MULTI-KERNEL COLLABORATIVE REPRESENTATION FOR IMAGE CLASSIFICATION

Weiyang Liu[†], *Zhiding Yu*[‡], *Yandong Wen*[#], *Meng Yang*[§*], *and Yuexian Zou*[†*]

[†]Sch. of ECE, Peking Univ. [‡]Dept. of ECE, Carnegie Mellon Univ.
[#]Sch. of EIE, South China Univ. of Tech. [§]College of CS & SE, Shenzhen Univ.
Email: wyliu@pku.edu.cn, yang.meng@szu.edu.cn*, zouyx@pkusz.edu.cn*

ABSTRACT

We consider the image classification problem via multiple kernel collaborative representation (MKCR). We generalize the kernel collaborative representation based classification to a multi-kernel framework where multiple kernels are jointly learned with the representation coefficients. The intrinsic idea of multiple kernel learning is adopted in our MKCR model. Experimental results show MKCR converges within reasonable iterations and achieves state-of-the-art performance.

Index Terms— Multi-Kernel, Collaborative Representation, Image Classification

1. INTRODUCTION

One of the major reasons for the prominence of the sparse representation-based classification (SRC) [1] is its discriminative power and robustness in classifying visual categories, especially faces. SRC classifies images by enforcing l_1 norm constraint to the representation coefficients and computing the residuals of each class. Such sparse representation technique is also widely used in a variety of problems including image restoration [2], image denoising [3] and data clustering [4]. Recently, Zhang et al. generalize the SRC model and propose the collaborative representation-based classification (CRC) [5, 6] with impressive results on face recognition. CRC instantiates an efficient regularized least square algorithm which constrains the representation error and regularization term with l_2 norm. To enhance the discrimination power of CRC, our previous work [7, 8] considers to apply kernel technique to CRC model, reporting promising results in image classification tasks. However, [7] only adopts a pre-fixed kernel that not only may greatly affect the classification accuracy but also is difficult to determine. Since image classification tasks usually handle various subjects and the images may vary a lot, such fixed kernel strategy may be detrimental to the performance. Naturally, we consider to learn multiple kernels for CRC, further boosting its discrimination power.

Recent efforts on multiple kernel learning (MKL) [9–11] have shown that learning support vector machine (SVM) [12] with multiple kernels not only increases the accuracy but also enhances the interpretability of the resulting classifiers. Single kernel-based approaches often have trouble dealing with large-scale data with various distributions [13], non-flat data [14], unnormalised data [10] or data containing heterogeneous information [15]. In most cases, MKL refers to learning the optimal linear combinations of kernels. The basic idea behind MKL is to add a set of parameters to the minimization problem of the learning algorithm. Thus, this paper generalizes the kernel framework in [7, 16] and proposes to learn the optimal linear combination of multiple kernels from training samples. Specifically, we learn the weights of kernels by minimizing the representation error, enhancing the representation ability of the kernel dictionary. Moreover, we constrain the sum of weights to be 1 with each weight nonnegative. Note that, MKCR with single kernel can degenerate to KCRC [8, 16]. In fact, learning multiple kernels for KCRC shares the similar philosophy with dictionary learning [17–20]. They both aim to enhance the representation ability of dictionaries. Dictionary learning uses a direct approach to learn a representation bases, while MKCR utilizes a different insight by learning multiple kernels.

2. PRELIMINARIES

Let $\boldsymbol{D}$ denote a class-specific dictionary that contains k-class training samples, i.e., $\boldsymbol{D}=\{\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_k\}\in\mathbb{R}^{m\times n}$ where $n=\sum_{j=1}^{k} n_j$ (n_j is the sample number of the jth class) and m is the feature dimension. The sub-dictionary corresponding to the ith class is denoted by $\boldsymbol{D}_i=\{\boldsymbol{d}_{u(i)-n_i+1},\cdots,\boldsymbol{d}_{u(i)}\}$ in which $u(i)=\sum_{j=1}^{i} n_j$ and $\boldsymbol{d}_j$ is the jth training samples in $\boldsymbol{D}$. CRC represents the query sample $\boldsymbol{y}$ by solving $\boldsymbol{x}$ in the following generic model:

$$\hat{\boldsymbol{x}}=\arg\min_{x}\left(\|\boldsymbol{y}-\boldsymbol{D}\boldsymbol{x}\|_q^q+\mu\|\boldsymbol{x}\|_p^p\right) \tag{1}$$

where μ is the regularization parameter and $p,q\in\{1,2\}$. The combinations of p,q give different instantiations. For example, SRC is under the condition of $p=1,q\in\{1,2\}$.

