Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Multi-Kernel Low-Rank Dictionary Pair Learning for Multiple Features Based Image Classification

Xiaoke Zhu,[1,4] **Xiao-Yuan Jing,**[*,1,2] **Fei Wu,**[2] **Di Wu,**[1] **Li Cheng,**[1] **Sen Li,**[1] **Ruimin Hu**[1,3]
[1]State Key Laboratory of Software Engineering, School of Computer, Wuhan University, China
[2]College of Automation, Nanjing University of Posts and Telecommunications, China
[3]National Engineering Research Center for Multimedia Software, Computer School of Wuhan University
[4] School of Computer and Information Engineering, Henan University, China. [*]Corresponding author.

Abstract

Dictionary learning (DL) is an effective feature learning technique, and has led to interesting results in many classification tasks. Recently, by combining DL with multiple kernel learning (which is a crucial and effective technique for combining different feature representation information), a few multi-kernel DL methods have been presented to solve the multiple feature representations based classification problem. However, how to improve the representation capability and discriminability of multi-kernel dictionary has not been well studied. In this paper, we propose a novel multi-kernel DL approach, named multi-kernel low-rank dictionary pair learning (MKLDPL). Specifically, MKLDPL jointly learns a kernel synthesis dictionary and a kernel analysis dictionary by exploiting the class label information. The learned synthesis and analysis dictionaries work together to implement the coding and reconstruction of samples in the kernel space. To enhance the discriminability of the learned multi-kernel dictionaries, MKLDPL imposes the low-rank regularization on the analysis dictionary, which can make samples from the same class have similar representations. We apply MKLDPL for multiple features based image classification task. Experimental results demonstrate the effectiveness of the proposed approach.

Introduction

Learning effective features plays a crucial role in image classification tasks. Dictionary learning (DL) is an important feature learning technique with state-of-the-art classification performance (Yang et al. 2016). Most of existing DL methods focus on solving single feature representation based learning problems (Van Nguyen et al. 2013). Some popular single feature representation based DL methods include fisher discrimination dictionary learning (FDDL) (Yang et al. 2011), label consistent K-SVD (Jiang, Lin, and Davis 2013), projective dictionary pair learning (DPL) (Gu et al. 2014) and kernelized supervised DL (Gangeh, Ghodsi, and Kamel 2013).

Since more information exists in multiple feature representations than in a single one, multiple feature representations based learning techniques have attracted a lot of research interests (Xu, Tao, and Xu 2013). Nowadays, several linear multiple feature representations based DL methods have been presented. (Shi et al. 2013) presents a multimodal sparse representation based classification method for classifying lung needle biopsy images, which aims to select the topmost discriminative samples for each individual modality as well as to guarantee the large diversity among different modalities. By designing uncorrelated constraint, uncorrelated multi-view discrimination DL (UMD^{2}L) method (Jing et al. 2014) jointly learns multiple uncorrelated discriminative dictionaries from multiple views. (Wu et al. 2016) presents a multi-view low-rank dictionary learning (MLDL) approach to cope with the situation where exist large noises. This family of methods learn the dictionary by regarding the feature space of samples to be linear for each representation; however, in many practical applications, samples usually lie on a non-linear feature space.

Kernel technology is an effective way to deal with non-linear data (Zhang et al. 2012). Multiple kernel learning (MKL) has been widely applied to problems involving data with multiple feature representations (Bucak, Jin, and Jain 2014; Liu et al. 2014). Recently, a few multiple kernel sparse representation or DL methods have been presented. With the predefined dictionary, MKL for sparse representation based classification (MKL-SRC) method (Shrivastava, Patel, and Chellappa 2014) uses a two-step training strategy to learn kernel weights and sparse codes. Multiple instance DL (MIDL) method (Shrivastava, Pillai, and Patel 2015) formulates multiple instance learning problem as a kernel learning problem, and separately learns kernel dictionaries for positive and negative bags. Discriminative multiple kernel DL (DMKDL) method (Thiagarajan, Ramamurthy, and Spanias 2014) performs DL by using multiple levels of 1-D subspace clustering in kernel space, and optimizes weights of the ensemble kernel based on graph-embedding principles.

Motivation

Although improved performance has been reported in the existing multi-kernel DL methods (Thiagarajan, Ramamurthy, and Spanias 2014; Shrivastava, Pillai, and Patel 2015), there still remains several critical issues.

(1) Researches in (Gu et al. 2014; Yang et al. 2016) demonstrate that analysis-synthesis dictionary could provide a more complete view of data representation than analysis dictionary or synthesis dictionary. However, existing multi-kernel DL methods only investigate the kernel representa-

tion of data from the viewpoint of synthesis dictionary.

(2) Existing multi-kernel DL methods do not make full use of class label information in DL process. MIDL learns a positive dictionary and a negative dictionary by employing the bag label information (positive or negative), and does not consider collaboratively representing samples by dictionary bases from all classes. DMKDL utilizes class label information for the learning of kernel weights, rather than for the learning of kernel dictionary, which will directly influence the discriminative ability of the learned dictionary.

