Multiple Kernel Learning for Dimensionality Reduction

Yen-Yu Lin, Tyng-Luh Liu, *Member*, *IEEE*, and Chiou-Shann Fuh, *Member*, *IEEE*

Abstract—In solving complex visual learning tasks, adopting multiple descriptors to more precisely characterize the data has been a feasible way for improving performance. The resulting data representations are typically high-dimensional and assume diverse forms. Hence, finding a way of transforming them into a unified space of lower dimension generally facilitates the underlying tasks such as object recognition or clustering. To this end, the proposed approach (termed MKL-DR) generalizes the framework of *multiple kernel learning* for *dimensionality reduction*, and distinguishes itself with the following three main contributions: First, our method provides the convenience of using diverse image descriptors to describe useful characteristics of various aspects about the underlying data. Second, it extends a broad set of existing dimensionality reduction techniques to consider multiple kernel learning, and consequently improves their effectiveness. Third, by focusing on the techniques pertaining to dimensionality reduction, the formulation introduces a new class of applications with the multiple kernel learning framework to address not only the supervised learning problems but also the unsupervised and semi-supervised ones.

Index Terms—Dimensionality reduction, multiple kernel learning, object categorization, image clustering, face recognition.

1 Introduction

The fact that most visual learning problems deal with high-dimensional data has made dimensionality reduction an inherent part of the current research. Besides having the potential for a more efficient approach, working with a new space of lower dimension can often gain the advantage of better analyzing the intrinsic structures in the data for various applications. For example, dimensionality reduction can be performed to compress data for a compact representation [25], [56], to visualize high-dimensional data [40], [47], to exclude unfavorable data variations [8], or to improve the classification power of the nearest neighbor rule [9], [54].

Despite the great applicability, existing dimensionality reduction methods often suffer from two main restrictions. First, many of them, especially the linear ones, require data to be represented in the form of feature vectors. The limitation may eventually reduce the effectiveness of the overall algorithms when the data of interest could be more precisely characterized in other forms, e.g., bag-of-features [2], [33], matrices, or high-order tensors [54], [57]. Second, there seems to be a lack of a systematic way of integrating multiple image features for dimensionality reduction. When addressing applications where no single descriptor can appropriately depict the whole data set, this shortcoming becomes even more evident. Alas, it is usually the case for addressing today's vision applications, such as the recognition task in the *Caltech-101* data set [14] or the classification and detection tasks in the *Pascal VOC challenge* [13]. On the other hand, the advantage of using multiple features has indeed been consistently pointed out in a number of recent research efforts, e.g., [7], [18], [31], [50], [51].

Aiming to overcome the above-mentioned restrictions, we introduce a framework called MKL-DR that incorporates *multiple kernel learning* (MKL) into the training process of *dimensionality reduction* (DR) methods. It works with multiple *base kernels*, each of which is created based on a specific kind of data descriptor, and fuses the descriptors in the domain of kernel matrices. We will illustrate the formulation of MKL-DR with *graph embedding* [54], which provides a unified view for a large family of DR methods. Any DR technique expressible by graph embedding can therefore be generalized by MKL-DR to boost their power by simultaneously taking account of data characteristics captured in different descriptors. It follows that the proposed approach can extend the MKL framework to address, as the corresponding DR methods would do, not only the *supervised* learning problems but also the *unsupervised* and *semi-supervised* ones.

2 Related Work

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

2.1 Dimensionality Reduction

Techniques to perform dimensionality reduction for high-dimensional data can vary considerably from each other due to, e.g., different assumptions about the data distribution or the availability of the data labeling. We categorize them as follows:

- *Y.-Y. Lin and T.-L. Liu are with the Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan. E-mail: {yylin, liutyng}@iis.sinica.edu.tw.*
- *C.-S. Fuh is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan. E-mail: fuh@csie.ntu.edu.tw.*

Manuscript received 25 Jan. 2010; revised 9 July 2010; accepted 28 July 2010; published online 30 Sept. 2010.
Recommended for acceptance by S. Belongie.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-01-0054.
Digital Object Identifier no. 10.1109/TPAMI.2010.183.

