Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 427462, 8 pages
http://dx.doi.org/10.1155/2013/427462

Research Article

Multiple Kernel Spectral Regression for Dimensionality Reduction

Bing Liu, Shixiong Xia, and Yong Zhou

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Shixiong Xia; xiasx@cumt.edu.cn

Received 13 July 2013; Revised 18 September 2013; Accepted 20 September 2013

Academic Editor: Zhihua Zhang

Traditional manifold learning algorithms, such as locally linear embedding, Isomap, and Laplacian eigenmap, only provide the embedding results of the training samples. To solve the out-of-sample extension problem, spectral regression (SR) solves the problem of learning an embedding function by establishing a regression framework, which can avoid eigen-decomposition of dense matrices. Motivated by the effectiveness of SR, we incorporate multiple kernel learning (MKL) into SR for dimensionality reduction. The proposed approach (termed MKL-SR) seeks an embedding function in the Reproducing Kernel Hilbert Space (RKHS) induced by the multiple base kernels. An MKL-SR algorithm is proposed to improve the performance of kernel-based SR (KSR) further. Furthermore, the proposed MKL-SR algorithm can be performed in the supervised, unsupervised, and semi-supervised situation. Experimental results on supervised classification and semi-supervised classification demonstrate the effectiveness and efficiency of our algorithm.

1. Introduction

In real applications, the resulting data representations are generally high dimensional. Practical algorithms usually behave badly when faced with many unnecessary features. Hence, finding a way of transforming them into a unified space of lower dimension can facilitate the underlying tasks such as pattern recognition or regression problems. Dimensionality reduction (DR) techniques, which have been widely used in many fields of information processing, include unsupervised, supervised, and semisupervised methods due to different assumptions about the data distribution or the availability of the data labeling.

In order to handle the data sampled from a nonlinear low dimensional manifold, many manifold learning techniques, such as ISOMAP [1], Locally Linear Embedding (LLE) [2], and Laplacian Eigenmap [3], have been proposed in recent years, which reduce the dimensionality of a fixed training set in a way that can maximally preserve certain interpoint relationships. One of the major limitations of these methods is that they do not generally address the out-of-sample problem. Although some methods explicitly require an embedding function either linear or in RKHS when minimizing the objective function [4, 5], the computation of these methods involves eigendecomposition of dense matrices which is expensive in both time and memory. Spectral regression (SR), which is fundamentally based on regression and spectral graph analysis [6–10], can avoid eigen-decomposition of dense matrices and has better performance at a faster learning speed. Moreover, it can be performed either in supervised, unsupervised, or semisupervised situation. Kernel SR (KSR) is the kernelized version of SR in the reproducing kernel Hilbert space (RKHS), which can further improve the performance of SR. While KSR is based on a single kernel, in practice it is often hard to select a suitable kernel. A common way to an automatic selection of optimal kernels is to learn a linear combination of base kernels. Motivated by the effectiveness of SR, we introduce a framework called MKL-SR that incorporates multiple kernel learning (MKL) into the training process of SR. We will illustrate the formulation of MKL-SR with graph embedding [11], which provides a unified view for a large family of DR methods. Any DR technique expressible by graph embedding can therefore be generalized by MKL-SR to boost their power by automatically selecting optimal kernels. As the corresponding SR algorithm would do, the proposed approach not only solves the out-of-sample extension problem but also improves the performance of kernel-based SR (KSR) for the supervised, semisupervised, and unsupervised learning problems.

The paper is structured as follows. In Section 2, we briefly introduce the related work. We provide the MKL-SR framework and present the optimization process in Section 3. The experimental results are shown in Section 4. Finally, we give the related conclusions in Section 5.

2. Related Work

Since the relevant literature is quite extensive, our survey instead emphasizes the key concepts crucial to the establishment of the proposed framework.

2.1. Spectral Regression Algorithm. In the traditional spectral dimensionality reduction algorithms, seeking an embedding function which minimizes the objective function involves eigen-decomposition of dense matrices, which has the high computational cost in both time and memory. The SR algorithm uses the least squares method to get the best projection direction, instead of computing the density matrix of features, so it has much faster learning speed. An affinity graph **G** of both labeled and unlabeled points is constructed to find the intrinsic geometry structure and to learn the responses with the given data. Then, with these responses, the ordinary regression is applied to learning the embedding function.

Given a training set with l labeled samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l$ and $(n - l)$ unlabeled samples $\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_m$, where the sample $\mathbf{x}_i \in R^d$ belongs to one of c classes, let l_k be the number of labeled samples in the kth class (the sum of l_k is equal to l). The SR algorithm is summarized as follows.

Step 1. Constructing the adjacency graph **G** let **X** be the training set and let **G** denote a graph with n nodes, where the ith node corresponds to the sample $\mathbf{x}_i$. In order to model the local structure as well as the label information, the graph **G** will be constructed through the following three steps.

