That maybe include two reasons. First, the KCCA algorithm is unable to effectively remove the redundancy from source dataset. This limitation is overcome well in the MKCCA method through implementing the PCA process. Second, the properly regularization in the KCCA is not chosen.

All these algorithms involve the generalized eigenproblem, the CCA needs the minimal time cost, and the KPCA is the second best one. To construct the kernel matrix in the PCA requires time cost, so the CCA outperforms the KPCA on running time. The KPCA outperforms both the KCCA and the MKCCA on time cost since it only employs one dataset for the learning process. Although the MKCCA method is decomposed into two processes, i.e. the PCA followed by the CCA in the RKHS, it is faster than the KCCA method. This is because the PCA process in the MKCCA method can reduce noises as well as the number of dimensions of the original dataset so that the running time of the CCA process in the MKCCA method can be reduced.

5.2.2. The comparison on supervised learning

If we regard $X^{(2)}$ in Eq. (1) as class label, CCA-based methods can serve as supervised dimensionality reduction methods through some encoding methods, such as binary class label encoding, or one-of-c label encoding. In this paper, we denote $Y = (y_1, \ldots, y_N) \in \mathbb{R}^{N \times N}$ ($N_c$ or $N$ is the number of classes or instances respectively in the dataset) as the class label matrix. The binary element $y_i(c)$ takes 1 if the ith instance contains the c-label and 0 otherwise.

In supervised learning, we use four real datasets, including dataset sector, rcvl, protein and mnist from [9]. And their summary is presented in Table 6.

We use the KDA method [7] to replace the KPCA method in supervised learning. The experimental results are presented in Tables 7 and 8.

We observe that the proposed MKCCA method yields the best performance in two tables. In particular, kernel correlation analysis algorithms (i.e. KCCA method and MKCCA method) are better than the KDA method for three datasets except dataset rcvl. In the dataset rcvl, the difference between the KDA method and the KCCA method is not significant (i.e. 0.02%) for classification accuracy.

According to the experimental results, we know that kernel correlation analysis algorithms can also use class label for effective learning in supervised learning. Moreover, the proposed MKCCA method is the best in terms of classification accuracy. Moreover, the MKCCA outperforms the other kernel methods, such as the KDA and the KCCA on time cost.

5.2.3. The comparison on transfer learning

We use dataset WiFi [31] and dataset news [14] for this set of experiments.

Dataset WiFi records WiFi signal strength collected in different time phases, including d0826 (collected at 08:26 am), d1112, d1354, d1621 and d1910 respectively. There are 7140 instances and 11 features with 119 classes for each time phrase. With dataset WiFi, we construct two experiments on transfer learning, i.e. d0826 vs. d1910 (short for WiFi15), and d1112 vs. d1621 (short for WiFi24).

Dataset news contains approximately 20,000 newsgroup documents, which are partitioned across 20 different newsgroups. In our experiments, we select the domains comp, rec and Sci to build two experiments on transfer learning, i.e. comp vs. rec (short for newsCR) and rec vs. Sci (short for newsRS).

The MMDE algorithm [31] and the KTPCA (i.e. the kernel edition of the algorithm TPCA in [42]), as the state-of-the-art dimensionality reduction methods on transfer learning, is compared with CCA-based methods. Tables 9 and 10 give the experimental results.
We observe that the MKCCA method yields the best performance on transfer learning where the distribution of the outer source is different from the distribution of the target source. Due to performing an optimization problem for generating the kernel matrix, the MMDE method is the most expensive on time cost among the kernel methods. Moreover, the KTPCA outperforms the MMDE. Due to employing the gradient method, the KTPCA needs the most cost on running time.

5.2.4. Conclusion on classification accuracy

In these three types of learning, i.e., multi-view learning, supervised learning and transfer learning, our MKCCA method outperforms the popular algorithms in each learning type on classification accuracy. Meantime, MKCCA method outperforms the other CCA-based method, such as, CCA and spectrum KCCA. We can make a conclusion that our MKCCA method can be applied in different types of learning, and performs better performance than existing popular algorithms in the corresponding learning type. Moreover, the proposed MKCCA method achieves better performance than the spectrum KCCA method on both the classification accuracy and the running time in these three types of learning.

Because no literature has employed CCA-based methods for implementing dimensionality reduction in transfer learning model, our MKCCA method obtains the better classification accuracy over the existing MMDE method in transfer learning model.

5.3. Effectiveness of dimensionality reduction

In this subsection, we investigate the effectiveness of dimensionality reduction for different dimensionality reduction methods. We construct two \textit{kNN} classifiers. One is built in the reduced space and the other is built in the original space.

Figs. 3–5 show the results for six datasets, where the \textit{x}-axis represents the keep ratio after implementing dimensionality reduction, and \textit{y}-axis means classification accuracy. We can conclude three main observations from the results presented in Fig. 5. Firstly and the most importantly, the proposed MKCCA method consistently outperforms all the other methods in the reduced subspaces with different kept ratio. This indicates that the MKCCA method can better identify redundant dimensions and noises in the new RKHS with a mixture of kernels. We also find that kernel methods are more successful than the standard CCA method for finding a subspace.

