Neural Process Lett (2017) 45:539–552
DOI 10.1007/s11063-016-9543-9

Ensemble Multiple-Kernel Based Manifold Regularization

Guo Niu[1] · **Zhengming Ma**[1] · **Shaogao Lv**[2]

Published online: 26 July 2016

Abstract As a class of semi-supervised learning methods, manifold regularization learning has recently attracted a lot of attention, due to their great success in exploiting the underlying geometric structures among data. This paper presents a novel semi-supervised approach by combining manifold regularization learning with the idea of multiple kernels, named after ensemble multiple-kernel manifold regularization learning. In our approach, multiple kernels we introduced are not only used to add the flexibility and diversity of the candidate space for the learning problem, but also act as a similarity measure to search for an optimal graph Laplacian in some sense. In other words, the proposed method allows us to learn an 'ideal' kernel and an optimal graph Laplacian simultaneously, which is of significant difference from existing methods. The associated optimization problem is solved efficiently by an alternating iteration procedure. We implement experiments over four real world data sets to demonstrate the benefits of the proposed method.

Keywords Manifold regularization learning · Multiple kernel learning · Reproducing kernel Hilbert space (RKHS) · Kernel learning

1 Introduction

Learning knowledge from labeled and unlabeled samples plays a key role in semi-supervised learning (SSL). Manifold regularization (MR) learning [1] is one of the representative and successful SSL methods which is achieved by exploiting the intrinsic geometric structure of the probability distribution of the data, and it recently has attracted a lot of attention [3,8,12–14,17]. Until recently, the researchers have extended the MR learning to many areas

✉ Guo Niu
niuguo@mail2.sysu.edu.cn

[1] The School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China

[2] Center of Statistics Research, Southwestern University of Finance and Economics, Chengdu 611130, China

of machine learning. For example, [17] and [14] relaxed the label function so that it is better to deal with novel samples and classification, and [3] aimed at solving small scale problems appearing in the consequent inverse operation, and [8] transformed the problem of manifold regularization into learning an optimal graph Laplacian.

Under the framework of MR learning, the decision function or a learner, is a linear combination of a single kernel function at the given instances and the performance of MR algorithms strongly depends on the choice of the kernel and the given data. Thus the misspecification of the kernel function may result in the distortion of the manifold structure, meanwhile it often leads to the underfitting for modeling data. Fortunately, multiple kernel learning (MKL) [2,9–11,15,20,21] has been proven useful and effective in terms of theoretical analysis and practical applications, compared to several single kernel methods. The main idea of MKL is to learn kernels with linear combinations of multiple specified kernels, so that the greater flexibility can be gained for any kernel-based method.

Among kernel-based methods [4,5,16,19], kernel function is either used as the so called kernel trick, i.e. mapping input sample into a high dimensional kernel space for learning nonlinear problem, or used as similar function for measuring the difference between two input samples. Compared to the existing work, one of significant differences is that we here make full use of the two advantages of kernel function. Based on this idea, the proposed method can produce a decision function with greater flexibility, and also find out an ideal graph Laplacian. Furthermore, the intrinsic manifold structure among data can be exploited efficiently. The MR methods utilize a graph Laplacian regularizer to constrain the decision function to be smooth with respect to the data manifold, a low-dimensional subspace on which the high-dimensional data actually resides; in application, the manifold is determined by a graph Laplacian from the given data. The graph Laplacian construction step is critical and important in MR methods but is still an open issue that has not been comprehensively studied [6,18]. In multiple manifolds or multiple view learning[22–24], the approximation of the optimal graph Laplacian draws on some pre-given manifold candidates according to different manifolds or views. In this paper, we utilize the pregiven multiple basis kernel functions to generate initial graph Laplacian candidates, and then a convex set over the constrained coefficients is constructed to search for the optimal graph Laplacian. At this point, it is a learning problem of the optimal graph Laplacian. Since the optimal coefficients in the learning problem of graph Laplacian are the same as the combination coefficients of kernels used in the decision function, we can solve the graph Laplacian optimization problem together with the learning problem of multiple kernel functions function.

In this paper, we propose a novel semi-supervised approach by combining manifold regularization with the idea of multiple kernels. In our case, a flexible learner within multi-kernel class and the optimal graph Laplacian are selected simultaneously. The advantages of our proposed method mainly consists of the following points: (1) learning both the composite manifold and the multiple kernel functions jointly; (2) the decision function is built by using multiple kernel functions for extracting more data information from data, in other words, the reproducing kernel Hilbert space associated with multiple kernels enables us to enrich and expand the search range of the optimal solution of learning problem; (3) the intrinsic manifold is further approximated by some initial manifold candidates that are induced by the multiple kernel functions. In term of theory, we present the formulations of the proposed algorithm and search for the optimal decision function in the sum space of reproducing kernel Hilbert spaces (RKHS). In term of numerical computation, the associated optimization problem is solved efficiently by an alternating iteration procedure like [15].

The paper is organized as follows. In Sect. 2, we introduce the framework of manifold regularization learning and some related concepts. Section 3 introduces our proposed approach called multiple-kernel manifold regularization (MKMR) and the details of its optimization problem. The experimental results on both synthetic and real-world data sets are presented in Sect. 4. Finally, some conclusive remarks are given in Sect. 5.

2 Manifold Regularization

In the semi-supervised learning, consider an available set $X = \{x_1, \ldots, x_{l+u}\} \subseteq \Omega, x_i \in R^n$, where $\{x_1, \ldots, x_l\}$ are the labeled samples with the corresponding labels $\{y_1, \ldots, y_l\}$ and $\{x_{l+1}, \ldots, x_{l+u}\}$ are the unlabeled samples. We suppose that $X_L = \{x_1, \ldots, x_l\}$ is the set of labeled samples drawn according to the joint distribution P defined on $\Omega \times R$, and $X_U = \{x_{1+u}, \ldots, x_{l+u}\}$ is the set of unlabeled samples drawn according to the marginal distribution P_X of P.