(1) If $\mathbf{x}_i$ is among p-nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among p-nearest neighbors of $\mathbf{x}_i$, then nodes i and j are connected by an edge.

(2) If $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same class (i.e., same label), then nodes i and j are also connected by an edge.

(3) Otherwise, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are not in the same class, then the edge will be deleted between nodes i and j.

Step 2. Constructing the weight matrix **W** let **W** be the sparse symmetric $n \times n$ matrix, where $\mathbf{W}_{ij}$ represents the weight of the edge joining vertices i and j.

(1) If there is no any edge between nodes i and j, then $\mathbf{W}_{ij} = 0$.

(2) Otherwise, if both $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the kth class, then $\mathbf{W}_{ij} = 1/l_k$, else $\mathbf{W}_{ij} = \delta \cdot s(i, j)$,

where δ $(0 < \delta \leq 1)$ is a given parameter to adjust the weight between supervised and unsupervised neighbor information. Therein, $s(i, j)$ is a similarity evaluation function between $\mathbf{x}_i$ and $\mathbf{x}_j$; we have two variations, the first one is simple-minded function $s(i, j) = 1$ and the second one is heat kernel function:

$$s(i, j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right), \quad (1)$$

where $\sigma \in \mathbb{R}$.

Step 3. For eigen-decomposing let **D** be the $n \times n$ diagonal matrix, whose (i, i)th element is the sum of the ith column (or row) of **W**. Find $y_0, y_1, \ldots, y_{c-1}$, which are the largest c generalized eigenvectors of the eigenproblem

$$\mathbf{W}\mathbf{y} = \lambda \mathbf{D}\mathbf{y}, \quad (2)$$

where the first eigenvector y_0 is a vector of all ones with eigenvalue 1.

Step 4. Calculate $c - 1$ vectors $\mathbf{a}_1, \ldots, \mathbf{a}_{m-1} \epsilon \mathbb{R}^d$. $\mathbf{a}_k$ $(k = 1, \ldots, c - 1)$ is the solution of the regularized least square problem

$$\mathbf{a}_k = \arg\min_{\mathbf{a}} \left(\sum_{i=1}^{n} \left(\mathbf{a}^{\mathrm{T}} \mathbf{x}_i - y_i^k\right)^2 + \gamma \|\mathbf{a}\|^2 \right), \quad (3)$$

where y_i^k is the ith element of $\mathbf{y}^k$.

Step 5. Let **A** be an $d \times (c - 1)$ transformation matrix, where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_{c-1}]$. The testing samples or new sample can be embedded into $c - 1$ dimensional subspace by

$$\mathbf{x} \rightarrow \mathbf{z} = \mathbf{A}^{\mathrm{T}} \mathbf{x}. \quad (4)$$

Next, we briefly discuss the kernel spectral regression. If we choose a nonlinear function in RKHS; that is, $y_i = f(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$, and $K(\mathbf{x}_i, \mathbf{x}_j)$ is the Mercer kernel of RKHS $\mathscr{H}_K$. Equation (3) can be rewritten as

$$\min_{\boldsymbol{\alpha}_k} \sum_{i=1}^{n} \left(\mathbf{K}\boldsymbol{\alpha}_k - y_i^k\right)^2 + \alpha \|f\|_K^2, \quad (5)$$

where **K** is $n \times n$ gram matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. Find $c - 1$ vectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{c-1} \in \mathbb{R}^n$. $\boldsymbol{\alpha}_k$ $(k = 1, \ldots, c - 1)$ is the solution of the following linear equations system:

$$(\mathbf{K} + \alpha I)\, \boldsymbol{\alpha}_k = \mathbf{y}_k. \quad (6)$$

Let $\Theta = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{c-1}]$, Θ is a $n \times (c - 1)$ transformation matrix. The samples can be embedded into $c - 1$ dimensional subspace by

$$\mathbf{x} \rightarrow \mathbf{z} = \Theta^T K(\cdot, \mathbf{x}), \quad (7)$$

where $K(\cdot, \mathbf{x}) = [K(\mathbf{x}_1, \mathbf{x}), \ldots, K(\mathbf{x}_n, \mathbf{x})]^T$.

2.2. Multiple Kernel Learning. MKL learns a kernel machine with multiple kernel functions or kernel matrices. Recent studies have shown that MKL not only increases the recognition accuracy but also enhances the interpretability of the resulting classifiers. Given a set of base kernel functions $\{k_m\}_{m=1}^M$, an ensemble kernel function k is defined by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{M} \beta_m k_m(\mathbf{x}_i, \mathbf{x}_j), \quad \beta_m \geq 0. \tag{8}$$

Consequently, an often-used MKL decision function derived from binary-class SVM is

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + \mathbf{b} = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{M} \beta_i k_m(\mathbf{x}_i, \mathbf{x}) + \mathbf{b}. \tag{9}$$

The training process of MKL generally optimizes over both the coefficients $\{(\alpha_i)\}_{i=1}^N$ and $\{(\beta_m)\}_{m=1}^M$.