After defining the nonlinear mapping $\boldsymbol{y}\in\mathbb{R}^m\mapsto\boldsymbol{\phi}(\boldsymbol{y})\in\mathbb{F}$ and kernel $K(\boldsymbol{v}',\boldsymbol{v}'')=\langle\boldsymbol{\phi}(\boldsymbol{v}'),\boldsymbol{\phi}(\boldsymbol{v}'')\rangle=\boldsymbol{\phi}(\boldsymbol{v}')^T\boldsymbol{\phi}(\boldsymbol{v}'')$, we

*corresponding author. This work was partially supported by the Shenzhen Science & Technology Fundamental Research Program (No. JCYJ20130329175141512, No. JCYJ20140509172609171), the National Natural Science Foundation of China (No. 61402289), and the National Science Foundation of Guangdong Province (No. 2014A030313558).

consider the dimensionality reduction in the kernel space by assuming the projection is a linear combination of images in $\mathbb{F}$ and further formulate the original model into the kernel CRC (KCRC) model [7]:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left(\|\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) - \boldsymbol{\Psi}^T \boldsymbol{G}\boldsymbol{x}\|_q^q + \mu\|\boldsymbol{x}\|_p^p \right) \quad (2)$$

where $\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) = [K(\boldsymbol{d}_1, \boldsymbol{y}), \cdots, K(\boldsymbol{d}_n, \boldsymbol{y})]^T$ and $\boldsymbol{G}$ ($G_{ij} = K(\boldsymbol{d}_i, \boldsymbol{d}_j)$) is the kernel Gram matrix. Note that, $\boldsymbol{\Psi}$ is used to perform dimensionality reduction [7] in kernel space, and $\boldsymbol{G} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$ in which $\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\boldsymbol{d}_1), \cdots, \boldsymbol{\phi}(\boldsymbol{d}_n)\}$. From the above model, two specific algorithms have been developed. With $p=2, q=2$, $\boldsymbol{x}$ can be solved by the least square algorithm at low computational cost. For more robustness, we can set $p=2, q=1$ and solve it with the augmented Lagrange multiplier (ALM) [21, 22]. Detailed algorithms refer to [7].

3. MULTI-KERNEL COLLABORATIVE REPRESENTATION

3.1. MKCR Model

Suppose we have a set of base kernel functions $\{K_u\}_{u=1}^U$, the multiple kernel function K is defined by

$$K(\boldsymbol{v}', \boldsymbol{v}'') = \sum_{u=1}^{U} \alpha_u K_u(\boldsymbol{v}', \boldsymbol{v}''),\ \alpha_u \geq 0 \quad (3)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_U]^T$ is the weighting vector for multiple kernels, and $\boldsymbol{v}', \boldsymbol{v}''$ denote two samples respectively. The multi-kernel gram matrix $\boldsymbol{G}$ can also be represented by base kernel gram matrices $\{\boldsymbol{G}_u\}_{u=1}^U$, written as follows:

$$\boldsymbol{G} = \sum_{u=1}^{U} \alpha_u \boldsymbol{G}_u,\ \alpha_u \geq 0. \quad (4)$$

To learn the optimal multiple kernels for KCRC, we need to feed a batch of input samples denoted by $\boldsymbol{Y}$. The collaborative representation of $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_h\}$ is defined as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_h\}$. After putting both Eq. (3) and Eq. (4) into the KCRC model and adding constraints for multiple kernels, we can derive the MKCR model:

$$\begin{aligned} \langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{\alpha}} \rangle = \arg\min_{\boldsymbol{X}, \boldsymbol{\alpha}} \Bigg(\Big\| \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{Y}) - \\ \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{G}_u \boldsymbol{X} \Big\|_F^2 \Bigg) \\ \text{s.t.} \begin{cases} \forall i,\ \|\boldsymbol{x}_i\|_p^p \leq \epsilon \\ \sum_{u=1}^{U} \alpha_u = 1,\ \alpha_u \geq 0 \end{cases} \end{aligned} \quad (5)$$

in which $\boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{Y}) = \{\boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{y}_1), \cdots, \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{y}_h)\}$ and ϵ is a constant. Similar to CRC and KCRC, both p and q can be set as 1 or 2 for different instantiations. we use l_1 norm to constrain the kernel weighting coefficients in order to assign more weights to kernels that can represent the input samples better. Following the conventional settings in MKL, the sum of the kernel weighting coefficients is equal to 1 and all of them are non-negative. Note that, the selection of $\boldsymbol{\Psi}$ has been discussed in [23]. However, we find no major performance advantages of using complex methods to construct $\boldsymbol{\Psi}$ in experiments (except for the speed and memory consumption), so we simply use the identity matrix as $\boldsymbol{\Psi}$ in the paper to retain the intuitive interpretation.