Motivated by the above analysis, we intend to solve the multiple feature representations based image classification task by learning kernel dictionary from the viewpoint of analysis-synthesis dictionary and exploiting the discriminative information contained in class label more effectively.

Contribution

The major contributions of this paper are summarized below.

(1) We propose a multiple kernel low-rank dictionary pair learning (MKLDPL) approach, and apply it to multiple feature representations based image classification. MKLDPL learns a pair of kernel synthesis and analysis dictionaries for each class. To the best of our knowledge, it is the first time to integrate multiple kernel learning and analysis-synthesis dictionary pair learning into a unified model.

(2) We design a low-rank regularization term, which requires that the learned analysis dictionary for each class should be low-rank, and therefore the obtained coding coefficients of samples from the same class are low-rank. This means that samples from the same class can have similar representations by using the learned analysis dictionary, which is beneficial to the following classification. To the best of our knowledge, low-rank technique has never been employed in analysis-synthesis dictionary pair learning.

(3) We design a structured discriminant term, which requires that each pair of class-specific kernel synthesis and analysis dictionaries should have good representation ability to samples from the associated class, but poor representation ability to samples from other classes.

Related Work

In this section, we briefly review the related multiple kernel dictionary learning methods including DMKDL (Thiagarajan, Ramamurthy, and Spanias 2014) and MIDL (Shrivastava, Pillai, and Patel 2015), then provide a discussion between our approach and related methods.

DMKDL - DMKDL incorporates the multilevel DL algorithm (Thiagarajan, Ramamurthy, and Spanias 2015) into MKL framework. In DMKDL, the ensemble kernel matrix is computed as $K = \sum_{b=1}^{H} \beta_b K_b$, where H is the number of feature representations, K_b is the kernel matrix corresponding to the b^{th} feature representation, β_b is the weight corresponding to K_b. $\beta = [\beta_1, \cdots, \beta_H]$ can be estimated by:

$$\begin{aligned} &\min_{\beta} \beta^T S_W^U \beta \\ &s.t.\ \beta^T S_{W'}^U \beta = 1, \beta \geq 0 \end{aligned} \tag{1}$$

where S_W^U and $S_{W'}^U$ are scatter matrices in the kernel space weighted by elements of intra-class and inter-class affinity matrices, respectively. With the obtained kernel matrix, DMKDL learns dictionary with the multilevel DL algorithm.

MIDL - MIDL aims to learn kernel dictionaries separately for positive and negative bags. Assume that Y_n (Y_p) is the concatenation of negative (positive) bags, the reconstruction error corresponding to negative bags is defined as: $\|\Phi(Y_n) - \Phi(Y_n)A_n X_n\|_F^2$. Here, $\Phi()$ is a nonlinear mapping function, A_n denotes the matrix corresponding to negative dictionary, and X_n is the coding coefficient matrix corresponding to $\Phi(Y_n)$. The reconstruction error corresponding to positive bags is defined in a similar manner. In addition, MIDL requires the positive bags to be orthogonal to negative dictionary:

$$\left\| A_n^T K\left(Y_n, Y_p\right) \Omega \right\|_F^2 \tag{2}$$

where Ω is a positive sample selection matrix, $K(Y_n, Y_p) = \sum_{b=1}^{H} \beta_b K_b\left(Y_n, Y_p\right)$.

Comparison with Related Methods - The major differences between our approach and the related multi-kernel dictionary learning methods (including DMKDL and MIDL) are three-folds. **Firstly**, the manners of learning multi-kernel dictionary are different. Specifically, DMKDL learns a kernel synthesis dictionary, and MIDL learns a positive dictionary and a negative dictionary, while our MKLDPL approach learns a structured kernel synthesis and analysis dictionary pair. **Secondly**, the manners of using class label information are different. In particular, DMKDL uses label information for kernel weights learning, and MIDL utilizes the bag label information (positive or negative) to learn the dictionaries. Different from them, MKLDPL uses the class label information for the learning of class-specific synthesis and analysis sub-dictionary pair. **Thirdly**, existing multi-kernel dictionary learning methods make no constraint on the relationship between the coding coefficients of samples from the same class, while our approach can ensure that the coding coefficients of samples from the same class are low-rank, which is beneficial to the subsequent classification.

Multi-kernel Low-rank Dictionary Pair Learning

Problem Formulation

In multiple feature representations based image classification task, each sample is represented by multiple kinds of features. Let $X = [X_1, ..., X_i, ..., X_C]$ be a set of N training images, where X_i represents images from the i^{th} class, and C is the number of classes. Assume that there are L kinds of features for training images X, we denote the j^{th} kind of feature representation for X and X_i by Y_j and $Y_{i,j}$, respectively. In this paper, we aim to learn discriminative dictionaries from multiple feature representations.