2.1.1 Unsupervised DR

Principal component analysis (PCA) [25] is the most well-known one that finds a linear mapping by maximizing the projected variances. For nonlinear DR techniques, *isometric feature mapping* (Isomap) [47] and *locally linear embedding* (LLE) [40] both exploit the manifold assumption to yield the embeddings. And, to resolve the out-of-sample problem in Isomap and LLE, *locality preserving projections* (LPP) [23] are proposed to uncover the data manifold by a linear relaxation.

2.1.2 Supervised DR

Linear discriminant analysis (LDA) assumes that the data of each class have a Gaussian distribution, and derives a projection from simultaneously maximizing the between-class scatter and minimizing the within-class scatter. Alternatively, *marginal Fisher analysis* (MFA) [54] and *local discriminant embedding* (LDE) [9] adopt the assumption that the data of each class spread as a submanifold, and seek a discriminant embedding over these submanifolds.

2.1.3 Semi-Supervised DR

If the observed data are partially labeled, dimensionality reduction can be performed by carrying out discriminant analysis over the labeled ones while preserving the intrinsic geometric structures of the remaining. Such techniques are useful, say, for vision applications where user interactions are involved, e.g., *semi-supervised discriminant analysis* (SDA) [6] for content-based image retrieval with relevance feedback.

2.1.4 Kernelization

It is possible to *kernelize* a certain type of linear DR techniques into nonlinear ones. As shown in [6], [9], [23], [34], [41], [54], the kernelized versions generally can achieve significant improvements. In addition, kernelization provides a convenient way for DR methods to handle data not in vector form by specifying an associated kernel, e.g., the *pyramid matching kernel* [21] for data in the form of bag-of-features or the *dissimilarity kernel* [38] based on the pairwise distances.

2.2 Graph Embedding

A number of dimensionality reduction methods focus on modeling the pairwise relationships among data and utilize graph-based structures. In particular, the framework of graph embedding [54] provides a unified formulation for a broad set of such DR algorithms. Let $\Omega = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ be the data set. A DR scheme accounted for by graph embedding involves a complete graph G whose vertices are over Ω. A corresponding affinity matrix $W = [w_{ij}] \in \mathbb{R}^{N\times N}$ is used to record the edge weights that characterize the similarity relationships between pairs of training samples. Then, the optimal linear embedding $\mathbf{v}^* \in \mathbb{R}^d$ can be obtained by solving

$$\mathbf{v}^* = \underset{\substack{\mathbf{v}^\top XDX^\top\mathbf{v}=1, \text{ or} \\ \mathbf{v}^\top XL'X^\top\mathbf{v}=1}}{\arg\min} \mathbf{v}^\top XLX^\top\mathbf{v}, \tag{1}$$

where $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_N]$ is the data matrix and $L = \mathrm{diag}(W\cdot\mathbf{1}) - W$ is the graph Laplacian of G. Depending on the property of a problem, one of the two constraints in (1) will be used in the optimization. If the first constraint is chosen, a diagonal matrix $D = [d_{ij}] \in \mathbb{R}^{N\times N}$ is included for scale normalization. Otherwise, another complete graph G' over Ω is required for the second constraint, where L' and $W' = [w'_{ij}] \in \mathbb{R}^{N\times N}$ are, respectively, the graph Laplacian and affinity matrix of G'. The optimization problem (1) has an intuitive interpretation: $\mathbf{v}^\top X = [\mathbf{v}^\top\mathbf{x}_1\ \cdots\ \mathbf{v}^\top\mathbf{x}_N]$ represents the projected data; graph Laplacian L (or L') is to explore the *pairwise distances* of the projected data, while diagonal matrix D is to weightedly combine their *distances to the origin*. More precisely, the meaning of (1) can be better understood with the following equivalent problem:

$$\min_{\mathbf{v}} \quad \sum_{i,j=1}^N \|\mathbf{v}^\top\mathbf{x}_i - \mathbf{v}^\top\mathbf{x}_j\|^2 w_{ij} \tag{2}$$

$$\text{subject to} \quad \sum_{i=1}^N \|\mathbf{v}^\top\mathbf{x}_i\|^2 d_{ii} = 1, \text{ or} \tag{3}$$

$$\sum_{i,j=1}^N \|\mathbf{v}^\top\mathbf{x}_i - \mathbf{v}^\top\mathbf{x}_j\|^2 w'_{ij} = 1. \tag{4}$$