The optimization model in Eq. (5) aims to find a set of weights for multiple kernels that can best represent the input samples, or in other word, minimize the representation error.

3.2. Optimization

Because it is difficult to directly optimize the MKCR model, we alternatively optimize $\boldsymbol{X}$ and $\boldsymbol{\alpha}$ with an iterative, two-step strategy. At each iteration, one of $\boldsymbol{X}$ and $\boldsymbol{\alpha}$ is optimized while the other is fixed. Iterations are repeated until convergence or a maximum number of iterations is reached. Specific training algorithm refers to Algorithm 1.

On optimizing $\boldsymbol{X}$. By fixing $\boldsymbol{\alpha}$ in Eq. (5), we can write the optimization in Eq. (5) as

$$\begin{aligned} \hat{\boldsymbol{X}} = \arg\min_{\boldsymbol{X}} \Bigg(\Big\| \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{Y}) - \\ \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{G}_u \boldsymbol{X} \Big\|_F^2 \Bigg) \ \text{s.t.} \forall i,\ \|\boldsymbol{x}_i\|_p^p \leq \epsilon \end{aligned} . \quad (6)$$

Since $\boldsymbol{X}$ is the combination of each representation coefficients $\boldsymbol{x}_i (1 \geq i \geq h)$, we can separately optimize $\boldsymbol{x}_i$ and eventually combine them into $\boldsymbol{X}$. Using the Lagrange duality, we can further rewrite Eq. (6) as an equivalent optimization:

$$\begin{aligned} \hat{\boldsymbol{x}}_i = \arg\min_{\boldsymbol{x}_i} \Bigg(\Big\| \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{y}_i) - \\ \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{G}_u \boldsymbol{x}_i \Big\|_2^2 \Bigg) + \mu\|\boldsymbol{x}_i\|_p^p \end{aligned} \quad (7)$$

where μ is the regularization parameter. It can be learned from standard optimization theory that Eq. (6) and are equivalent if ϵ and μ obey some special relationship [24]. Eq. (7) becomes the standard KCRC model. When $p=1$, the optimization problem is identical to SRC [1]. When $p=2$, it has an efficient closed-form solution via the regularized least square algorithm [5]:

$$\hat{\boldsymbol{x}}_i = \left(\boldsymbol{G}^T \boldsymbol{\Psi} \boldsymbol{\Psi}^T \boldsymbol{G} + \mu \cdot \boldsymbol{I} \right)^{-1} \boldsymbol{G}^T \boldsymbol{\Psi} \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}_i) \quad (8)$$

in which $\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}_i) = \sum_{u=1}^{U} \alpha_u \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{y}_i)$, and $\boldsymbol{I}$ denotes an identical matrix.

On optimizing $\boldsymbol{\alpha}$. After fixing the representation coefficients $\boldsymbol{X}$, the optimization in Eq. (5) becomes

$$\begin{aligned}&\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \Bigg(\Big\| \boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{Y}) - \\ &\boldsymbol{\Psi}^T \sum_{u=1}^{U} \alpha_u \boldsymbol{G}_u \boldsymbol{X} \Big\|_F^2 \Bigg) \quad \text{s.t.} \ \sum_{u=1}^{U} \alpha_u = 1, \ \alpha_u \geq 0\end{aligned} \quad . \tag{9}$$