In practice, samples usually lie on non-linear feature space, which means that dictionaries learned in a linear way cannot well characterize the corresponding feature space. Kernel technique is an effective way to cope with this issue. By embedding multiple feature representations into Reproducing Kernel Hilbert Space (RKHS), dictionary learning

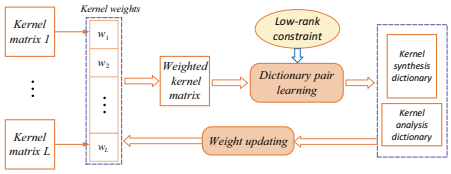


Figure 1: Basic framework of our approach.

can be conducted in RKHS. Let $\Phi_j(.)$ represent the kernel mapping function for the j^{th} feature type, and $\mathcal{K}_j(Y_j, Y_j) = \Phi_j(Y_j)^T\Phi_j(Y_j)$ denotes the kernel Gram matrix of the j^{th} feature representation. To make full use of the discriminative information contained in multiple feature representations, a commonly used way is to combine multiple base kernels into a weighted kernel. Assume that $w = [w_1, ..., w_j, ..., w_L]^T$ represents the weights for L base kernels, the weighted kernel can be computed as follows:

$$\mathcal{K}(X, X) = \sum_{j=1}^{L} w_j \mathcal{K}_j(Y_j, Y_j). \tag{3}$$

where $w_j \geq 0$ and $\sum_{j=1}^{L} w_j = 1$. Denote by $\Phi(X)$ the kernel sample set in the weighted RKHS space, i.e., $\Phi(X)^T\Phi(X) = \mathcal{K}(X, X)$.

Since analysis-synthesis dictionary pair can provide a more complete view of data representation, we introduce dictionary pair learning into the multiple kernel learning, and propose a novel kernel dictionary learning framework, with which a pair of structured kernel synthesis and analysis dictionaries can be learned. The learned structured synthesis dictionary is denoted by $\Phi(X)D$, where $D = [D_1, ..., D_i, ..., D_C]$, and $D_i \in \mathbb{R}^{N\times m_i}$. Here, m_i is the is number of atoms in D_i. Similarly, the learned structured analysis dictionary is denoted by $P\Phi(X)^T$, where $P = [P_1; ...; P_i; ...; P_C]$, and $P_i \in \mathbb{R}^{m_i\times N}$.

The learned kernel synthesis and analysis dictionaries are used for image classification task, and thus should own favorable discriminability. To this end, we design a discriminant term by using the class label information, which requires that each pair of class-specific sub-dictionaries (D_i, P_i) should have good representation ability to samples from the associated class, but poor representation ability to samples from other classes. **Furthermore**, from the perspective of classification, we hope that coding coefficients of samples from the same class have high similarity, i.e., the coefficient matrix for each class is low-rank. Therefore, we can employ the low-rank technique to improve the similarity between coding coefficients from the same class, which will facilitate the following classification. Figure 1 illustrates the basic framework of our approach.

Based on the above analysis, the objective function of our approach is designed as follows:

$$\min_{D,P,w} \sum_{i=1}^{C}(E_{rep} + E_{dis} + \mu E_{rank}) \;\; s.t. \; \|d_j\|_2^2 \leq 1 \; \forall j, \tag{4}$$

where μ is a scalar constant, d_j denotes the j^{th} atom of dictionary D, and the constraint is to restrict the energy of each atom. The details of three terms are as follows:

- $E_{rep} = \|\Phi(X_i) - \Phi(X)DP\Phi(X)^T\Phi(X_i)\|_F^2$ is the reconstruction error, which ensures that the learned dictionary pair can well reconstruct samples in the kernel space.
- $E_{dis} = \|\Phi(X_i) - \Phi(X)D_iP_i\Phi(X)^T\Phi(X_i)\|_F^2 + \lambda\|P_i\Phi(X)^T\Phi(\overline{X_i})\|_F^2$ is the dictionary discriminant term, which ensures that the i^{th} sub-dictionary pair can well represent samples from the i^{th} class in the kernel space, but has poor representation ability to samples from other classes. Here, λ is a scalar constant, and $\overline{X_i}$ denotes the complementary data matrix of X_i in the whole training set X.
- $E_{rank} = \|P_i\|_*$ is the low-rank regularization term, which ensures that each analysis sub-dictionary is low-rank, such that the obtained coding coefficients for each class have high similarity. Here, $\|.\|_*$ represents the nuclear norm of a matrix.

The Optimization of MKLDPL

The objective function in (4) is generally non-convex. We introduce a variable matrix A and relax (4) to the following problem:

$$\begin{aligned}\min_{D,P,A,w} \sum_{i=1}^{C}(&\|\mathcal{Z}_i - \Phi(X)D_iA_i\|_F^2 + \|\Phi(X_i) - \Phi(X)D_iA_i\|_F^2 \\ &+ \lambda\|P_i\Phi(X)^T\Phi(\overline{X_i})\|_F^2 + \mu\|P_i\|_* \\ &+ \tau\|P_i\Phi(X)^T\Phi(X_i) - A_i\|_F^2) \\ &s.t. \; \|d_j\|_2^2 \leq 1 \; \forall j\end{aligned} \tag{5}$$

where $\mathcal{Z}_i = \Phi(X_i) - \sum_{j=1, j\neq i}^{C}\Phi(X)D_jP_j\Phi(X)^T\Phi(X_i)$, and τ is a scalar constant. Then we divide (5) into two subproblems: (i) optimizing dictionary pair $\{D, P, A\}$ by fixing w; (ii)optimizing kernel weights w by fixing D, P and A , and then optimize them alternatively.