Manifold regularization aims at exploiting the geometry of the marginal distribution P_X. It assumes that if the two data points $x, x' \in X$ are close in the intrinsic geometry of P_X, then the conditional distributions $P(y|x)$ and $P(y|x')$ are similar accordingly. Then an additional regularization term is introduced to characterize manifold structure for the given learning problem. Consider an reproducing kernel Hilbert space (RKHS) H associated with a symmetric nonnegatively definite kernel $\{k(x, x') : \Omega \times \Omega \to R\}$, and the manifold regularization learning problem can be expressed as the following form:

$$\min_{f \in H} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(y_i, f(x_i)) + \kappa \|f\|_H^2 + \lambda \|f\|_M^2 \right\}, \tag{1}$$

where $f : \Omega \to R$ is the decision function. The first term of the objective function(1) is defined on the loss function V which measures the discrepancy between the predicted value $f(x_i)$ and the actual label y_i. $\|f\|_M^2$ is the *manifold regularizer* and measures the smoothness of the function f on data manifold and $\|f\|_H^2$ is the norm of the function f in the RKHS H. κ and λ are the regularization parameters, balancing different terms. The aim of the objective function (1) is to find the optimal function f in the RKHS space H.

In most applications, P_X is not known. Therefore, the manifold regularizer is usually approximated by the graph Laplacian matrix associated with X and the function prediction $\mathbf{f} = [f(x_1) \cdots f(x_{l+u})]^T$. Hence the optimization problem can be reformulated as:

$$\min_{f \in H} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(y_i, f(x_i)) + \kappa \|f\|_H^2 + \lambda \mathbf{f}^T L \mathbf{f} \right\}, \tag{2}$$

where L is the graph Laplacian matrix given by $L = D - W$. Here D is the diagonal matrix with $d_{ii} = \sum_{j=1}^{l+u} w_{ij}$ and W is the similar matrix, where the element w_{ij} denotes the similarity between points x_i and x_j. For example, a commonly-used measurement of w_{ij} is the Gaussian kernel function defined on the Euclidean distance [5,12], i.e., if the data $x_j \in N(x_i)$ or $x_i \in N(x_j)$, $w_{ij} = exp(-\frac{\|x_i - x_j\|^2}{t^2})$ and 0 otherwise, where $N(x_i)$ is the neighborhood of x_i and t is the tuning parameter.

Following the Representer Theorem [9], the minimizer of optimization problem (2) admits an expansion

$$f(x) = \sum_{i=1}^{l+u} \alpha_i k(x, x_i) \tag{3}$$

in term of the unlabeled and labeled samples [1]. Therefore, the decision of the kernel function k plays a key role for the performance of MR algorithms. When lose function V in Eq.(2) is adopt to be the squared loss function $V(y_i, f(x_i)) = (y_i - f(x_i))^2$, the problem of (2) can be reduced to a typical quadratic programme, and its analytic solution can be easily obtained (refer to [1] in details). And the MR algorithm formulates the Laplacian Regualrized Least Squares (LapRLS).

3 Proposed Ensemble Multiple-Kernel Manifold Regularization (MKMR)

In this section, we propose a novel semi-supervised method that combines the two characteristics of multiple kernels and the specific merit of manifold regularization.

3.1 Framework

We denote by $\{k_b(x, x') : b = 1, \ldots, m\}$ the set of m base kernels to be combined. Each base kernel k_b can generate an RKHS H_b. Under the MKL framework, $f(x) = \sum_b f_b(x)$ where each function f_b belongs to the different RKHS H_b . Therefore with a non-negative coefficient β_b, we can define an equivalent space of H_b by $H'_b = \{f_b : \frac{\|f_b\|^2_{H_b}}{\beta_b} < \infty, f_b \in H_b\}$. When $\sum_{b=1}^m \beta_b = 1$ is constrainted, a new RKHS H' is defined as the direct sum of the spaces H'_b, i.e., $H' = H'_1 \oplus \cdots \oplus H'_m$, and its reproducing kernel is the combined kernel function $k'(x, x') = \sum_{b=1}^m \beta_b k_b(x, x')$. H' is a multiple-kernel space and can be used to further explore the data and solve the problems involved in multiple data sources. The learning problem of MKMR method is working in the new RKHS H', formulated as the following minimization

$$\min_{f \in H'} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(y_i, f(x_i)) + \kappa \|f\|^2_{H'} + \lambda \|f\|^2_M \right\}, \tag{4}$$

where $\|f\|^2_{H'}$ penalizes the classifier complexities measured in the RKHS H'. Obviously, the solution range of f is expanded from the single kernel space to the multiple-kernel space.

As mentioned before, the kernel function can be used to measure the similar relationship between pair data. Therefore, in order to find an optimal graph Laplacia for manifold regularizer, we employ a series of initial graph Laplacian candidates $\{L_b\}_1^m$ corresponding to the multiple kernels $\{k_b\}_1^m$ to construct a convex set $\{L | L = \beta_1 L_1 + \cdots + \beta_m L_m, \sum_{b=1}^m \beta_b = 1\}$. Such a combination for graph Laplacian naturally allows us to learn a better graph Laplacian. Note that the initial graph candidates may be not limited to those base kernels. By using the above expression, the manifold regularizer becomes

$$\|f\|^2_M = \mathbf{f}^T \left(\sum_{b=1}^{m} \beta_b L_b \right) \mathbf{f} = \sum_{b=1}^{m} \beta_b \|f\|^2_{M_b}. \tag{5}$$

Substituting (5) into the framework (4), we have

$$\min_{\beta_b \in \Delta} \min_{f_b \in H'_b} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(y_i, f(x_i)) + \kappa \sum_{b=1}^{m} \beta_b \|f_b\|^2_{H'_b} + \lambda \sum_{b=1}^{m} \beta_b \|f\|^2_{M_b} \right\}, \tag{6}$$

where $\Delta = \{\beta_b \in R_+^m : \sum_{b=1}^m \beta_b = 1\}$ is the domain of β_b . It can be seen that we actually establish two learning problems: the learning problem of the classifier f and the optimization problem of graph Laplacian. We can also use different parameters in our learning problem, denoted by β_b and β'_b. However, by doing this, we need to estimate additional m coefficients, resulting in adding computation cost. In view of computational efficiency, we only use the single tuning parameter λ to measure the differences between the two sets of coefficients in Eq. (6), instead of the set of coefficients β'_b.