Since $\boldsymbol{\Psi}$, $\boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{Y})$ and $\boldsymbol{G}_u$ are given a prior, Eq. (9) can be therefore transformed to a standard constrained quadratic program [25], which is proven to be convex. We first rewrite the optimization to a concise form:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \Bigg(\Big\| \sum_{u=1}^{U} \alpha_u \boldsymbol{e}_u \Big\|_F^2 \Bigg) \quad \text{s.t.} \ \sum_{u=1}^{U} \alpha_u = 1, \ \alpha_u \geq 0 \tag{10}$$

where $\boldsymbol{e}_u$ is a matrix equal to $\boldsymbol{\Psi}^T \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{Y}) - \boldsymbol{\Psi}^T \boldsymbol{G}_u \boldsymbol{X}$. Then we stack each column of $\boldsymbol{e}_u$ into one column, turning it to a column vector $\tilde{\boldsymbol{e}}_u$. Defining $\tilde{\boldsymbol{E}} = \{\tilde{\boldsymbol{e}}_1, \tilde{\boldsymbol{e}}_2, \cdots, \tilde{\boldsymbol{e}}_U\}$, we can transform Eq.(10) to an equivalent one:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \big\| \tilde{\boldsymbol{E}} \boldsymbol{\alpha} \big\|_2^2 \quad \text{s.t.} \ \boldsymbol{C} \cdot \boldsymbol{\alpha} = 1, \ \alpha_u \geq 0 \tag{11}$$

where $\boldsymbol{C} = [1, 1, \cdots, 1] \in \mathbb{R}^{1 \times U}$. It is clear that Eq. (11) is a constrained quadratic program problem, which can be solved by various standard convex optimization solvers [26].

Initialization. While we first optimize $\boldsymbol{X}$, we uniformly initialize the weights for multiple kernels. Specifically, we initialize $\boldsymbol{\alpha}$ with $[\frac{1}{U}, \frac{1}{U}, \cdots, \frac{1}{U}]^T$. Detailed parameter selection is elaborated in experiments.

Algorithm 1 Training Procedure of MKCR.

```
Input: $\boldsymbol{\Psi}, {\boldsymbol{K}_u}_{u=1}^{U}, \boldsymbol{\alpha}^{(0)} = [\frac{1}{U}, \frac{1}{U}, \cdots, \frac{1}{U}]^T, \mu, p, r, i=0$
Output: $\boldsymbol{\alpha}$
 1: Optimization of $\boldsymbol{X}$
 2: $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^{(i)}$.
 3: Use Eq. (7) to optimize $\boldsymbol{x}_j (1 \leq j \leq h)$ with $\boldsymbol{\alpha}^{(i)}$ fixed.
 4: If $p=1$, the optimization can be solved by various $l_1$
    solvers, e.g. basis pursuit [27], FISTA [28], ALM [22]. If
    $p=2$, the optimization has a closed-form solution shown
    in Eq. (8).
 5: Obtain the representation $\boldsymbol{X}$.
 6: Optimization of $\boldsymbol{\alpha}$
 7: Transform the problem into a constrained quadratic pro-
    gram and use Eq. (11) to optimize $\boldsymbol{\alpha}$ with $\boldsymbol{X}$ fixed.
 8: $i \leftarrow i + 1$.
 9: $\boldsymbol{\alpha}^{(i)} \leftarrow \boldsymbol{\alpha}$.
10: If $i > r$, output $\boldsymbol{\alpha}$; Else, go to Step 1.
```

3.3. Classification Strategy

After learning the weights for multiple kernels, we perform the classification strategy which is similar to KCRC [7]. Because the weights $\boldsymbol{\alpha}$ for multiple kernels are already obtained, both the combination of multiple kernels and the multi-kernel Gram matrix can be determined. Therefore, the classification can be performed by the model in Eq. (2). Following similar settings to dictionary learning [17–20], both $\boldsymbol{Y}$ and $\boldsymbol{D}$ in the MKCR model comes from the training samples. After learning multiple kernels, we still adopt the original dictionary samples as the $\boldsymbol{D}$ in Eq. (2) in classification.

Specifically, if we set $p=1, q \in \{1, 2\}$, the optimization in Eq. (2) becomes exactly SRC except that the dictionary is $\boldsymbol{\Psi}^T \sum_{u=1}^{U} \boldsymbol{G}_u$ instead of $\boldsymbol{D}$ and the query sample is $\boldsymbol{\Psi}^T \sum_{u=1}^{U} \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{y})$ instead of $\boldsymbol{y}$. So in such circumstance, the procedure of classification is identical to SRC. If we set $p=2, q=2$, Eq. (2) can be efficiently solved by regularized least square algorithm [5]. The closed-form solution for the representation coefficients is