In recent years, dimensionality reduction methods based on multiple kernels have been proposed to improve the performance of those using single kernel. In [12], kernel learning was first incorporated into DR methods. Then, a multiple kernel DR framework was designed in [13]. Recently, Zhu et al. proposed a dimensionality reduction method by Mixed Kernel Canonical Correlation Analysis (CCA) [14, 15]. In this method, the high dimensional data space is mapped into the reproducing kernel Hilbert space (RKHS) with a linear combination between a local kernel and a global kernel. Kernel CCA is further improved by performing Principal Component Analysis (PCA) followed by CCA for effective dimensionality reduction, which can be implemented in supervised learning, semisupervised learning, and transfer learning. Motivated by their work, we aim to incorporate the MKL optimization into SR to yield more flexible dimensionality reduction schemes.

3. The MKL-SR Framework

We first explain how to integrate MKL and SR for dimensionality reduction. Then, we propose an optimization procedure to complete the framework.

3.1. MKL-SR Model. Suppose that the ensemble kernel K in MKL-SR is generated by linearly combining the base kernels $\{k_m\}_{m=1}^M$ as in (8). Selecting a nonlinear function in RKHS induced by the kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^M \beta_m k_m(\mathbf{x}_i, \mathbf{x}_j)$, we have $y_i = f(\mathbf{x}_i) = \sum_{j=1}^n \sum_{m=1}^M \alpha_j \beta_m k_m(\mathbf{x}_i, \mathbf{x}_j)$. The constrained optimization problem for 1D MKL-SR is defined as follows:

$$\min_{\alpha,\beta} \sum_{i,j=1}^{N} \left\| \boldsymbol{\alpha}^T \mathbb{K}^{(i)} \boldsymbol{\beta} - \boldsymbol{\alpha}^T \mathbb{K}^{(j)} \boldsymbol{\beta} \right\|^2 w_{ij} \tag{10}$$

$$\text{Subject to} \quad \sum_{i,j=1}^{N} \left\| \boldsymbol{\alpha}^T \mathbb{K}^{(i)} \boldsymbol{\beta} \right\|^2 d_{ii} = 1 \tag{11}$$

$$\beta_m \geq 0, \quad m = 1, 2, \ldots, M, \tag{12}$$

where

$$\begin{gathered} \boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T \in \mathbb{R}^n, \\ \boldsymbol{\beta} = [\beta_1, \ldots, \beta_M]^T \in \mathbb{R}^M, \\ \mathbb{K}^{(i)} = \begin{bmatrix} k_1(1,i) & \cdots & k_M(1,i) \\ \vdots & \ddots & \vdots \\ k_1(n,i) & \cdots & k_M(n,i) \end{bmatrix} \in \mathbb{R}^{n\times M}. \end{gathered} \tag{13}$$

The additional constraints in (12) arise from the use of the ensemble kernel in (8) and are to ensure that the resulting kernel K in MKL-SR is a nonnegative combination of base kernels.

Observe from (10) that the one-dimensional projection of MKL-SR is specified by a sample coefficient vector $\boldsymbol{\alpha}$ and a kernel weight vector $\boldsymbol{\beta}$. The two vectors, respectively, account for the relative importance among the samples and the base kernels in the construction of the projection. To generalize the formulation to uncover a multidimensional projection, we consider a set of $c - 1$ sample coefficient vectors, denoted by

$$\mathbf{A} = [\boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \cdots \boldsymbol{\alpha}_{c-1}]. \tag{14}$$

The resulting projection will map samples to a $(c - 1)$-dimensional euclidean space. Similar to the 1D case, a projected sample $\mathbf{x}_i$ can be written as

$$\mathbf{A}^T \mathbb{K}^{(i)} \boldsymbol{\beta} \in \mathbb{R}^{c-1}. \tag{15}$$

The optimization problem (10) can now be extended to multidimensional MKL-SR as

$$\min_{\mathbf{A},\boldsymbol{\beta}} \sum_{i,j=1}^{N} \left\| \mathbf{A}^T \mathbb{K}^{(i)} \boldsymbol{\beta} – \mathbf{A}^T \mathbb{K}^{(j)} \boldsymbol{\beta} \right\|^2 w_{ij}, \tag{16}$$

$$\begin{gathered} \text{subject to} \quad \sum_{i,j=1}^{N} \left\| \mathbf{A}^T \mathbb{K}^{(i)} \boldsymbol{\beta} \right\|^2 d_{ii} = 1, \\ \beta_m \geq 0, \quad m = 1, 2, \ldots, M. \end{gathered} \tag{17}$$

3.2. Optimization Algorithm. Since direct optimization to (16) is difficult, we instead adopt an iterative, two-step strategy to alternately optimize $\mathbf{A}$ and $\boldsymbol{\beta}$. At each iteration, one of $\mathbf{A}$ and $\boldsymbol{\beta}$ is optimized while the other is fixed, and then the roles of $\mathbf{A}$ and $\boldsymbol{\beta}$ are switched. Iterations are repeated until convergence or a maximum number of iteration is reached.