(1) Optimizing D, P and A

Here, we optimize D, P and A class by class. When one class is updated, variables related to other classes are fixed. For the i^{th} class, we update D_i, P_i, A_i alternatively (updating one by fixing the other two). We initialize D and P as random matrices with unit Frobenius norm for each column vector. Detailed updating steps are as follows.

Step 1: Updating A_i. When D_i and P_i are fixed, the objective function related to A_i can be written as:

$$\begin{aligned}\min_{A_i} &\|\mathcal{Z}_i - \Phi(X)D_iA_i\|_F^2 + \|\Phi(X_i) - \Phi(X)D_iA_i\|_F^2 \\ &+ \tau\|P_i\Phi(X)^T\Phi(X_i) - A_i\|_F^2\end{aligned} \tag{6}$$

By setting the derivative to zero, we can get

$$\begin{aligned}A_i = &(2D_i^T\Phi(X)^T\Phi(X)D_i + \tau I)^{-1} \times (D_i^T\Phi(X)^T\mathcal{Z}_i + \\ &D_i^T\Phi(X)^T\Phi(X_i) + \tau P_i\Phi(X)^T\Phi(X_i))\end{aligned} \tag{7}$$

where I is an identity matrix of $m_i \times m_i$. Note that the values of $\Phi(X)^T\Phi(X)$ and $\Phi(X)^T\Phi(X_i)$ are $\mathcal{K}(X, X)$ and $\mathcal{K}(X, X_i)$, respectively.

Step 2: Updating P_i. When D_i and A_i are fixed, P_i can be updated by (8):

$$\begin{aligned}\min_{P_i} \; &\lambda\|P_i\Phi(X)^T\Phi(\overline{X_i})\|_F^2 + \mu\|P_i\|_* \\ &+ \tau\|P_i\Phi(X)^T\Phi(X_i) - A_i\|_F^2\end{aligned} \tag{8}$$

Algorithm 1 Optimization process of MKLDPL

Require: Kernel matrix $\mathcal{K}_j$, $j = 1, 2, ..., L$
1: Initialize λ, μ, τ, D and P;
2: **while** not converge **do**
3: Fix D, P, w, update A by (7);
4: Fix D, A, w, update P by (11);
5: Fix P, A, w, update D by solving (14) and (15) iteratively;
6: Fix D, P, A, update w by solving $\mathcal{J}_w$;
7: **end while**
Ensure: D, P and w

To address the optimization of problem (8), we transform it into the same minimization problem by introducing a relaxing variable Z:

$$\min_{P_i,Z} f(P_i)+\mu\|Z\|_* \ s.t.\ P_i = Z, \tag{9}$$

where $f(P_i)=\lambda\|P_i\Phi(X)^T\Phi(\overline{X_i})\|_F^2+\tau\|P_i\Phi(X)^T\Phi(X_i)-A_i\|_F^2$. Problem (9) can be addressed by solving the following Augmented Lagrange Multiplier problem:

$$\min_{P_i,Z} f(P_i)+\mu\|Z\|_* + \langle T_1, P_i-Z\rangle+\frac{\beta}{2}\|P_i-Z\|_F^2 \tag{10}$$

where $\beta>0$ is a penalty parameter whilst T_1 is the Lagrange multiplier. The optimal solution of (10) can be obtained by the ADMM algorithm (Gabay and Mercier 1976):

$$\begin{cases} P_i=\arg\min_{P_i}\frac{1}{\beta}f(P_i)+\frac{1}{2}\|P_i-Z+\frac{T_1}{\beta}\|_F^2 \\ Z=\arg\min_{Z}\frac{\mu}{\beta}\|Z\|_*+\frac{1}{2}\|Z-P_i-\frac{T_1}{\beta}\|_F^2 \\ T_1=T_1+\beta(P_i-Z) \end{cases} \tag{11}$$

where the optimization of Z can be solved with Singular Value Thresholding (SVT) (Cai, Candès, and Shen 2010). In computation, the value of $\Phi(X)^T\Phi(\overline{X_i})$ is $\mathcal{K}(X,\overline{X_i})$.

Step 3: Updating D_i. When P_i and A_i are fixed, the objective function related to D_i can be written as:

$$\begin{aligned} &\min_{D_i}\|\mathcal{Z}_i-\Phi(X)D_iA_i\|_F^2+\|\Phi(X_i)-\Phi(X)D_iA_i\|_F^2 \\ &\quad s.t.\ \|d_j\|_2^2\le 1\ \forall j \end{aligned} \tag{12}$$

To solve problem (12), we use a similar way as (Gu et al. 2014). Specifically, we first relax (12) by introducing a variable matrix B:

$$\begin{aligned} &\min_{D_i,B}\|\mathcal{Z}_i-\Phi(X)B\|_F^2+\|\Phi(X_i)-\Phi(X)B\|_F^2+\alpha\|B-D_iA_i\|_F^2 \\ &\quad s.t.\ \|d_j\|_2^2\le 1\ \forall j \end{aligned} \tag{13}$$

where α is a scalar constant. Then we can solve problem (13) by updating D_i and B alternatively. By fixing D_i, B can be updated as follows:

$$\min_{B}\|\mathcal{Z}_i-\Phi(X)B\|_F^2+\|\Phi(X_i)-\Phi(X)B\|_F^2+\alpha\|B-D_iA_i\|_F^2 \tag{14}$$

The solution of problem (14) can be easily obtained by setting the derivative to zero.