The constrained optimization problem (2) implies that only distances to the origin or pairwise distances of projected data (in the form of $\mathbf{v}^\top\mathbf{x}$) are modeled by the framework. By specifying W and D (or W and W'), Yan et al. [54] show that a set of dimensionality reduction methods, such as PCA [25], LPP [23], LDA, and MFA [54] can be expressed by (1). Clearly, LDE [9] and SDA [6] are also in the class of graph embedding.

2.3 Multiple Kernel Learning

MKL refers to the process of learning a kernel machine with multiple kernel functions or kernel matrices. Recent research efforts on MKL, e.g., [1], [20], [29], [39], [45], have shown that learning SVMs with multiple kernels not only increases the accuracy but also enhances the interpretability of the resulting classifiers. Our MKL formulation is to find an optimal way to linearly combine the given kernels. Suppose we have a set of base kernel functions $\{k_m\}_{m=1}^M$ (or base kernel matrices $\{K_m\}_{m=1}^M$). An *ensemble kernel function* k (or an *ensemble kernel matrix* K) is then defined by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^M \beta_m k_m(\mathbf{x}_i, \mathbf{x}_j), \quad \beta_m \geq 0, \tag{5}$$

$$K = \sum_{m=1}^M \beta_m K_m, \quad \beta_m \geq 0. \tag{6}$$

Consequently, an often-used MKL model from binary-class data $\{(\mathbf{x}_i, y_i \in \pm 1)\}_{i=1}^N$ is

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{7}$$

$$= \sum_{i=1}^N \alpha_i y_i \sum_{m=1}^M \beta_m k_m(\mathbf{x}_i, \mathbf{x}) + b. \tag{8}$$

Optimizing over both the coefficients $\{\alpha_i\}_{i=1}^N$ and $\{\beta_m\}_{m=1}^M$ is one particular form of the MKL problems. Our approach utilizes such an MKL optimization to yield more flexible dimensionality reduction schemes for data in different feature representations.

2.4 Dimensionality Reduction with Multiple Kernels

Our approach is related to the work of Kim et al. [27], where learning an optimal kernel over a given convex set of kernels is coupled with *kernel Fisher discriminant analysis* (KFDA) for binary-class data. Motivated by their idea of learning an optimal kernel for improving the KFDA performance, we instead consider establishing a general framework of dimensionality reduction for data in various feature representations via multiple kernel learning [32]. As we will show later, MKL-DR can be used to conveniently deal with image data depicted by different descriptors, and effectively tackle not only supervised but also semi-supervised and unsupervised learning tasks. To the best of our knowledge, such a generalization of multiple kernel learning is novel.

3 The MKL-DR Framework

We first discuss the construction of base kernels from multiple descriptors, and then explain how to integrate them for dimensionality reduction. Finally, we present an optimization procedure to complete the framework.

3.1 Kernel as a Unified Feature Representation

Consider again a data set Ω of N samples, and M kinds of descriptors to characterize each sample. Let $\Omega = \{\mathbf{x}_i\}_{i=1}^{N}$, $\mathbf{x}_i = \{\mathbf{x}_{i,m} \in \mathcal{X}_m\}_{m=1}^{M}$, and $d_m : \mathcal{X}_m \times \mathcal{X}_m \rightarrow 0 \cup \mathbb{R}^+$ be the distance function for data representation under the mth descriptor. In general, the domains resulting from distinct descriptors, e.g., feature vectors, histograms, or bags of features, are different. To eliminate such variances in representation, we express data under each descriptor as a kernel matrix. There are several ways to accomplish this goal, such as using the RBF kernel for data in the form of vector or the pyramid match kernel [21] for data in the form of bag-of-features. We may also convert pairwise distances between data samples to a kernel matrix [50], [58]. By coupling each representation with its corresponding distance function, we obtain a set of M *dissimilarity-based* kernel matrices $\{K_m\}_{m=1}^{M}$, where