$$\hat{\boldsymbol{x}} = \big(\boldsymbol{G}^T \boldsymbol{\Psi} \boldsymbol{\Psi}^T \boldsymbol{G} + \mu \cdot \boldsymbol{I} \big)^{-1} \boldsymbol{G}^T \boldsymbol{\Psi} \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) \tag{12}$$

where $\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) = \sum_{u=1}^{U} \alpha_u \boldsymbol{K}_u(\boldsymbol{D}, \boldsymbol{y})$ and $\boldsymbol{G} = \sum_{u=1}^{U} \boldsymbol{G}_u$. The label of the query sample $\boldsymbol{y}$ is given by

$$\text{identity}(\boldsymbol{y}) = \text{arg}\ \text{min}_j \{\boldsymbol{r}_j\} \tag{13}$$

where $\boldsymbol{r}_j = \frac{1}{\|\hat{\boldsymbol{x}}_j\|_2} \big(\|\delta_j\{\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y})\} - \delta_j\{\boldsymbol{\Psi}^T \boldsymbol{G}\} \delta_j\{\hat{\boldsymbol{x}}\}\|_2 \big)$. $\delta_j\{\cdot\}$ denotes the function that removes the elements in a vector or matrix that are not related to the samples of the jth class in $\boldsymbol{D}$. If we set $p=2, q=1$, the optimization becomes

$$\begin{aligned}&\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \big(\|\boldsymbol{e}\|_1 + \mu \|\boldsymbol{x}\|_2^2 \big) \\ &\text{s.t.} \ \ \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) = \boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{x} + \boldsymbol{e}\end{aligned} \tag{14}$$

where $\boldsymbol{e} = \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) - \boldsymbol{\Psi}^T \boldsymbol{G} \boldsymbol{x}$. It is actually a constrained convex optimization that can be solved by ALM [21, 22]. Detailed algorithms of classification strategy are similar to [7].

3.4. Kernel Selection

The selection of kernels for MKCR can take insights from MKL [9–11, 13–15], since they share the same purpose to learn an optimal set of kernels. We briefly present a few examples for the selections of kernels.

Naive Combination. Naive combination refers to a simple multiple kernel combination that selects some widely used kernels, e.g. linear kernel, Log kernel, Gaussian kernel, Laplacian kernel, polynomial kernel, Perceptron kernel.

Discriminative Combination. Discriminative combination uses a set of discriminative kernels. It can be a unified kernel $K_u(\boldsymbol{v}', \boldsymbol{v}'') = \exp(-d_u^2(\boldsymbol{v}', \boldsymbol{v}')/\sigma^2)$ with different discriminative dissimilarity measures [11, 16].

Multi-Scale Combination. Multi-Scale combination [14] usually adopts Gaussian radial basis function (RBF) as the base kernel $K(\boldsymbol{v}', \boldsymbol{v}'') = \exp(-\gamma \|\boldsymbol{v}' - \boldsymbol{v}'\|^2)$. Then it varies the parameter γ of the base kernel to construct the multi-scale multiple kernels. The other base kernels can be constructed to multi-scale kernels following this procedure.

4. EXPERIMENTS AND RESULTS

4.1. Convergence Evaluation

The convergence of MKCR is evaluated in this experiment. We test three types of multiple kernel combinations. For naive combination, we use linear kernel, Gaussian kernel, Laplacian kernel, Perceptron kernel and Log kernel. Except for linear kernel, we use three different parameter for the other kernels, so it is totally 13 kernels. For discriminative combination, we adopt the same kernel combination as [11]. These discriminative kernels include GB-Dist, GB, SIFT-Dist, SIFT-SPM, SS-Dist, SS-SPM, C2-SWP, C2-ML, PHOG and GIST. The specific kernel construction can refer to [11]. For multi-scale combination, we use the Gaussian kernels with γ ranging from 0.1 to 1.5 with 0.1 step. It is totally 15 kernels. We use MNIST data set [29] where 500 input samples, 500 dictionary atoms and 1000 testing samples are used. From the results in Fig. 1, we can see the MKCR model can be converged in a reasonable iteration times. When MKCR model converges, its performance is better than the initial state.