By fixing B, D_i can be updated by:

$$\min_{D_i}\alpha\|B-D_iA_i\|_F^2 \ s.t.\ \|d_j\|_2^2\le 1\ \forall j \tag{15}$$

Problem (15) can be easily solved by the similar way as (Gu et al. 2014).

(2) Optimizing kernel weights w

To update kernel weights w, we need to reformulate the objective function (5) into the form w.r.t. w. There are four terms that are related to w in Eq. (5), including $\|\mathcal{Z}_i-\Phi(X)D_iA_i\|_F^2$, $\|\Phi(X_i)-\Phi(X)D_iA_i\|_F^2$, $\lambda\|P_i\Phi(X)^T\Phi(\overline{X_i})\|_F^2$ and $\tau\|P_i\Phi(X)^T\Phi(X_i)-A_i\|_F^2$). Here, we only provide the reformulation of one term, other terms can be transformed similarly.

$$\begin{aligned} &\lambda\|P_i\Phi(X)^T\Phi(\overline{X_i})\|_F^2=\lambda\|P_i\mathcal{K}(X,\overline{X_i})\|_F^2 \\ &=tr(\mathcal{K}(X,\overline{X_i})^TP_i^TP_i\mathcal{K}(X,\overline{X_i})) \\ &=w^T\times Q_i\times w, \end{aligned} \tag{16}$$

where

$$Q_i=\begin{bmatrix} Q_i(1,1) & ... & Q_i(1,L) \\ \vdots & \vdots & \vdots \\ Q_i(L,1) & ... & Q_i(L,L) \end{bmatrix}. \tag{17}$$

Here, $Q_i(m,n)=tr(\mathcal{K}_m(X,\overline{X_i})^TP_i^TP_i\mathcal{K}_n(X,\overline{X_i}))$. Denote by $\mathcal{J}_w$ the completed objective function w.r.t. w. Combined with the constraints $w_j\ge 0$ and $\sum_{j=1}^L w_j=1$, $\mathcal{J}_w$ can be easily solved with quadratic program (QP) solver (Coleman and Li 1996). The optimization process of MKLDPL is summarized in Algorithm 1.

Complexity and Convergence Analysis

We firstly give a detail discussion on the computation complexity. In the training phase, $\{A,\ P,\ D\}$ and w are updated alternatively. In the process of optimizing $\{A,\ P,\ D\}$ for each class, the time complexity of updating A_i is $O(m_iN^2+m_i^2N+m_i^3+m_in_iN+m_iMN+n_iMN)$, where M is the number of atoms in D, n_i is the sample number in the i^{th} class; Updating P_i takes $kO(N^3+m_in_iN+m_i^3)$, where k is the iteration number in the ADMM algorithm, and k is usually smaller than 20; For the optimization of D_i, updating B costs $O(N^3+MN^2+n_iMN)$, and updating D_i takes $pO(m_in_iN+m_i^2N+m_i^2n_i+m_i^3)$, where p is the iteration number, which is usually smaller than 10. For updating w, most of the time is spent on computing the trace of each sub-matrix, and the time complexity is $O(N^3)$. The most time-consuming parts include updating P_i, updating B and updating w. Fortunately, the operations that cost $O(N^3)$ when updating P_i and B, are not changed for all classes, thus we only need to compute them once. This will greatly improve the efficiency of our approach.

Although the objective function in (5) is not jointly convex w.r.t. $\{D, P, A, w\}$, it is convex w.r.t. each of them when the others are fixed, i.e., in each step of the optimization, the sub-problem is convex. Figure 2 shows the convergence curve of our algorithm on Flowers17. One can see that the energy converges quickly and well. In most of our experiments, our algorithm will converge in less than 25 iterations.

Classification Scheme

In our MKLDPL model, the analysis dictionary is used to produce the representation coefficient for samples, and the

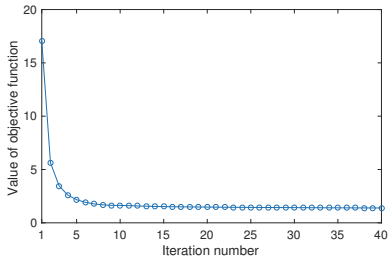


Figure 2: Convergence curve on the Flowers17 dataset.

synthesis dictionary is used to reconstruct the samples in the kernel space. With the learned dictionary pair (D, P) and kernel weights w, we can perform the image classification task easily.

Let $\Phi(y)$ be a test image in the weighted multi-kernel space. We can classify y as follows:

$$class(y) = \min_i \|\Phi(y) - \Phi(X)D_iP_i\Phi(X)^T\Phi(y)\|_F^2 \quad (18)$$

The classification is done by assigning the test sample to the class with the smallest reconstruction error.