3.2 Parameters Optimization

In our current work, we adopt the group-Lasso minimization MKL method [15] as a solution to optimize the decision function of MKMR. Following the Representer Theorem [9], the solution of the optimization problem (6) admits

$$f(x) = \sum_{i=1}^{l+u} \alpha_i \sum_{b=1}^{m} \beta_b k_b(x, x_i). \tag{7}$$

Therefore, the MKMR problem only refers to the optimization of coefficients $\{\alpha_i\}_1^{l+u}$ and $\{\beta_b\}_1^m$. To solve it, we employ the alternating way with two steps: finding the optimal $\{\beta_b^*\}_1^{l+u}$ with fixed $\{\alpha_i\}_1^{l+u}$, and then finding the best classifier weights $\{\alpha_i\}_1^{l+u}$ with the fixed $\{\beta_b^*\}_1^{l+u}$. In the first step, inspired by the method [9] we can define a new classification functions set

$$\hat{f}_b = \beta_b f_b, \tag{8}$$

where $b = 1 \cdots m$. With it, the formulation in (6) is rewritten as

$$\underset{\beta\in\Delta}{min}\ \underset{\hat{f}_b\in H'_b}{min} \left\{ \frac{1}{l}\sum_{i=1}^{l} V\Big(y_i, \sum_{b=1}^{m} \hat{f}_b(x_i)\Big) + \kappa \sum_{b=1}^{m} \frac{1}{\beta_b} \|\hat{f}_b\|^2_{H'_b} + \lambda \left(\sum_{b=1}^{m} \hat{f}_b\right)^T \left(\sum_{b=1}^{m} \beta_b L_b\right)\left(\sum_{b=1}^{m} \hat{f}_b\right)\right\}. \tag{9}$$

By taking the minimization over β, we obtain:

$$\beta_b^* = \frac{\|\hat{f}_b\|^2_{H'_b}}{\sum_{b=1}^{m} \|\hat{f}_b\|^2_{H'_b}}. \tag{10}$$

When taking derivative of the Lagrangian with respect to β_b, we set

$$\sum \beta_b L_b = \sum_{j\neq b} \beta_j L_j + L_b\Big(1 - \sum_{j\neq b} \beta_j\Big), \quad j = 1 \cdots m.$$

Observing from the Eq. (10), we update the parameter β_b by the norm of the decision function. Second, with the fixed β^*, the object $J(\alpha)$ is given by

$$J(\boldsymbol{\alpha}) = \underset{\boldsymbol{\alpha}\in R^{l+u}}{min} \left\{ \frac{1}{l}\sum_{i=1}^{l} V\left(y_i, \sum_{i=1}^{l+u} \alpha_i k^{'}(x, x_i)\right) + \kappa \boldsymbol{\alpha}^T K' \boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^T K' \left(\sum_{b=1}^{m} \beta_b L_b\right) K' \boldsymbol{\alpha} \right\}, \tag{11}$$

where $\mathbf{y} = [y_1, \ldots, y_l, 0, \ldots, 0]_{1\times(l+u)}$ is the label vector; K' is the kernel matrix of the combined kernel function k'. When the squared loss is used as the loss function V, i.e., $V = (\mathbf{y} - GK'\boldsymbol{\alpha})^T(\mathbf{y} - GK'\boldsymbol{\alpha})$, we have

$$\underset{\boldsymbol{\alpha}\in R^{l+u}}{min} \left\{ \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T GK'\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \left(K^{'T} G^T G K' + \kappa K' + \lambda K^{'T} \left(\sum_{b=1}^{m} \beta_b L_b\right) K^{'} \right) \boldsymbol{\alpha} \right\}, \tag{12}$$

where G is the diagonal matrix in which the first l diagonal elements are 1 and the rest are 0. The problem above becomes a typical symmetric quadratic form, that is

$$\boldsymbol{\alpha}^* \;=\; \underset{\boldsymbol{\alpha}\in R^{l+u}}{argmin}\left\{\mathbf{y}^T\mathbf{y} - 2B^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T A\boldsymbol{\alpha}\right\} = A^{-1}B, \tag{13}$$

where $B = K'^T G^T \mathbf{y}$, $A = K'^T G^T G K' + \kappa K' + \lambda K'^T\left(\sum_{b=1}^m \beta_b L_b\right)K'$. Note that the matrix A is a symmetric and positive definite matrix, and then the final analytic solution is

$$\boldsymbol{\alpha}^* = \left(K'^T\left(G^T G + \kappa I + \lambda\left(\sum_{b=1}^{m}\beta_b L_b\right)\right)K'\right)^{-1} K' G^T \mathbf{y}, \tag{14}$$

where I is the unit matrix. Repeat the above two steps until meeting the stopping criteria of the alternating optimization procedure.

4 Experiments

This experimental study aims at verifying the effectiveness and advantages of MKMR. The proposed approach introduces the multiple kernels to the MR approach and jointly learns the optimal graph Laplacian. In order to validate the proposed MKMR, the classical MR [1] and the general multi-kernel based MR approach (be called MMR in short) [7] are considered for comparisons with the proposed MKMR. This study includes the two parts: first, we focus on MR, MMR, and MKMR for classification problem with four real-world datasets; second, we compare MKMR with MMR and an optimized MMR method (be called OMMR in short) to show the effectiveness of learning the optimal graph Laplacian. OMMR method uses a pre-optimized graph Laplacian obtained by the cross-validation method.

4.1 Data Sets

In our experiments, we use one object database(ETH80), one handwritten digits database(USPS), one face database(Yale-B), and the Caltech 101 dataset. These datasets are chosen since they correspond to several types.