3.2.1. On Optimizing $\mathbf{A}$. We can indirectly utilize 1D MKL-SR to solve multidimensional MKL-SR. By fixing $\boldsymbol{\beta}$, the problem (10) can be transformed into the following optimal problem:

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}^T \mathbb{K} W \mathbb{K}^T \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbb{K} D \mathbb{K}^T \boldsymbol{\alpha}}, \tag{18}$$

where $\mathbb{K} = [\mathbb{K}^{(1)}\boldsymbol{\beta}, \ldots, \mathbb{K}^{(n)}\boldsymbol{\beta}] \in \mathbb{R}^{n\times n}$. The optimal $\boldsymbol{\alpha}$'s are the eigenvectors corresponding to the maximum eigenvalue of the eigenproblem

$$\mathbb{K} W \mathbb{K} \boldsymbol{\alpha} = \lambda \mathbb{K} D \mathbb{K} \boldsymbol{\alpha}. \tag{19}$$

```
The training procedure of MKL-SR
  Input: A set of training data, matrices $W$ and $D$, a set of base kernels $\{k_m\}_{m=1}^{M}$
         and parameter $\gamma$ in (21).
  Output: Sample coefficient vectors $\mathbf{A} = [\boldsymbol{\alpha}_1 \ \ \boldsymbol{\alpha}_2 \ \ \ldots \ \ \boldsymbol{\alpha}_{c-1}]$;
          Kernel weight vectors $\boldsymbol{\beta}$;
  Make an initial guess for $\mathbf{A}$ or $\boldsymbol{\beta}$;
  Computing the largest $c$ generalized eigenvectors $\mathbf{y}^k$ ($k = 1, \ldots, c-1$) of eigen-problem (2);
  For $\mathbf{i} \leftarrow 1, 2, \ldots, \mathbf{i}_{\max}$ do
      (1) Compute $\mathbb{K}$;
      (2) $\mathbf{A}$ is obtained by solving the least squares problem (21);
      (3) Compute $\mathbb{K}'^T W \mathbb{K}'$ in (24) and $\mathbb{K}'^T D \mathbb{K}'$ in (25);
      (4) $\boldsymbol{\beta}$ is optimized by solving optimization problem (24) via SDP;
  Return $\mathbf{A}$ and $\boldsymbol{\beta}$;
```

Algorithm 1: MKL-SR algorithm.

Consequently, the columns of the optimal $A^* = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{c-1}]$ in (16) are the eigenvectors corresponding to the first $c-1$ smallest eigenvalues in (19).

Solving the problem (19) directly involves eigen-decomposition of dense matrices, which has the high computational cost in both time and memory. In order to solve the eigenproblem in (19) efficiently, we use the following theorem.

Theorem 1. *Let* $\mathbf{y}$ *be the eigenvector of the eigenproblem in* (2) *with eigenvalue* λ. *If* $\mathbb{K}\boldsymbol{\alpha} = \mathbf{y}$*; then* $\boldsymbol{\alpha}$ *is the eigenvector of the eigenproblem in* (19) *with the same eigenvalue* λ.

Proof. We have $W\mathbf{y} = \lambda D\mathbf{y}$. At the left side of (19), replacing $\mathbb{K}\boldsymbol{\alpha}$ by $\mathbf{y}$, we have

$$\mathbb{K}W\mathbb{K}\boldsymbol{\alpha} = \mathbb{K}W\mathbf{y} = \mathbb{K}\lambda D\mathbf{y} = \lambda\mathbb{K}D\mathbf{y} = \lambda\mathbb{K}D\mathbb{K}\boldsymbol{\alpha}. \tag{20}$$

Thus, $\boldsymbol{\alpha}$ is the eigenvector of the eigenproblem (19) with the same eigenvalue λ. □

Theorem 1 shows that, instead of solving the eigenproblem (19), the embedding functions can be acquired through two steps.

(1) Solve the eigenproblem in (2) to get $\mathbf{y}$.