Experiments

To evaluate the effectiveness of our approach, we conduct extensive experiments on three commonly used datasets that provide multiple feature representations, including Oxford Flowers17 dataset (Nilsback and Zisserman 2006), Oxford Flowers102 dataset (Nilsback and Zisserman 2008) and Caltech101 dataset (Fei-Fei, Fergus, and Perona 2007).

Baselines

In this section, We compare our approach with two types of methods, which are linear multiple feature representations based DL methods and multiple kernel based methods. The compared linear multiple representations based DL methods include uncorrelated multi-view discrimination DL (UMD^{2}L) (Jing et al. 2014) and multi-view low-rank DL (MLDL) (Wu et al. 2016). The compared multiple kernel based methods include ℓ_p-norm multiple kernel learning (ℓ_p-norm MKL) (Kloft et al. 2011), multiple kernel learning for sparse representation-based classification (MKL-SRC) (Shrivastava, Patel, and Chellappa 2014), and discriminative multiple kernel DL (DMKDL) (Thiagarajan, Ramamurthy, and Spanias 2014).

Parameter Settings

In objective function (5), there are three parameters, i.e., λ, μ and τ. In the experiments, we choose these parameters by 5-fold cross validation on each dataset. For all the competing methods, we tune their parameters for the best performance. In experiments, we set the dictionary size (i.e., m_i) of D_i as 60, 70 and 70 for Flowers17, Flowers102 and Caltech101 datasets, respectively. In addition, the kernel function $k(x, y) = exp(-\|x - y\|^2/s)$ is used for all types of feature representations. We set the kernel parameter with a similar way as (Gönen 2012; Thiagarajan, Ramamurthy, and Spanias 2014), i.e., s is set as the mean of the pairwise distances of samples.

Table 1: Average classification accuracies ($\pm$ standard deviation) (%) on three datasets.

Methods	Flowers17	Flowers102	Caltech101
UMD^{2}L	85.17±1.09	73.25±0.54	76.49±0.28
MLDL	85.88±0.98	73.76±0.35	76.71±0.19
L_p-norm MKL	85.29±1.07	73.62±0.49	78.25±0.22
MKL-SRC	86.47±1.85	74.38±0.53	76.13±0.17
DMKDL	88.13±2.33	76.54±0.55	82.66±0.36
MKLDPL	91.69±1.15	80.17±0.48	86.81±0.21

Evaluation on Oxford Flowers17 Dataset

Oxford Flowers17 dataset (Nilsback and Zisserman 2006) consists of 17 species of flowers with 80 images per class. We use the three predefined splits with 40 images for training and 20 images for testing from each class. Classification is carried out based on distance matrices of 7 different features, including color, shape, texture, HSV, HOG and SIFT on the foreground internal region, and SIFT on the foreground boundary. The parameters λ, μ and τ used in our algorithm are set as 0.3, 0.05 and 0.4, respectively. We conduct experiments 10 times and report the average classification accuracies (the same strategy is used for other datasets).

Table 1 shows classification results of compared methods on the Oxford Flowers17 dataset. We can see that MKLDPL improves the average classification accuracy at least by 3.56% (=91.69-88.13) on the Oxford Flowers17 dataset. **The major reasons why MKLDPL can achieve better results are three-fold:** (i) MKLDPL employs a more effective DL manner, i.e., analysis-synthesis dictionary pair learning. (ii) MKLDPL imposes the low-rank constraint on the analysis dictionary, which can improve the similarity of coding coefficients from the same class. (iii) MKLDPL designs a discriminant term to make use of the class label information, which ensures that the learned dictionary pair owns favorable discriminability.

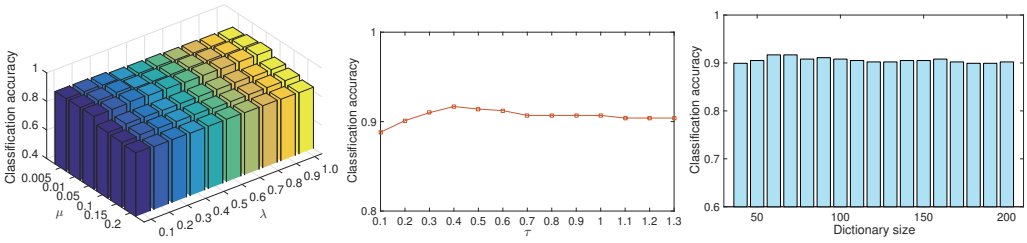
Evaluation on Oxford Flowers102 Dataset

Oxford Flowers102 dataset (Nilsback and Zisserman 2008) contains flower images from 102 different types with more than 40 images per class. There is a predefined split consisting of 2040 training and 6149 testing images. There are four precomputed distance matrices over different feature representations. The parameters λ, μ and τ used in our algorithm are set as 0.2, 0.05 and 0.5, respectively.

Table 1 shows the classification accuracies on the Oxford Flower102 dataset. We can see that MKLDPL improves the average accuracy at least by 3.63% (=80.17-76.54) as compared with other methods.