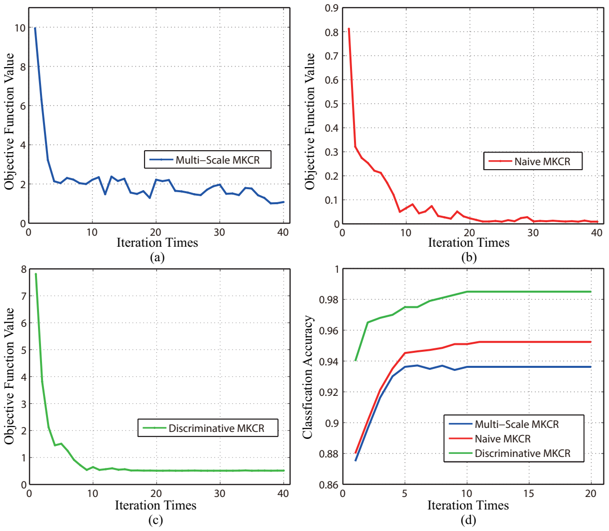

Fig. 1. (a) Example of convergence of multi-scale MKCR. (b) Example of convergence of naive MKCR. (c) Example of convergence of discriminative MKCR. (d) Classification accuracy v.s. iteration times in MNIST.

4.2. Experiments on Public Data sets

4.2.1. Experimental Settings

We apply MKCR to image classification tasks and evaluate its performance on Caltech101 data set [30] and 15 scene categories data set [31]. We use the same features as in [32] for clear comparison. For the Caltech1001 data set, we first extract SIFT descriptors from patches that are sampled via a grid and then extract the spatial pyramid feature based on the SIFT features. Finally, PCA is used to reduce the dimension to 3,000. For the 15 scene category data set, we compute the spatial pyramid feature via a four-level spatial pyramid and a SIFT-descriptor codebook. Similarly, PCA is used to reduce the dimension to 3,000. For KCRC, we use the Gaussian RBF kernel with $\gamma = 0.5$ as kernel function. The settings for D-KSVD and LC-KSVD follow [19, 20]. μ is $0.001 \times n/700$ (n is the size of the dictionary). We evaluate the three multi-kernel combinations using the same setup as Section 4.1.

4.2.2. Caltech101 Data Set

The Caltech101 data set contains 9,144 images from 102 classes (101 objects and 1 background class). We train on 5, 10, 15, 20, 25 samples per category and test on the rest. For MKCR, we use 5 more samples per category to learn the multiple kernels when the dictionary size is 510, 1020, 1530, 2040 and 2550. Note that, the discriminative MKCR in fact uses different features from the other approaches in the comparison, because its kernels are a unified representation of multiple features [11]. To show the gain from learning multiple kernels, we use three baselines (all weights of multiple kernels are equal, namely $\boldsymbol{\alpha}$ is pre-defined instead of learned) for comparison. Baseline1 refers to KCRC with the multi-scale multi-kernels, Baseline2 for the naive multi-kernels and Baseline3 for the discriminative multi-kernels. Experiments show MKCR performs best.

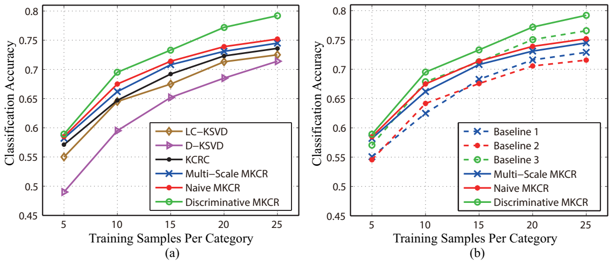

Fig. 2. Classification accuracy v.s. training samples per category. (a) MKCR and the other competitive algorithms. (b) MKCR and the baselines.

4.2.3. 15 Scene Categories Data Set

15 Scene Categories Data set contains 15 natural scene categories. Following the same experimental settings as [32], we randomly select 100 images per category for training and the rest for testing. For MKCR, we randomly use 30 images per category as the dictionary, and the remaining 70 images per category for learning multiple kernels.

Table 1. Classification accuracy (%) on 15 scene categories data set.

Method	Accuracy	Method	Accuracy
LC-KSVD [20]	92.94	D-KSVD [19]	89.16
KCRC [7]	97.21	Multi-Scale MKCR	97.78
Naive MKCR	98.00	Discriminative MKCR	98.15

5. CONCLUDING REMARKS

This paper proposes a novel multi-kernel collaborative representation approach for image classification. We generalize the KCRC model and learn the weights of multiple kernels for KCRC. To the best of our knowledge, such multi-kernel framework in SRC or CRC is first proposed. Experiments show MKCR achieves the state-of-the-art performance.