(2) Find $\boldsymbol{\alpha}$ which satisfies $\mathbb{K}\boldsymbol{\alpha} = \mathbf{y}$. Similar to SR, a possible way is to find $\boldsymbol{\alpha}$ which can best fit the equation in the least squares sense as

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{n} \left(\boldsymbol{\alpha}^T \mathbb{K}^{(i)} \boldsymbol{\beta} - \mathbf{y}_i \right)^2 + \gamma \|\boldsymbol{\alpha}\|^2 \right), \tag{21}$$

where $\mathbf{y}_i$ is the ith element of $\mathbf{y}$.

Since the matrix $\mathbf{D}$ is guaranteed to be positive definite, the eigenproblem in (2) can be stably solved. Moreover, both $\mathbf{D} - \mathbf{W}$ and $\mathbf{D}$ are sparse matrices. The top c eigenvectors of eigenproblem in (2) can be efficiently calculated with Lanczos algorithms [13]. In addition, the technique to solve the least square problem is already matured and there exist many efficient iterative algorithms that can handle very large scale least square problems.

3.2.2. On Optimizing $\boldsymbol{\beta}$. By fixing A, the optimization problem (16) becomes

$$\min_{\boldsymbol{\beta}} \quad \boldsymbol{\beta}^T \mathbb{K}'^T W \mathbb{K}' \boldsymbol{\beta} \tag{22}$$

$$\text{Subject to} \quad \boldsymbol{\beta}^T \mathbb{K}'^T D \mathbb{K}' \boldsymbol{\beta} = 1, \quad \boldsymbol{\beta} \geq 0, \tag{23}$$

where $\mathbb{K}' = [\mathbf{A}^T\mathbb{K}^{(1)}, \ldots, \mathbf{A}^T\mathbb{K}^{(n)}]^T \in \mathbb{R}^{n \times M}$.

The additional constraints $\boldsymbol{\beta} \geq 0$ cause the optimization to (22) to be no longer transformed into a generalized eigenvalue problem. It is actually a nonconvex quadratically constrained quadratic programming (QCQP) problem [13], which is a NP-hard problem. Thus, we instead consider solving its convex relaxation by adding an auxiliary variable T of size $M \times M$ as

$$\min_{\boldsymbol{\beta}, B} \quad \operatorname{trace}\left(\mathbb{K}'^T W \mathbb{K}' T \right) \tag{24}$$

$$\text{Subject to} \quad \operatorname{trace}\left(\mathbb{K}'^T D \mathbb{K}' T \right) = 1, \tag{25}$$

$$\mathbf{e}_m^T \boldsymbol{\beta} \geq 0, \quad m = 1, 2, \ldots, M, \tag{26}$$

$$\begin{bmatrix} 1 & \boldsymbol{\beta}^T \\ \boldsymbol{\beta} & T \end{bmatrix} \succcurlyeq 0, \tag{27}$$

where $\mathbf{e}_m$ in (26) is a column vector whose elements are 0 except that its mth element is 1. To obtain the convex relaxation of the nonconvex QCQP problem (22), we relax the equation $T = \boldsymbol{\beta}\boldsymbol{\beta}^T$ to $T \succcurlyeq \boldsymbol{\beta}\boldsymbol{\beta}^T$, which can be equivalently expressed by the constraint in (27) according to the Schur complement lemma [16]. The optimization problem (24) is a semidefinite programming (SDP), which can be efficiently solved. It can be note that the numbers of constraints and variables in (24) are linear and quadratic to M, respectively. In practice, the value of M is often small. Thus, the proposed MKL-SR algorithm listed in Algorithm 1 mainly includes a sequence of SR training.

3.3. Novel Sample Embedding. After accomplishing the training procedure of MKL-SR, we can project a testing sample $\mathbf{z}$ into the learned subspace by

$$\mathbf{z} \rightarrow \mathbf{A}^T \mathbb{K}^{(\mathbf{z})} \boldsymbol{\beta}, \tag{28}$$

where

$$\mathbb{K}^{(\mathbf{z})} \in \mathbb{R}^{n \times M}, \qquad \mathbb{K}^{(\mathbf{z})}(i, m) = k_m(x_i, \mathbf{z}). \tag{29}$$

Several algorithms such as the nearest neighbor rule or k-means clustering can be used to complete classification or clustering tasks. In the experiments of this paper, we specifically discuss the effectiveness of MKL-SR in different learning tasks, including unsupervised learning for clustering, supervised, and semisupervised learning for face recognition.

4. Experiments

We used seven datasets (ionosphere, letter, digit, and satellite) from the UCI machine learning repository to perform unsupervised learning task. For the letter and satellite data sets, we only used their first two classes. Several multiclass data sets were created from the digits data. The experiments on supervised and semisupervised classification were performed on the CMU PIE face data set and the extended Yale B data set [17, 18], respectively. All the face images are manually aligned and cropped. The pixel values are scaled to [0, 1]. The basic information about these data sets is listed in Table 1. All the experiments have been performed in MATLAB 7.14.0 environment running in a 3.10 GHZ Intel Core i5-2400 with 3GB RAM.