$$K_m(i,j) = k_m(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-d_m^2(\mathbf{x}_{i,m}, \mathbf{x}_{j,m})}{\sigma_m^2}\right) \tag{9}$$

and σ_m is a positive constant. Our use of dissimilarity-based kernels is convenient and advantageous in solving visual learning tasks, especially due to the fact that a number of well-designed descriptors and their associated distance functions have been introduced over the years. However, K_m in (9) is not always guaranteed to be positive semidefinite. Following [58], we resolve this issue by first computing the smallest eigenvalue of K_m. Then, if it is negative, we add its absolute value to the diagonal of K_m. With (5), (6), and (9), determining a set of optimal ensemble coefficients $\{\beta_1, \beta_2, \ldots, \beta_M\}$ can now be interpreted as finding appropriate weights for best fusing the M feature representations.

Note that in our formulation, accessing the data is restricted to referencing the resulting M kernels defined in (9). The main advantage of doing so is that it enables our approach to work with different descriptors and distance functions, without the need to explicitly handle the variations among the representations.

3.2 The MKL-DR Algorithm

Instead of designing a specific dimensionality reduction algorithm, we choose to describe MKL-DR upon graph embedding. This way we can emphasize the flexibility of the proposed approach: If a dimensionality reduction scheme is explained by graph embedding, then it will also be extendible by MKL-DR to handle data in multiple feature representations. Recall that there are two possible types of constraints in graph embedding. For ease of presentation, we discuss how to develop MKL-DR subject to constraint (4). However, the derivation can be analogously applied when using constraint (3).

Kernelization in MKL-DR is accomplished in a similar way to that in kernel PCA [41], but with the key difference in using multiple kernels $\{K_m\}_{m=1}^{M}$. Suppose the ensemble kernel K in MKL-DR is generated by linearly combining the base kernels $\{K_m\}_{m=1}^{M}$ as in (6). Let $\phi : \mathcal{X} \rightarrow \mathcal{F}$ denote the feature mapping induced by K. Via ϕ, the training data can be implicitly mapped to a high-dimensional Hilbert space, i.e.,

$$\mathbf{x}_i \mapsto \phi(\mathbf{x}_i), \quad \text{for } i = 1, 2, \ldots, N. \tag{10}$$

Since optimizing (1) or (2) can be reduced to solving the eigenvalue problem $XLX^\top \mathbf{v} = \lambda XL'X^\top \mathbf{v}$, it implies that an optimal $\mathbf{v}$ lies in the span of training data, i.e.,

$$\mathbf{v} = \sum_{n=1}^{N} \alpha_n \phi(\mathbf{x}_n). \tag{11}$$

To show that the underlying algorithm can be reformulated in the form of inner product and accomplished in the new feature space $\mathcal{F}$, we observe that by plugging each mapped sample $\phi(\mathbf{x}_i)$ into (2), projection $\mathbf{v}$ would appear exclusively in the form of $\mathbf{v}^\top \phi(\mathbf{x}_i)$. Hence, it suffices to show that in MKL-DR, $\mathbf{v}^\top \phi(\mathbf{x}_i)$ can be evaluated via the kernel trick

$$\mathbf{v}^\top \phi(\mathbf{x}_i) = \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_n \beta_m k_m(\mathbf{x}_n, \mathbf{x}_i) = \boldsymbol{\alpha}^\top \mathbb{K}^{(i)} \boldsymbol{\beta}, \tag{12}$$

where

$$\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_N]^\top \in \mathbb{R}^N, \tag{13}$$

$$\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_M]^\top \in \mathbb{R}^M, \tag{14}$$

$$\mathbb{K}^{(i)} = \begin{bmatrix} K_1(1,i) & \cdots & K_M(1,i) \\ \vdots & \ddots & \vdots \\ K_1(N,i) & \cdots & K_M(N,i) \end{bmatrix} \in \mathbb{R}^{N \times M}. \tag{15}$$