6. REFERENCES

[1] John Wright et al., "Robust face recognition via sparse representation," *IEEE TPAMI*, vol. 31, no. 2, pp. 210–227, 2009. 1, 3.2

[2] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma, "Image super-resolution as sparse representation of raw image patches," in *CVPR*, 2008. 1

[3] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE TIP*, vol. 15, no. 12, pp. 3736–3745, 2006. 1

[4] Ehsan Elhamifar and René Vidal, "Sparse subspace clustering," in *CVPR*, 2009. 1

[5] Lei Zhang and Meng Yang et al., "Sparse representation or collaborative representation: Which helps face recognition?," in *ICCV*, 2011. 1, 3.2, 3.3

[6] Lei Zhang and Meng Yang et al., "Collaborative representation based classification for face recognition," *arXiv Preprint arXiv:1204.2358*, 2012. 1

[7] Weiyang Liu, Lijia Lu, Hui Li, Wei Wang, and Yuexian Zou, "A novel kernel collaborative representation approach for image classification," in *ICIP*, 2014. 1, 2, 2, 3.3, 3.3, 1

[8] Weiyang Liu, Yandong Wen, Kai Pan, Hui Li, and Yuexian Zou, "A kernel-based l 2 norm regularized least square algorithm for vehicle logo recognition," in *Digital Signal Processing (DSP), 19th International Conference on*. IEEE, 2014. 1

[9] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *ICML*, 2004. 1, 3.4

[10] Cheng S Ong, Robert C Williamson, and Alex J Smola, "Learning the kernel with hyperkernels," *JMLR*, 2005. 1, 3.4

[11] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE TPAMI*, 2011. 1, 3.4, 4.1, 4.2.2

[12] Bernhard Scholkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2001. 1

[13] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet, "More efficiency in multiple kernel learning," in *ICML*, 2007. 1, 3.4

[14] Danian Zheng, Jiaxin Wang, and Yannan Zhao, "Non-flat function estimation with a multi-scale support vector regression," *Neurocomputing*, 2006. 1, 3.4

[15] Kristin P Bennett, Michinari Momma, and Mark J Embrechts, "Mark: A boosting algorithm for heterogeneous kernel models," in *ACM SIGKDD*, 2002. 1, 3.4

[16] Weiyang Liu, Zhiding Yu, Lijia Lu, Yandong Wen, Hui Li, and Yuexian Zou, "Kcrc-lcd: Discriminative kernel collaborative representation with locality constrained dictionary for visual categorization," *Pattern Recognition*, 2015 (to appear). 1, 3.4

[17] Meng Yang, D Zhang, and Xiangchu Feng, "Fisher discrimination dictionary learning for sparse representation," in *ICCV*, 2011. 1, 3.3

[18] Meng Yang, Dengxin Dai, Lilin Shen, and Luc Van Gool, "Latent dictionary learning for sparse representation based classification," in *CVPR*, 2014. 1, 3.3

[19] Qiang Zhang and Baoxin Li, "Discriminative k-svd for dictionary learning in face recognition," in *CVPR*, 2010. 1, 3.3, 4.2.1, 1

[20] Zhuolin Jiang, Zhe Lin, and Larry S Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *CVPR*, 2011. 1, 3.3, 4.2.1, 1

[21] Dimitri P Bertsekas, "Nonlinear programming," *Athena Scientific*, 1999. 2, 3.3

[22] Zhouchen Lin et al., "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv Preprint arXiv:1009.5055*, 2010. 2, 4, 3.3

[23] Li Zhang et al., "Kernel sparse representation-based classifier," *IEEE TSP*, vol. 60, no. 4, pp. 1684–1695, 2012. 3.1

[24] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès, "Nesta: a fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, 2011. 3.2

[25] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2009. 3.2

[26] Michael Grant, Stephen Boyd, and Yinyu Ye, *Disciplined convex programming*, Springer, 2006. 3.2

[27] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, 2001. 4

[28] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, 2009. 4

[29] Yann LeCun et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 4.1

[30] Li Fei-Fei et al., "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, vol. 106, no. 1, pp. 59–70, 2007. 4.2.1

[31] Svetlana Lazebnik et al., "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006. 4.2.1

[32] Zhuolin Jiang et al., "Label consistent k-svd: learning a discriminative dictionary for recognition," *IEEE TPAMI*, vol. 35, no. 11, pp. 2651–2664, 2013. 4.2.1, 4.2.3