The ETH-80[1] dataset has been used in many object recognition studies. It contains eight categories (apples, pears, tomatoes, cows, dogs, horses, cups, and cars) of images. Each class has 10 objects, and each object is represented by 41 views. Thus there are 3,280 images in total in this dataset. The size of image is 16 × 16.

The USPS[2] handwritten digits database contains 10 handwritten digits from “0” to “9”, and each digit consists of 1100 grayscale images. The images we loaded have been resized to 16 × 16 pixels. The size is already small.

The Caltech 101[3] dataset contains 9146 images of 101 categories (about 40 to 800 images per category). We resize all the images and the final size is 150 × 150. This dataset object displays a wide variety of complex geometry and reflectance characteristic.

The Yale-B[4] face dataset contains 10 individuals, and each is seen under 576 viewing conditions (9 poses and 64 illumination conditions). All images are cropped based on the

[1] http://people.csail.mit.edu/jjl/libpmk/samples/eth.html.

[2] http://www.cs.nyu.edu/~roweis/data.html.

[3] http://www.vision.caltech.edu/Image_Datasets/Caltech101/.

[4] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html.

location of eyes, and resize to 32 × 32 pixels. We select a subset, including 64 images of 8 individuals (under the front pose) to evaluate the performance of algorithms.

4.2 Parameters Settings

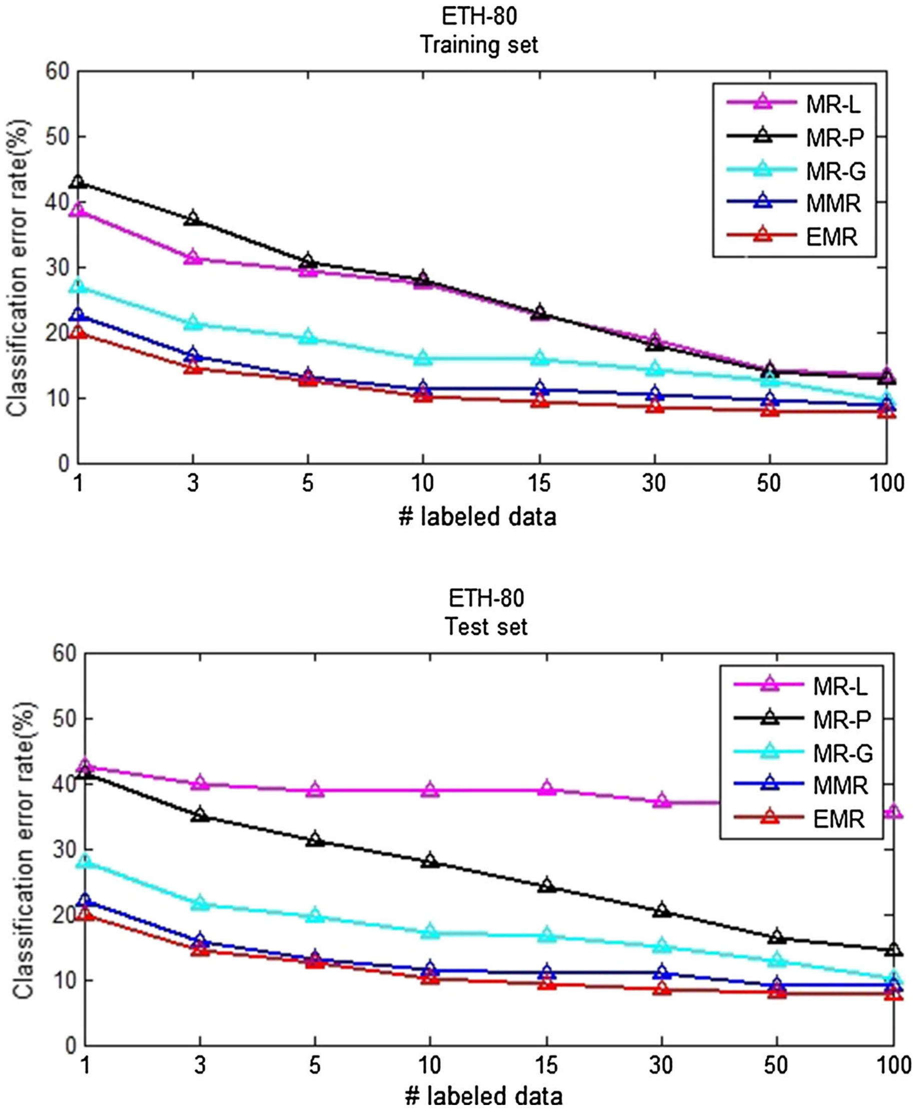
There are three parameters (the number of nearest neighbors, and the two regularization parameters κ and λ) to be tuned for all comparison methods. For all experiments, the number of nearest neighbors is set to 6. and the two regularization parameters are selected from set $\{10^{-4}, 10^{-2}, 10^{-1}, 1, 10, 100\}$ through a twofold cross-validation of labeled training samples on the training set. We use the Gaussian RBF kernel $k(x, v) = exp(-\frac{\|x-v\|^2}{t^2})$ with 10 different widths, i.e., $t = \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^2, 2^3, 2^4, 2^5, 2^7\} * \theta$, and the Polynomial kernel of degree 1 to 3 to construct base kernels, where θ is set to the average of squared distances in the training set. During the training procedure of MR, we calculated the kernel matrix with linear one-order polynomial, two-order polynomial and a Gaussian kernel, respectively. For MKMR, MMR, and OMMR, the maximum number of iterations as the stopping criteria is set to 15 for all datasets.