With (2) and (12), we define the constrained optimization problem for *1D* MKL-DR as follows:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \sum_{i,j=1}^{N} \|\boldsymbol{\alpha}^\top \mathbb{K}^{(i)} \boldsymbol{\beta} - \boldsymbol{\alpha}^\top \mathbb{K}^{(j)} \boldsymbol{\beta}\|^2 w_{ij} \tag{16}$$

$$\text{subject to} \sum_{i,j=1}^{N} \|\boldsymbol{\alpha}^\top \mathbb{K}^{(i)} \boldsymbol{\beta} - \boldsymbol{\alpha}^\top \mathbb{K}^{(j)} \boldsymbol{\beta}\|^2 w'_{ij} = 1, \tag{17}$$

$$\beta_m \geq 0, m = 1, 2, \ldots, M. \tag{18}$$

The additional constraints in (18) arise from the use of the ensemble kernel in (5) or (6), and are to ensure that the resulting kernel K in MKL-DR is a nonnegative combination of base kernels.

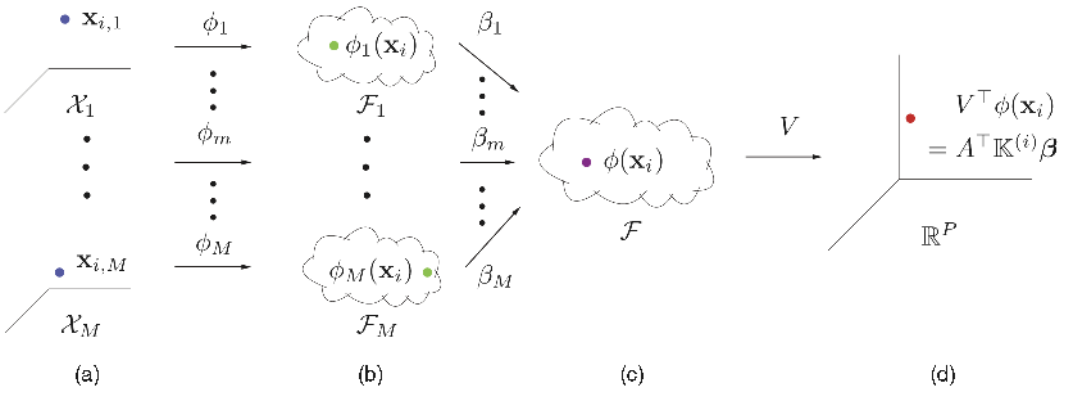


Fig. 1. Four kinds of spaces in MKL-DR: (a) the input space of each feature representation, (b) the RKHS induced by each base kernel, (c) the RKHS by the ensemble kernel, and (d) the projected euclidean space.

Observe from (12) that the one-dimensional projection $\mathbf{v}$ of MKL-DR is specified by a *sample coefficient vector* $\boldsymbol{\alpha}$ and a *kernel weight vector* $\boldsymbol{\beta}$. The two vectors, respectively, account for the relative importance among the samples and the base kernels in the construction of the projection. To generalize the formulation to uncover a multidimensional projection, we consider a set of P sample coefficient vectors, denoted by

$$A = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2\ \cdots\ \boldsymbol{\alpha}_P]. \tag{19}$$

With A and $\boldsymbol{\beta}$, each 1D projection $\mathbf{v}_i$ is determined by a specific sample coefficient vector $\boldsymbol{\alpha}_i$ and the (shared) kernel weight vector $\boldsymbol{\beta}$. The resulting projection $V = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_P]$ will map samples to a P-dimensional euclidean space. Analogously to the 1D case, a projected sample $\mathbf{x}_i$ can be written as

$$V^\top\phi(\mathbf{x}_i) = A^\top \mathbb{K}^{(i)}\boldsymbol{\beta} \in \mathbb{R}^P. \tag{20}$$