Secondly, different methods reach their peak performance in the different subspaces for different datasets. This shows that it is actually difficult to set the optimal subspace in practice since different datasets may have different characteristics. Moreover, the parameter setting could be another issue in affecting the performance.

Thirdly, in Fig. 4a for \textit{mnist} and Fig. 5a for WiFi, a few methods (such as the MKCCA method in the two cases, and the KTPCA in the transfer learning) outperform the classifier on the original data. This maybe because datasets \textit{mnist} and \textit{WiFi} have very low original dimensions (780 for \textit{mnist} and 11 for \textit{WiFi}). This confirms that our MKCCA method is more reliable than others in low dimensional space.

5.4. Discussion

In this subsection, we provide further discussions on the proposed MKCCA method.

Table 9

<table>
<thead>
<tr>
<th>Method</th>
<th>WiFi15</th>
<th>WiFi24</th>
<th>newsCR</th>
<th>newsRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCA</td>
<td>0.571(0.013)</td>
<td>0.572(0.018)</td>
<td>0.503(0.013)</td>
<td>0.717(0.066)</td>
</tr>
<tr>
<td>KCCA</td>
<td>0.57(0.012)</td>
<td>0.576(0.018)</td>
<td>0.523(0.015)</td>
<td>0.739(0.014)</td>
</tr>
<tr>
<td>MMDE</td>
<td>0.635(0.021)</td>
<td>0.642(0.018)</td>
<td>0.635(0.039)</td>
<td>0.737(0.018)</td>
</tr>
<tr>
<td>KTPCA</td>
<td>0.647(0.016)</td>
<td>0.650(0.042)</td>
<td>0.652(0.023)</td>
<td>0.742(0.031)</td>
</tr>
<tr>
<td>MKCCA</td>
<td>0.668(0.017)</td>
<td>0.678(0.018)</td>
<td>0.682(0.022)</td>
<td>0.745(0.015)</td>
</tr>
</tbody>
</table>

Table 10

Comparing run time (s) on transfer learning.

<table>
<thead>
<tr>
<th>Method</th>
<th>WiFi15</th>
<th>WiFi24</th>
<th>newsCR</th>
<th>newsRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCA</td>
<td>6.504</td>
<td>5.457</td>
<td>3.922</td>
<td>2.465</td>
</tr>
<tr>
<td>MMDE</td>
<td>15.278</td>
<td>16.287</td>
<td>6.485</td>
<td>6.004</td>
</tr>
<tr>
<td>KTPCA</td>
<td>31.453</td>
<td>21.365</td>
<td>8.713</td>
<td>9.175</td>
</tr>
<tr>
<td>MKCCA</td>
<td>8.388</td>
<td>8.323</td>
<td>4.226</td>
<td>4.320</td>
</tr>
</tbody>
</table>

Fig. 3. Reduced dimension effect on multi-view learning. (a) ads (url vs. origurl), (b) citeseer.
5.4.1. Advantages of PCA in the MKCCA method

As we mentioned, MKCCA involves two processes for performing dimensionality reduction, i.e. PCA followed by CCA in the new RKHS.

Such an implementation on dimensionality reduction contains at least two advantages. First, the PCA process in the high dimensional space can increase the effect of dimensionality reduction by efficiently removing noise and redundancy. In fact, in the process of implementing PCA method, the diagonal terms of eigenvalue matrix are ranked in a decreasing order, and all the off-diagonal terms are zero. Thus PCA can effectively remove noise and redundancy by eliminating those insignificant principal components corresponding to zero or trivial eigenvalues. Moreover, noise or redundancy can be detected more easily in higher dimensional spaces (i.e. RKHS) than the original spaces.

Second, the PCA process can help deal with the issue of trivial learning. On the one hand, some noises or redundancy have been removed by PCA. That decreases the probability of trivial learning in the reduced subspace. On the other hand, the MKCCA method (i.e. PCA followed by CCA) essentially induces a linear dependence among features, then makes an unregularized CCA step for nontrivial learning.

5.4.2. Advantages of the UDEM method

The main advantages of our UDEM method over existing methods (such as grid search method or cross-validation method) for model selection include the followings.

First, the points selected by our UDEM method are “far more uniform” and “far more space filling” than lattice grid points chosen by existing methods. That is, the UDEM method can basically 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 complexity of the UDEM method for model selection is linear to the number of the levels per parameter, i.e. $O(k)$. However, the traditional methods (e.g., cross-validation method) for model selection is exponential to the number of parameters (i.e. $O(k^l)$).
Second, in the UDEM method, the single kernel model becomes a special case of our MKCCA method. Although the mixture of kernels model in the MKCCA method needs to set three parameters, it is with same complexity for model selection to the model with the single kernel function. Moreover, the MKCCA method with UDEM method can obtain both interpolation ability and extrapolation ability in the learning process.