Table 1: Description of the datasets.

Data	Size (n)	Feature (d)	Class
Ionosphere	351	34	2
Letter A-B	1555	16	2
Satellite C1-C2	2236	36	2
Digits 0689	713	64	4
Digits 1279	718	64	4
CMU PIE	850	1024	5
Extended Yale B	2114	1024	38

4.1. Experiments on Unsupervised Learning. To validate that MKL-SR is effective for an unsupervised dimensionality reduction task, we applied the proposed algorithm as a tool to learn an appropriate kernel function for KSR. Each data set was reduced by SR, single kernel based SR, kernel principal component analysis (KPCA), and MKL-SR, respectively. The normalized cut spectral clustering (NC) algorithm was adopted to evaluate the clustering performance on the reduced data. We set the number of clusters equal to the true number of classes and compared the clusters generated by these algorithms with the true classes by computing the clustering accuracy measure as

$$\mathbf{Acc} = \frac{1}{N} \max_{(C_k, L_m)} \left(\sum_{(C_k, L_m)} T(C_k, L_m) \right), \tag{30}$$

where C_k denotes the kth cluster in the final results, L_m is the true mth class, and $T(C_k, L_m)$ is the number of entities which belong to class m and are assigned to cluster k.

To obtain stable results, for each data set, we computed the average results of each algorithm over 20 runs. For comparison, we also performed the NC algorithm in the original data space (Baseline). For SR, KSR, and MKL-SR, the dimension of the subspace is the number of categories. For KPCA, we tested its performance with all the possible dimensions and report the best result. For SR, KSR, and MKL-SR, we simply set the value of the parameter γ as 1. For KSR and KPCA, the Gaussian function $\exp(-b\|\mathbf{x} - \mathbf{a}\|^2)$ with width 1 was selected. For MKL-SR, we use a linear kernel function, a polynomial kernel function, and a Gaussian kernel function.

Table 2 lists the mean of 20 different random repetitions as well as the standard deviation. From Table 2, we observe that the performance of kernel based algorithms is much better than SR, which indicates that the performance of linear DR algorithms can be improved by virtue of nonlinear kernel functions. MKL-SR significantly surpasses KSR and KPCA, which are single kernel based approaches. This is due to the fact that MKL-SR is able to learn a better kernel by MKL, which is considerably more effective than a single Gaussian kernel. The performance of KSR is very close to that of KPCA, but the number of reduced dimensions of KPCA has to be verified by testing many times. In addition to the fixed number of reduced dimensions, we also try to examine how the compared algorithms work when applying KPCA to obtain projected data of a varied number of dimensions. Thus, MKL-SR is easy to be implemented and has better performance than other algorithms.

4.2. Experiments on Supervised Learning. In this experiment, we mainly compared MKL-SR with the following approaches: KPCA, LDA, SR, and KSR. In order to evaluate the performance of these algorithms, we performed the SVM algorithm in the original face image space (baseline) and KPCA, LDA, SR, KSR, and MKL-SR subspace. The kernels and parameters are set in the same way as in the unsupervised learning. From each class of the CMU PIE face data sets, we randomly selected l (the number of training samples per class) samples for training.

For each given l, we averaged the results over 30 random splits and computed the mean as well as the standard deviation, which are listed in Table 3. As can be seen from Table 3, the performance of KPCA and LDA is even worse than that of the baseline method, which resulted from the limitation of KPCA and LDA. As is well known that KPCA is unsupervised, thus it cannot effectively exploit the supervised information, which results in the worst performance in supervised case. LDA does not utilize the regularization approach to control the model complexity. Thus, it cannot solve the over-fitting problem in small sample size case. In contrast, SR, KSR, and MKL-SR take advantage of the

Table 2: Clustering accuracy (in percent) based on different DR methods.

Data	Baseline	SR	KSR	KPCA	MKL-SR
Ionosphere	75.1 ± 0.8	80.6 ± 0.5	85.6 ± 0.4	85.2 ± 0.3	89.5 ± 0.2
Letter A-B	86.2 ± 0.6	89.4 ± 0.4	90.7 ± 0.3	91.4 ± 0.3	93.4 ± 0.3
Satellite C1-C2	95.7 ± 0.7	96.3 ± 0.2	97.3 ± 0.3	97.3 ± 0.3	98.7 ± 0.2
Digits 0689	90.3 ± 0.4	92.5 ± 0.3	93.6 ± 0.3	94.6 ± 0.2	95.6 ± 0.2
Digits 1279	93.4 ± 0.2	94.3 ± 0.2	95.7 ± 0.3	94.5 ± 0.2	96.8 ± 0.2

Table 3: Recognition accuracy rates on PIE (mean ± std-dev%).