The entire classification proceeds on the four datasets include a binary classification problem and a multiple-class problem. The multiple-class problem can be converted into a set of binary classification problems via the one-vs-rest strategy. For the binary classification (two-class) experiments, 60 % of the data per class is used as the training set and the remaining 40 % as the test set. The number of labeled samples l for ETH-80 and USPS datasets is set to 1, 3, 5, 10, 15, 30, 50, and 100, respectively, and for Yale-B and Caltech 101 datasets it is set to 1, 3, 5, 8, 10, and 15, respectively. It is a critical variable in semisupervised learning algorithms. Each two-class experiment is repeated ten times, and the average classification error rates are reported. For the multiclass experiments, we vary the number of training data. For ETH80 and USPS datasets, we randomly select 20, 30, 40, 50, 80, 100, and 120 data from each class to form the training set, and use the remaining data as the test set. For Caltech 101 and Yale-B datasets, the number of the data in most cases is less than 60, then we randomly choose 20, 30, and, 40 data per class to form the training set, the remaining data serve as the test set. At the same time, l is set to 5. The multiclass process is run on the same set of 10 randomly drawn replication of the training and test set. Then the averaged error rates and the corresponding standard deviation are reported. The running time of MKMR, MMR, and OMMR in multiclass experiments are presented for discussing the effectiveness of MKMR. All the kernel matrices are pre-computed and loaded into the memory. This enables us to avoid re-computing and loading kernel matrices at each iteration of optimization. Our experiments are implemented in a 64 b Windows PC with 2.3 GHz CPU and 4 GB RAM. All the algorithms are implemented with MATLAB.

4.3 Classification Results

The binary classification results of five methods (MR-L, MR-P, MR-G, MMR, and MKMR) for ETH-80, USPS, Caltech 101, and Yale-B datasets are shown in Figs. 1, 2, 3, and 4, respectively. MR-L, MR-P, and MR-G represent the MR method with a linear kernel, a polynomial kernel, and a Gaussian kernel, respectively. For MR and MMR, the graph Laplacian matrix is estimated by an empirical value. In this experiment, we use the average of squared distances of training data θ to compute the graph Laplacian for MMR and MR.

From the Fig. 1, we see that our MKMR method has got the smallest classification error rates on both the test and unlabeled training samples. Moreover, MKMR performs much better than MR methods (MR-L, MR-P, and MR-G). For MR, in most case, polynomial kernel gives



ETH-80
Training set

ETH-80
Test set

Fig. 1 Binary classification results (%) of five methods on the ETH-80 data set

better performance compared to linear kernel because of its nonlinearity. We also observe that the performance of MKMR is better than MMR, that indicates the effectiveness of MMR for applying the two benefits of kernel function. Fig. 2 shows that the proposed MKMR method provides more stable and effective classification results. From this figure, we see that MKMR performs much better than MR-L, MR-P, and MR-G, particularly for a small amount of labeled samples. The proposed MKMR works better than MMR consistently along with different number of labeled data. In Fig. 3, our proposed MKMR has got the best classification results on both the test and unlabeled training samples. Moreover, the improvement obtained by the proposed MKMR method on unlabeled data is significant compared with the MR methods (MR-L, MR-P, and MR-G) and the MMR approach. For Caltech 101 dataset, MMR

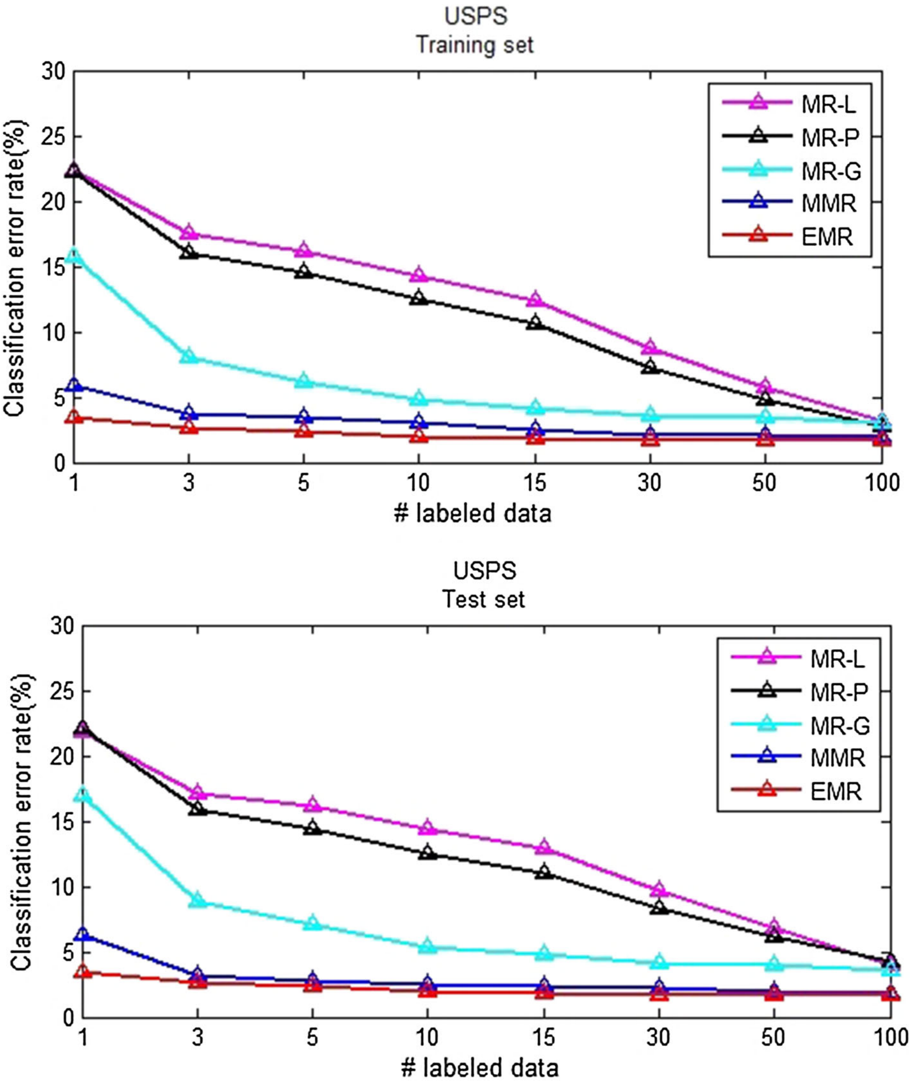


Fig. 2 Binary classification results (%) of five methods on the USPS data set

performs a little worse than MR-G on the classification of test data when $l = 1$. The reason may be that when the number of training examples is small, it may be insufficient to determine the appropriate kernel combination. For face dataset Yale-B, the binary classification results of MKMR are better than the MR methods, particularly for the test data (see Fig. 4). MKMR also performs better than MMR. Moreover, the MKMR becomes more effective as the number of labeled examples increase.