Evaluation on Caltech101 Dataset

The Caltech101 dataset (Fei-Fei, Fergus, and Perona 2007) contains object images from 102 classes. We use the three predefined splits with 30 images for training and 15 images



(a) (b) (c)

Figure 3: Classification accuracies on the Flowers17 dataset versus different values of (a) λ, μ, (b) τ and (c) dictionary size.

Table 2: Average classification accuracies ($\pm$ standard deviation) (%) of MKLDPL, MKDPL and DMKDL.

Datasets	**MKLDPL**	**MKDPL**	DMKDL
Flowers17	91.69±1.15	90.31±1.08	88.13±2.33
Flowers102	80.17±0.48	78.88±0.51	76.54±0.55
Caltech101	86.81±0.21	85.65±0.25	82.66±0.36

for testing from each class. We conduct classification experiment based on the PHOW color, geometric blur and self-similarity descriptors. The parameters λ, μ and τ used in our algorithm are set as 0.3, 0.05 and 0.6, respectively.

Table 1 shows classification results on the Caltech101 dataset. It is observed that our MKLDPL approach obtains much higher classification accuracy than the competing methods. In particular, MKLDPL improves the average matching rate at least by 4.15% (=86.81%-82.66%).

Discussion

Effect of the Low-rank Regularization Term. In our model, the low-rank regularization is employed to ensure that the matrix formed by the coding coefficients (which are obtained using analysis dictionary) of samples from the same class is low-rank, such that the similarity between coding coefficients of the same class can be improved. To evaluate the effect of the low-rank regularization term to our approach, we conduct MKLDPL with or without the low-rank term. We call the version of MKLDPL without low-rank term as "MKDPL". Table 2 shows the comparison of classification results on all datasets. We can see that MKLDPL outperforms MKDPL at least by 1.16%, which means that our approach can obtain more favorable discriminative capability by employing the low-rank regularization term.

Effect of Dictionary Pair Learning. Our work first introduces the dictionary pair learning into multiple kernel learning. To evaluate the effectiveness of dictionary pair learning, we made a comparison between MKDPL (the modified version of our approach without using the low rank term) and DMKDL (a representative multiple kernel dictionary learning method based on synthesis dictionary). Table 2 reports the results of MKDPL and DMKDL. We can see that MKDPL achieves better results than DMKDL, which means that dictionary pair learning is beneficial to the performance improvement of multiple-kernel dictionary learning.

Comparison in the Presence of Noise. In this experiment, we aim to evaluate the effect of noise to the performance of all compared methods. To this end, we first add random noises to each image with the same way as (Wu et al. 2016), and then conduct experiments using the features extracted from the noisy images. Table 3 reports the classification accuracies of all methods versus different noise percentages on the Caltech101 dataset. We can see that MKLDPL still achieves better results than competing methods under each noise percentage, which means that our approach has favorable robustness in the presence of noise.

Table 3: Average classification accuracies ($\pm$ standard deviation) (%) versus different noise percentages on Caltech101.

Methods	10%	20%	30%
UMD^{2}L	73.51±0.36	68.55±0.58	56.70±0.93
MLDL	75.28±0.25	72.48±0.52	64.02±0.91
L_p-norm MKL	73.75±0.31	65.36±0.64	46.49±1.15
MKL-SRC	73.06±0.33	67.52±0.67	56.24±1.08
DMKDL	79.78±0.52	75.62±0.78	65.16±1.36
MKLDPL	84.96±0.35	81.66±0.71	72.13±1.17

Comparison of Computational Cost. Among the compared methods: L$_p$-norm MKL is based on SVM; UMD^{2}L, MLDL, MKL-SRC and DMKDL are based on dictionary learning or sparse representation. Comparing to L$_p$-norm MKL, the other four methods have higher computation complexities. Specifically, computation complexities of linear methods UMD^{2}L and MLDL are respectively $O(N * d^2 * M^\varepsilon) + O(M^2 * d * N)$ and $O(N * d^2 * M^\varepsilon) + O(d^3)$, where d denotes the feature dimension, M is the number of atoms in dictionary (usually not very large), and $\varepsilon >= 1.2$. Since the computation complexity of our approach is $O(N^3)$, the complexity comparison between our approach and UMD^{2}L, MLDL depends on the values of d and N. For multi-kernel methods MKL-SRC and DMKDL, their computation complexities are $O(N^3)$ and $O(N^4)$, respectively. We can see that the computational complexity of our approach is comparable to that of MKL-SRC, and lower than that of DMKDL.

Parameter Analysis. Next, we provide a discussion about the sensitivity of MKLDPL to different choices of the parameters λ, μ and τ. We take the Oxford Flowers17 dataset as an example and conduct experiments by changing values of λ, μ and τ. When evaluating one parameter, the others are fixed to the values used in the Flowers17 classification

experiment. Figure 3 (a) and (b) shows the classification accuracies of MKLDPL versus different values of λ, μ and τ, respectively. We can observe that MKLDPL is not sensitive to the choice of λ in the range [0.1,1], and MKLDPL achieves the best results when λ and μ are separately set as 0.3 and 0.05, and MKLDPL can achieve good performance when τ is in the range [0.3, 1]. Similar results can also be obtained on the other datasets.