Third, the purpose of introducing the mixture of kernels in the MKCCA method is to embed a “pinch” of Gaussian kernel into the lower order polynomial kernel for adding the interpolation ability into the learning process. From Fig. 2, we can see that the higher order degree in polynomial kernel trends to present interpolation ability and its lower order presents the extrapolation ability. Intuitively, we can build a mixture of kernels model by combining the polynomial kernel of higher order degree with the polynomial kernel of lower order degree, to achieve both interpolation ability and extrapolation ability. However, this will result in at least two issues. First, the polynomial kernel of higher order degree will generate large magnitude over the lower order one. So the learning will be biased to interpolation ability rather than to balance the two abilities. Second, such an interpolation ability will present a global influence. However, the interpolation ability only needs to learn from the neighbors of the test point. Therefore, to achieve better interpolation ability, the Gaussian kernel is a good alternative. For balancing the magnitude of the different kernel values between the polynomial kernel and the Gaussian kernel, a small width in Gaussian kernel is enough. For example, in Fig. 2, the smaller value of parameter $\sigma$ will not result in better extrapolation ability because we cannot balance the magnitude of the kernel values between these two kernel functions. Moreover, it cannot also balance the interpolation ability and the extrapolation ability. In a word, for achieving the best interpolation and extrapolation abilities, we only need to set $\sigma$ at lower order, small value for $\sigma$ and higher weight for $\omega$. This can not only strengthen the learning but also decreases the complexity for model selection.

5.4.3. Advantages of the MKCCA method

In short, the MKCCA method achieves satisfactory performance for different types of learning in terms of classification accuracy and dimension reduction effectiveness in real datasets. The CCA method presents the worst result because it always regards the relationship between two variables as linear.

In fact, CCA-base methods (e.g., CCA, or the MKCCA method) have been applied for all kinds of learning assignments or scenarios, such as classification [18], regression [15], clustering [6,10], and dimensionality reduction [13]. However, no literature has talked about that the CCA-based methods can be applied for multiple learning assignments in multiple types of learning. In this paper, we know that the proposed MKCCA method can be applied for all kinds of learning assignments (e.g., dimensionality reduction, classification, regression, clustering and others) in different types of learning, such as multi-view learning (e.g. [10]), supervised learning (e.g., this paper), semi-supervised learning (e.g., [6]), and transfer learning (e.g., this paper). Hence, we can regard CCA-based methods (including our MKCCA method) as generalized learning methods.

6. Conclusion and future work

In this paper, we have presented a correlation analysis algorithm, called MKCCA, for dimensionality reduction in multiple types of learning. The proposed algorithm performs dimensionality reduction with two processes, i.e. the PCA followed by the CCA in the new RKHS mapped by a mixture of kernels. Then we can implement different types of learning with the reduced data. The experimental results on real datasets demonstrated that the MKCCA method achieves the best performance over existing methods for different types of learning. In future, we plan to further explore the difference between our method with the mixture of kernels and multiple kernel learning.

REFERENCES

Xiaofeng Zhu is a PhD candidate in Information Technology & Electrical Engineering, The University of Queensland. His research topics include feature selection and analysis, pattern recognition and data mining.

Zi Huang is a lecturer and an Australian postdoctoral fellow in School of Information Technology & Electrical Engineering, The University of Queensland. She received her BSc degree from Department of Computer Science, Tsinghua University, China, and her PhD in computer science from School of IEEE, The University of Queensland. (polynomial kernel) Dr. Huang’s research interests include multimedia search, knowledge discovery, and bioinformatics.

Heng Tao Shen is a professor of computer science in School of Information Technology & Electrical Engineering, The University of Queensland. He obtained his BSc (with 1st class honours) and PhD from Department of Computer Science, National University of Singapore in 2000 and 2004 respectively. His research interests include multimedia/mobile/web search, database management, dimensionality reduction, etc. Heng Tao has extensively published and served on program committees in most prestigious international publication venues in multimedia and database societies. He is also the winner of Chris Wallace Award for outstanding Research Contribution in 2010 from CORE Australasia. He is a senior member of IEEE and member of ACM.

Jian Cheng is an associate professor at Institute of Automation, Chinese Academy of Sciences. He received his PhD degree from Institute of Automation, Chinese Academy of Sciences in 2004. His research interests include information retrieval, video analysis, and machine learning.

Changsheng Xu is a professor in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He is also an executive director of China–Singapore Institute of Digital media. He received the PhD degree from Tsinghua University, China in 1996. He was with National Lab of Patterns Recognition, Institute of Automation, Chinese Academy of Sciences as a postdoctoral fellow and associate professor from 1996 to 1998. He was with Institute for Infocomm Research, Singapore from 1998 to 2008. His research interests include multimedia content analysis, image processing, pattern recognition, computer vision, and digital watermarking. He has published over 150 refereed book chapters, journal and conference papers in these areas. He is a senior member of IEEE and member of ACM.