The overall multiclass classification results obtained by the proposed MKMR and the comparison methods are listed in Tables 1, 2, 3 and 4.

Table 1 summarizes the multiclass classification results of five methods (MR-L, MR-P, MR-G, MMR, and MKMR) on the ETH-80 dataset. First, we observe that the MKMR algorithm outperforms the MR algorithms on both the training and test data. In particular,

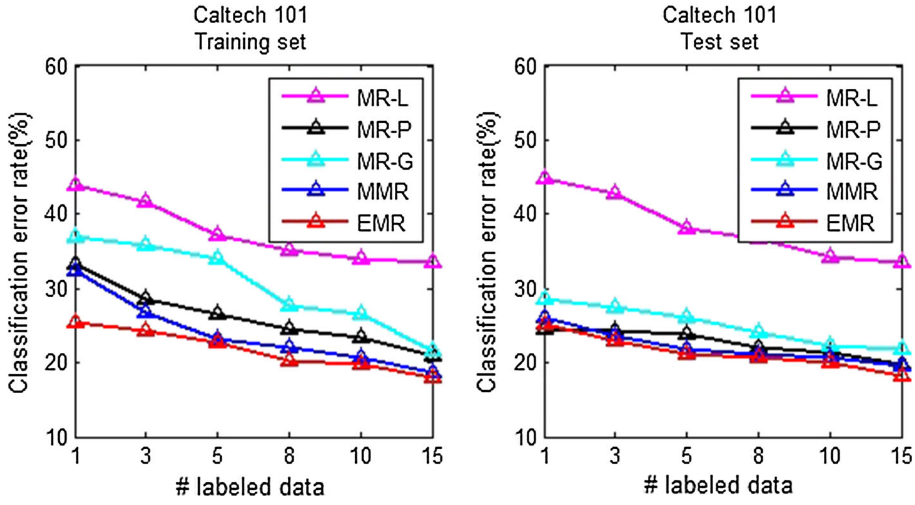


Fig. 3 Binary classification results (%) of five methods on the Caltech 101 data set

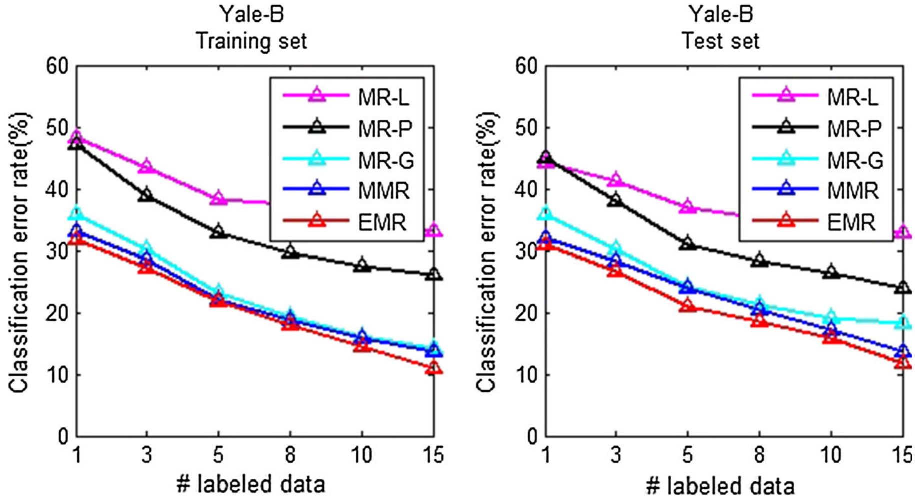


Fig. 4 Binary classification results (%) of five methods on the Yale-B data set

the proposed MKMR can obtain about 5–15 % improvements to the MR algorithms in object recognition of the ETH-80 dataset. We also observe that MKMR performs better than MMR. Moreover, MKMR becomes more effective as the number of training data increases.

The multiple-class experiments results of five comparison methods for the USPS dataset are reported in Table 2. A good improvement can be observed. First, we observe that the MKMR method outperforms the MR methods on both the training and test samples. Particularly, MKMR obtains about 5–10 % improvements to MR in the handwritten digit classification of the USPS dataset. Compared to the MMR approach, the MKMR algorithm has about 1–4 % improvements.

Table 3 shows the classification error results of five methods on the Caltech 101 dataset. Again, we see that the MKMR algorithm outperforms the MR algorithms and obtains about

Table 1 Multiclass experimental results (%) on the ETH-80 data set using five methods

No. training samples per class	ETH-80	MR-L	MR-P	MR-G	MMR	MKMR
20	U	60.34 ± 1.73	48.34 ± 1.68	38.26 ± 1.33	30.67 ± 1.99	27.78 ± 0.75
	T	45.50 ± 1.45	41.03 ± 1.16	39.68 ± 1.77	31.45 ± 1.87	25.42 ± 1.07
30	U	51.74 ± 1.56	40.74 ± 1.43	35.76 ± 1.17	29.31 ± 0.86	26.20 ± 0.70
	T	43.15 ± 1.39	36.48 ± 1.33	34.27 ± 1.63	29.63 ± 1.82	23.76 ± 0.72
50	U	45.69 ± 1.79	40.69 ± 1.12	29.20± 1.51	27.41 ± 0.95	23.86 ± 0.68
	T	38.54 ± 1.74	33.05 ± 1.56	29.72 ± 1.92	26.72 ± 1.23	22.93 ± 0.66
80	U	40.55 ± 1.90	38.52 ± 1.69	28.52 ± 1.52	25.05 ± 0.70	21.43 ± 0.98
	T	35.84 ± 1.85	34.83 ± 1.33	25.06 ± 1.45	26.11 ± 0.67	21.05 ± 0.88
100	U	37.62 ± 1.66	32.03 ± 1.52	27.75 ± 1.84	23.91 ± 0.62	20.04 ± 0.63
	T	34.73 ± 1.39	34.09 ± 1.68	23.59 ± 1.63	22.75 ± 0.60	20.14 ± 0.52
120	U	37.62 ± 1.81	30.81 ± 1.89	27.09 ± 1.90	20.19 ± 0.70	18.12 ± 0.84
	T	33.86 ± 1.62	32.68 ± 1.10	22.51 ± 1.72	22.04 ± 0.70	19.74 ± 0.88