Train size	Baseline	KPCA	LDA	SR	KSR	MKL-SR
5 × 68	68.0 ± 1.7	40.2 ± 0.3	58.2 ± 1.5	71.9 ± 1.4	72.4 ± 0.7	79.2 ± 0.8
10 × 68	83.2 ± 0.7	51.3 ± 0.3	70.3 ± 1.3	85.0 ± 1.3	87.2 ± 0.4	90.2 ± 0.5
20 × 68	91.1 ± 0.6	68.7 ± 0.3	79.5 ± 0.8	92.3 ± 0.7	94.0 ± 0.3	95.7 ± 0.3
30 × 68	93.4 ± 0.6	70.2 ± 0.4	89.1 ± 0.5	93.4 ± 0.7	95.9 ± 0.2	96.4 ± 0.3
40 × 68	94.6 ± 0.6	82.3 ± 0.4	91.8 ± 0.4	94.8 ± 0.4	96.6 ± 0.2	97.9 ± 0.2

Tikhonov regularizer to improve the smoothness of projection functions, and they can perform better than KPCA and LDA. By substituting the nonlinear embedding functions with the linear ones, KSR and MKL-SR all outperform SR. The performance of MKL-SR is better than that of KSR based on a single kernel, which indicates that MKL-SR can select an appropriate kernel and validates the effectiveness of our method.

The key parameter in MKL-SR is the regularization parameter $\gamma \geq 0$ which controls the smoothness of the embedding function based on multiple kernels. Next, we discuss the impact of parameter γ on the performance of MKL-SR. Figure 1 shows the performance of MKL-SR as a function of the parameter γ. For convenience, the X-axis is plotted as $\gamma/(1+\gamma)$ which is strictly in the interval $[0,1]$. As can be seen from Figure 1, MKL-SR obtains the best performance near the middle of the interval. When $\gamma/(1+\gamma)$ decreases to zero or increases to one, the performance of MKL-SR decreases sharply. Fortunately, good performance can be achieved over a wide range of γ, which shows that the parameter selection is not a crucial problem in MKL-SR algorithm. In reality, we can use cross validation to verify the best parameter or simply select a value between 0.1 and 1.

4.3. Experiments on Semisupervised Learning. In the semisupervised case, we compared the performance of MKL-SR with KPCA and semisupervised KSR. For comparison, we performed the SVM algorithm in the original face image space (baseline), KPCA, and semisupervised KSR and MKL-SR subspace. For KSR and MKL-SR, we simply set the value of the parameter γ as 1. In the semisupervised MKL-SR, the parameter δ $(0 < \delta \leq 1)$ was selected by cross validation. The kernels and parameters are set in the same way as in the unsupervised learning. For the extended Yale B face data set, a random subset with l $(= 5, 10, 20, 30, 40)$ images per individual was first taken to form the training set and the rest of the data set was used to be the testing set. In the training set, we only use one half data as labeled data and the rest as unlabeled data. KPCA only uses unlabeled data and the SVM algorithm is also performed on the reduced data based on KPCA. KSR and MKL-SR use both labeled and unlabeled data. The p is set to be 7 for the p-nearest neighbor graph over all the training samples in KSR and MKL-SR.

We average the classification accuracy over 30 random splits for each given l. The mean as well as the standard deviation is shown in Table 4. From Table 4, we can observe that KSR and MKL-SR can efficiently exploit both labeled and unlabeled data to discover the intrinsic geometry structure in the data; that is, the reduced data can preserve the original intrinsic geometry structure very well. Thus, they outperform the baseline method and KPCA, which cannot utilize all the available data. The performance of MKL-SR is much better than that of KSR, which indicates that the final kernel matrix learned by MKL-SR is still better than the one based on a single kernel in the semisupervised case. Overall, the proposed MKL-SR algorithm can achieve better performance in the supervised, semisupervised, and unsupervised case.

5. Conclusion

In this paper, we propose a new dimensionality reduction framework called MKL-SR. By means of SR, we solve the out-of-sample extension problem by seeking an embedding function in RKHS induced by multiple kernels. Thus, this method can not only construct the nonlinear embedding function in the form of convex combination of base kernels but also improve the performance of single kernel based SR in the supervised, semisupervised, and unsupervised case. Experimental results validate the effectiveness and efficiency of the MKL-SR algorithm. In the near future, we will further explore how to integrate different MKL methods into our model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