Dictionary size is also an important parameter in our approach. To observe the effect of dictionary size (i.e., m_i), we conduct experiments by changing m_i in the range of [40,200] with step length 10. Figure 3 (c) shows the classification accuracies with different dictionary size on Flowers17 dataset. We achieved similar results on the other datasets. We can see that our approach can obtain a relatively good performance when m_i is set as 60. Due to limited space, the evaluation of statistical significance of difference is reported in supplemental material.

Conclusion

In this paper, we propose a multi-kernel low-rank dictionary pair learning (MKLDPL) approach for multiple features based image classification. Different from existing kernel dictionary learning methods, MKLDPL jointly learns a kernel synthesis dictionary and a kernel analysis dictionary from the training data. With the designed discriminant term and the low-rank regularization term, MKLDPL can ensure that the learned dictionary pair owns favorable discriminability. Experimental results on three public datasets show that our approach achieves the best classification accuracies, and also demonstrate the effectiveness of applying low-rank regularization to analysis dictionary.

Acknowledgments

Thanks for the valuable comments of Editor and reviewers. This work was supported by the NSFC (Nos. 61272273, 61231015), 863 Program (No. 2015AA016306).

References

Bucak, S. S.; Jin, R.; and Jain, A. K. 2014. Multiple kernel learning for visual object recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36(7):1354–1369.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Coleman, T. F., and Li, Y. 1996. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization* 6(4):1040–1058.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding* 106(1):59–70.

Gabay, D., and Mercier, B. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* 2(1):17–40.

Gangeh, M. J.; Ghodsi, A.; and Kamel, M. S. 2013. Kernelized supervised dictionary learning. *Signal Processing, IEEE Transactions on* 61(19):4753–4767.

Gönen, M. 2012. Bayesian efficient multiple kernel learning. In *International Conference on Machine Learning*, 1–8.

Gu, S.; Zhang, L.; Zuo, W.; and Feng, X. 2014. Projective dictionary pair learning for pattern classification. In *Conference on Neural Information Processing Systems*, 793–801.

Jiang, Z.; Lin, Z.; and Davis, L. S. 2013. Label consistent k-svd: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(11):2651–2664.

Jing, X.-Y.; Hu, R.-M.; Wu, F.; Chen, X.-L.; Liu, Q.; and Yao, Y.-F. 2014. Uncorrelated multi-view discrimination dictionary learning for recognition. In *AAAI Conference on Artificial Intelligence*, 2787–2795.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; and Zien, A. 2011. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research* 12:953–997.

Liu, X.; Wang, L.; Zhang, J.; and Yin, J. 2014. Sample-adaptive multiple kernel learning. In *AAAI Conference on Artificial Intelligence*, 1975–1981.

Nilsback, M.-E., and Zisserman, A. 2006. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1447–1454.

Nilsback, M., and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 722–729.

Shi, Y.; Gao, Y.; Yang, Y.; Zhang, Y.; and Wang, D. 2013. Multimodal sparse representation-based classification for lung needle biopsy images. *Biomedical Engineering, IEEE Transactions on* 60(10):2675–2685.

Shrivastava, A.; Patel, V. M.; and Chellappa, R. 2014. Multiple kernel learning for sparse representation-based classification. *Image Processing, IEEE Transactions on* 23(7):3013–3024.

Shrivastava, A.; Pillai, J. K.; and Patel, V. M. 2015. Multiple kernel-based dictionary learning for weakly supervised classification. *Pattern Recognition* 48(8):2667–2675.

Thiagarajan, J. J.; Ramamurthy, K. N.; and Spanias, A. 2014. Multiple kernel sparse representations for supervised and unsupervised learning. *Image Processing, IEEE Transactions on* 23(7):2905–2915.

Thiagarajan, J. J.; Ramamurthy, K. N.; and Spanias, A. 2015. Learning stable multilevel dictionaries for sparse representations. *Neural Networks and Learning Systems, IEEE Transactions on* 26(9):1913–1926.

Van Nguyen, H.; Patel, V. M.; Nasrabadi, N. M.; and Chellappa, R. 2013. Design of non-linear kernel dictionaries for object recognition. *Image Processing, IEEE Transactions on* 22(12):5123–5135.

Wu, F.; Jing, X.-Y.; You, X.; Yue, D.; Hu, R.; and Yang, J.-Y. 2016. Multi-view low-rank dictionary learning for image classification. *Pattern Recognition* 50:143–154.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *CoRR* abs/1304.5634.

Yang, M.; Zhang, L.; Feng, X.; and Zhang, D. 2011. Fisher discrimination dictionary learning for sparse representation. In *IEEE International Conference on Computer Vision*, 543–550.

Yang, M.; Liu, W.; Luo, W.; and Shen, L. 2016. Analysis-synthesis dictionary learning for universality-particularity representation based classification. In *AAAI Conference on Artificial Intelligence*, 2251–2257.

Zhang, L.; Zhou, W.-D.; Chang, P.-C.; Liu, J.; Yan, Z.; Wang, T.; and Li, F.-Z. 2012. Kernel sparse representation-based classifier. *Signal Processing, IEEE Transactions on* 60(4):1684–1695.