U represents the unlabeled training data, *T* represents the test data

Table 2 Multiclass experimental results (%) on the USPS data set using five methods

No. training samples per class	USPS	MR-L	MR-P	MR-G	MMR	MKMR
20	U	41.44 ± 0.58	40.94 ± 0.29	33.76 ± 0.79	29.80 ± 0.70	26.50 ± 0.38
	T	46.02 ± 0.58	32.69 ± 0.91	32.69 ± 0.67	30.25 ± 0.57	27.73 ± 0.76
30	U	38.51 ± 0.69	35.15 ± 0.77	28.21 ± 0.76	26.26 ± 0.45	25.14 ± 0.46
	T	45.02 ± 0.74	28.90 ± 0.67	29.60 ± 0.75	28.99 ± 0.31	24.80 ± 0.48
50	U	37.70 ± 0.81	33.25 ± 0.56	27.12 ± 0.66	24.96 ± 0.71	21.29 ± 0.32
	T	43.60 ± 0.75	29.69 ± 0.91	25.79 ± 0.7	22.29 ± 0.79	20.04 ± 0.41
80	U	35.82 ± 0.96	32.25 ± 0.75	23.46 ± 0.61	20.21 ± 0.56	19.36 ± 0.29
	T	37.40 ± 0.84	25.90 ± 0.67	24.70 ± 0.81	21.40 ± 0.51	16.69 ± 0.40
100	U	35.80 ± 0.78	30.62 ± 0.69	21.88 ± 0.84	20.06 ± 0.70	17.19 ± 0.32
	T	33.06 ± 0.97	24.37 ± 0.63	21.49 ± 0.59	17.18 ± 0.74	13.89 ± 0.48
120	U	34.94 ± 0.82	30.46 ± 0.71	20.95 ± 0.57	18.13 ± 0.51	12.31 ± 0.29
	T	32.69 ± 0.90	23.51 ± 0.63	16.82 ± 0.54	14.63 ± 0.44	13.11 ± 0.37

U represents the unlabeled training data, *T* represents the test data

7–30 % improvements to them. Moreover, our proposed algorithm has about 1–5 % improvements to the MMR approach. The competitive results of our method indicate the good generalization to the multiple object dataset.

For the face dataset Yale-B, the multiple-class classification error rates of five methods are listed in Table 4. As can been see, the MKMR method outperforms the MR methods on both the training and test data and obtains about 1–18 % improvements to them. Moreover, the proposed MKMR performs better than MMR.

Table 3 Multiclass experimental results (%) on the Caltech 101 data set using five methods

Caltech 101	The number of training samples per class					
	20		30		40	
Method	U	T	U	T	U	T
MR-L	62.31 ± 2.19	55.55 ± 2.25	49.18 ± 2.19	41.09 ± 1.76	39.51 ± 2.51	35.64 ± 1.98
MR-P	38.14 ± 1.51	35.74 ± 2.17	33.80 ± 1.91	30.47 ± 1.80	29.95 ± 1.23	27.34 ± 1.43
MR-G	40.39 ± 2.14	36.65 ± 1.54	34.11 ± 1.72	31.37 ± 1.63	31.02 ± 1.09	30.65 ± 1.54
MMR	33.84 ± 0.93	31.76 ± 1.01	30.41 ± 0.86	29.05 ± 0.66	28.50 ± 0.95	26.71 ± 0.67
MKMR	32.13 ± 0.89	29.53 ± 0.62	28.56 ± 0.57	24.34 ± 0.51	26.87 ± 0.49	25.21 ± 0.37

U represents the unlabeled training data, *T* represents the test data

Table 4 Multiclass experimental results (%) on the Yale-B data set using five methods

Yale-B	The number of training samples per class					
	20		30		40	
Method	U	T	U	T	U	T
MR-L	31.31 ± 0.63	27.55 ± 0.68	25.88 ± 0.53	27.09 ± 0.56	22.66 ± 0.46	24.36 ± 0.76
MR-P	20.36 ± 0.69	18.34 ± 0.74	18.80 ± 0.51	16.63 ± 0.60	17.57 ± 0.81	15.50 ± 0.52
MR-G	15.49 ± 0.44	14.86 ± 0.63	13.86 ± 0.71	12.48 ± 0.54	12.20 ± 0.50	11.51 ± 0.60
MMR	13.88 ± 0.53	13.76 ± 0.70	11.77 ± 0.66	12.05 ± 0.58	10.84 ± 0.23	11.17 ± 0.29
MKMR	12.29 ± 0.49	12.13 ± 0.62	10.36 ± 0.37	11.35 ± 0.51	9.92 ± 0.50	10.35 ± 0.42

U represents the unlabeled training data, *T* represents the test data

4.4 Running Time

In the second experiments, we evaluate the training time and the error rates of MMR, OMMR, and MKMR on four datasets. We empirically use θ to compute the graph Laplacian of MMR. By the cross-validation method, the graph Laplacian of OMMR is selected from the Laplacian candidates set used in MKMR. For ETH-80 and USPS sets, the number of training data per class is set to 120, and for Caltech 101 and Yale-B sets it is set to 40. We only report the multiple-class results and training time of MMR, OMMR, and MMR on the training data. Table 5 summarizes the classification error rates and time of different methods. We observe that compared to the proposed MKMR, OMMR achieves similar classification performance but requires a considerable time cost. This is not surprising because OMMR applies the cross-validation method to decide the optimal graph Laplacian matrix. At the same time, we observe that although the training time of MMR is similar to MKMR, the performance MMR becomes worse.