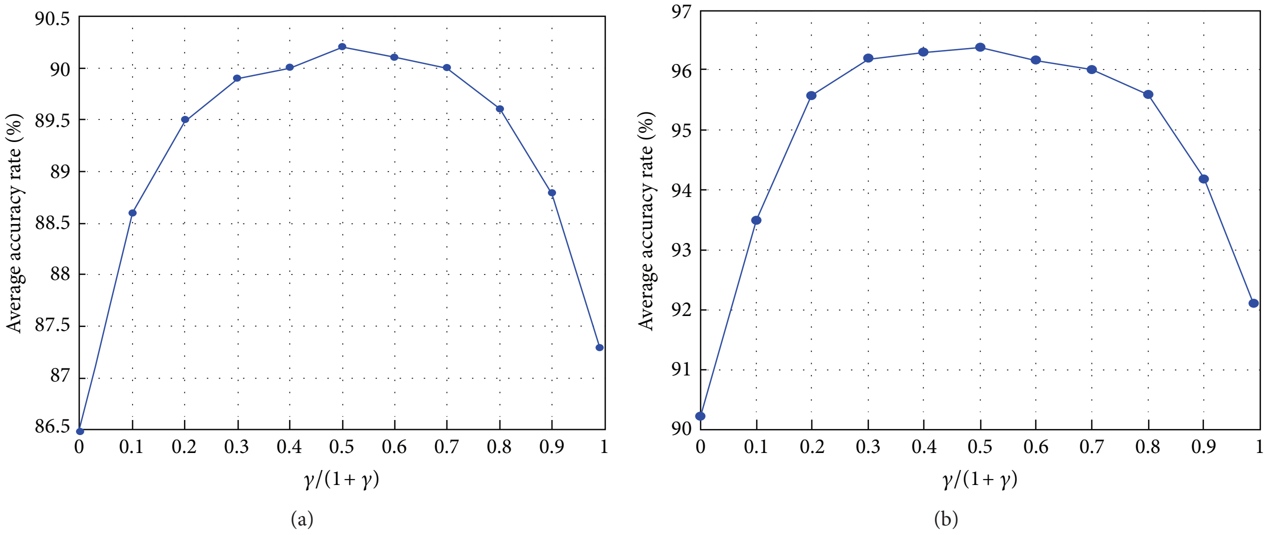

FIGURE 1: Performance of MKL-SR with respect to $\gamma/(1+\gamma)$ on PIE. (a) $l = 10$. (b) $l = 30$.

TABLE 4: Recognition accuracy rates on Extended Yale B (mean ± std-dev%).

The number of labeled samples	Baseline	KPCA	Semi-supervised KSR	Semisupervised MKL-SR
$l = 95$	50.8 ± 2.3	57.4 ± 2.0	61.5 ± 1.7	73.2 ± 1.5
$l = 190$	69.3 ± 1.4	73.7 ± 1.4	76.3 ± 1.3	80.4 ± 0.9
$l = 380$	83.2 ± 0.6	85.8 ± 0.7	89.7 ± 0.3	92.6 ± 0.3
$l = 570$	90.1 ± 0.3	92.5 ± 0.3	95.7 ± 0.2	96.3 ± 0.2
$l = 760$	91.6 ± 0.2	95.4 ± 0.3	97.2 ± 0.2	97.9 ± 0.2

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities under Grant no. 2013XK10.

References

[1] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2169–2178, usa, June 2006.

[2] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian processes for object categorization," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 169–188, 2010.

[3] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 26–33, June 2005.

[4] X. He and P. Niyogi, "Locality Preserving Projections," Advances in Neural Information Processing Systems, MIT Press, 2003.

[5] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with Gaussian processes for object categorization," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, pp. 1–8, Rio de Janeiro, Brazil, October 2007.

[6] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV 07)*, Rio de Janeiro, Brazil, October 2007.

[7] D. Cai, X. He, and J. Han, "Efficient Kernel Discriminant Analysis via spectral regression," in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM '07)*, pp. 427–432, Omaha, Neb, USA, October 2007.

[8] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, October 2007.

[9] D. Cai, X. He, W. V. Zhang, and J. Han, "Regularized locality preserving indexing via spectral regression," in *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07)*, pp. 741–750, Lisboa, Portugal, November 2007.

[10] D. Cai, X. He, and J. Han, "SRDA: an efficient algorithm for large scale discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, 2008.

[11] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[12] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the 21rst International Conference on Machine Learning (ICML '04)*, pp. 839–846, Banff, Canada, July 2004.

[13] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, vol. 33, no. 6, pp. 1147–1160, 2011.

[14] X. Zhu, Z. Huang, Y. Yang, H. T. Shen, C. Xu, and J. Luo, "Self-taught dimensionality reduction on the high-dimensional small-sized data," *Pattern Recognition*, vol. 46, no. 1, pp. 215–229, 2013.

[15] X. Zhu, Z. Huang, H. Tao Shen, J. Cheng, and C. Xu, "Dimensionality reduction by Mixed Kernel Canonical Correlation Analysis," *Pattern Recognition*, vol. 45, no. 8, pp. 3003–3016, 2012.

[16] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[17] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[18] A. A. Mohammed, R. Minhas, Q. M. Jonathan Wu, and M. A. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2588–2597, 2011.