5 Conclusion

Manifold regularization learning is a successful SSL approach in machine learning. From the viewpoint of kernel choice, we introduce multiple kernels to improve the original MR learning. In the proposed MKMR, the reproducing kernel Hilbert space associated with multiple

Table 5 Comparison of classification error rate (%) and training time (s) among MMR, oMMR, and MKMR on the training set

Data set	Method	Error rate (%)	Time(s)
ETH-80	n = 120		
	MMR	20.19 ± 0.70	19.27 ± 3.15
	OMMR	19.35 ± 0.58	213.07 ± 14.24
	MKMR	18.12 ± 0.84	24.20 ± 4.29
USPS	n = 120		
	MMR	18.13 ± 0.51	14.69 ± 4.06
	OMMR	11.88 ± 0.38	158.85 ± 15.15
	MKMR	12.31 ± 0.29	17.27 ± 5.91
Caltech 101	n = 40		
	MMR	28.50 ± 0.95	12.62 ± 3.34
	OMMR	26.43 ± 0.58	124.37 ± 11.09
	MKMR	26.87 ± 0.49	14.58 ± 4.79
Yale-B	n = 40		
	MMR	10.84 ± 0.23	0.13 ± 0.05
	OMMR	10.17 ± 0.38	1.45 ± 0.23
	MKMR	9.92 ± 0.50	0.17 ± 0.08

Here, n is the number of training data per class

kernels contains multiscale structures, such as functional complexity and the measurement of the graph Laplacian. Therefore, the MKMR learning has comparable performance compared to the classical MR learning, in term of exploiting the intrinsic geometric structure of the data. We also implement several real-data experiments to show this improvement.

Acknowledgements We thank the editor and two anonymous reviewers for their helpful comments. This work is supported in part by the Guangdong Provincial Science and Technology Major Projects of China under Grant 67000-42020009. The coauthor Lv's research is supported partially by National Natural Science Foundation of China (Grant No.11301421), and Fundamental Research Funds for the Central Universities of China (Grants No. JBK120509 and JBK140507), as well as KLAS-130026507.

References

1. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
2. Bucak S, Jin R, Jain A (2014) Multiple kernel learning for visual object recognition: a review. IEEE Trans Pattern Anal Mach Intell 39(7):1354–1369
3. Chen L, Tsan IW, Xu D (2012) Laplacian embedded regression for scalable manifold regularization. IEEE Trans Neural Netw Learn Syst 33(6):902–915
4. Chapelle O, Weston J, Schokopf B (2003) Cluster kernels for semi-supervised learning. In: Proceedings of the advances in neural information processing system 15
5. Chen S, Chris H, Ding Q, Luo B (2015) Similarity learning of manifold data. IEEE Trans Cybern 45(9):1744–1756
6. Dornaika F, El Traboulsi Y (2016) Learning flexible graph-based semi-supervised embedding. IEEE Trans Neural Netw Learn Syst 46(1):206–218
7. Fu D, Yang T (2013) Manifold regularization multiple kernel learning machine for classification. In: Proceedings of the 2013 international conference on machine learning and cybernetics

8. Geng B, Tao DC, Xu C, Yang LJ, Hua S (2012) Ensemble manifold regularization. IEEE Trans Pattern Anal Mach Intell 34(6):1227–1233
9. Gnen M, Alpaydin E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
10. Han Y, Yang K, Ma YL, Liu GZ (2014) Localized multiple kernel learning via sample-wise alternating optimization. IEEE Trans Cybern 44(1):137–138
11. Lanckriet G, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5:27–72
12. Luo Y, Tao DC, Geng B, Xu C, Maybank SJ (2013) Manifold regularized multitask learning for semi-supervised multilabel image classification. IEEE Trans Image Process 22(2):523–536
13. Liu W, Wang J, Chang SF (2012) Robust and scalable graph-based semisupervised learning. Proc IEEE 100(9):2624–2638
14. Nie F, Xu D, Tsang IW, Zhang CS (2010) Flexible manifold embedding a framework for semi-supervised and unsupervised dimension reduction. IEEE Trans Image Process 19(7):1921–1932
15. Rakotomamonjy A, Bach F, Canu S, Grandvalet Y (2008) SimpleMKL. J Mach Learn Res 9:2491–2521
16. Scholkopf B, Smola AJ (2002) Learning with kernel. MIT press, Cambrige
17. Sindhwani V, Niyogi P, Belkin M (2005) Linear manifold regularization for large scale semi-supervised learning. In: Proceedings of the 22nd international conference on machine learning
18. Wang G, Wang F, Chen T, Yeung D, Lochovsky F (2011) Solution path for manifold regularized semisupervised classification. IEEE Trans Syst Man Cybern B Cybern 42(2):308–319
19. Xu J, Paiva ARC, Park Il (Memming), Principe JC (2008) A reproducing kernel Hilbert space framework for information-theoretic learning. IEEE Trans Signal Process 56(12):5891–5902
20. Xu Y, Chen DR, Li HX, Liu L (2013) Least square regularized regression in sum space. IEEE Trans Neural Netw Learn Syst 24(4):635–646
21. Xu Z, Jin R, Zhu SH, Lyu M, King I (2010) Smooth optimization for effective multiple kernel learning. In: Proceedings of the 24nd AAAI conference on artificial intelligence
22. Yu J, Rui Y, TaoJ DC, Yu Y Rui, Tao DC (2014) High-order distance based multiview stochastic learning in image classification. IEEE Trans Cybern 44(12):2431–2442
23. Yu J, Rui Y, Tao DC (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
24. Yu J, Rui Y, Chen B (2014) Exploiting click constraints and multiview features for image reranking. IEEE Trans Multimed 16(1):159–168