The optimization problem (16) can now be extended to accommodate the multidimensional projection

$$\min_{A,\boldsymbol{\beta}} \quad \sum_{i,j=1}^{N} \|A^\top \mathbb{K}^{(i)}\boldsymbol{\beta} - A^\top \mathbb{K}^{(j)}\boldsymbol{\beta}\|^2 w_{ij} \tag{21}$$

$$\text{subject to} \quad \sum_{i,j=1}^{N} \|A^\top \mathbb{K}^{(i)}\boldsymbol{\beta} - A^\top \mathbb{K}^{(j)}\boldsymbol{\beta}\|^2 w'_{ij} = 1, \tag{22}$$

$$\beta_m \geq 0, \quad m = 1, 2, \ldots, M. \tag{23}$$

Before specifying the details of how to solve the constrained optimization problem (21) in the next section, we give an illustration of the four kinds of spaces related to MKL-DR and the connections among them in Fig. 1. The four spaces, in order, are the input space of each feature representation, the reproducing kernel Hilbert space (RKHS) induced by each base kernel and the ensemble kernel, and the projected euclidean space.

3.3 Optimization

Since direct optimization to (21) is difficult, we instead adopt an iterative, two-step strategy to alternately optimize A and $\boldsymbol{\beta}$. At each iteration, one of A and $\boldsymbol{\beta}$ is optimized while the other is fixed, and then the roles of A and $\boldsymbol{\beta}$ are switched. Iterations are repeated until convergence or a maximum number of iterations is reached.

On optimizing A. By fixing $\boldsymbol{\beta}$ and using the property $\|\mathbf{u}\|^2 = \text{trace}(\mathbf{u}\mathbf{u}^\top)$ for a column vector $\mathbf{u}$, the optimization problem (21) is reduced to

$$\begin{aligned} \min_{A} \quad & \text{trace}(A^\top S_W^{\boldsymbol{\beta}} A) \\ \text{subject to} \quad & \text{trace}(A^\top S_{W'}^{\boldsymbol{\beta}} A) = 1, \end{aligned} \tag{24}$$

where

$$S_W^{\boldsymbol{\beta}} = \sum_{i,j=1}^{N} w_{ij}(\mathbb{K}^{(i)} - \mathbb{K}^{(j)})\boldsymbol{\beta}\boldsymbol{\beta}^\top(\mathbb{K}^{(i)} - \mathbb{K}^{(j)})^\top, \tag{25}$$

$$S_{W'}^{\boldsymbol{\beta}} = \sum_{i,j=1}^{N} w'_{ij}(\mathbb{K}^{(i)} - \mathbb{K}^{(j)})\boldsymbol{\beta}\boldsymbol{\beta}^\top(\mathbb{K}^{(i)} - \mathbb{K}^{(j)})^\top. \tag{26}$$

The optimization problem (24) is a *trace ratio* problem, i.e., $\min_A \text{trace}(A^\top S_W^{\boldsymbol{\beta}} A)/\text{trace}(A^\top S_{W'}^{\boldsymbol{\beta}} A)$. Following [9] and [52], one can obtain a closed-form solution by transforming (24) into the corresponding *ratio trace* problem, i.e., $\min_A \text{trace}[(A^\top S_{W'}^{\boldsymbol{\beta}} A)^{-1}(A^\top S_W^{\boldsymbol{\beta}} A)]$. Consequently, the columns of the optimal $A^* = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2 \cdots \boldsymbol{\alpha}_P]$ are the eigenvectors corresponding to the first P smallest eigenvalues in

$$S_W^{\boldsymbol{\beta}}\boldsymbol{\alpha} = \lambda S_{W'}^{\boldsymbol{\beta}}\boldsymbol{\alpha}. \tag{27}$$

On optimizing $\boldsymbol{\beta}$. By fixing A and $\|\mathbf{u}\|^2 = \mathbf{u}^\top\mathbf{u}$, the optimization problem (21) becomes

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \boldsymbol{\beta}^\top S_W^A \boldsymbol{\beta} \\ \text{subject to} \quad & \boldsymbol{\beta}^\top S_{W'}^A \boldsymbol{\beta} = 1 \quad \text{and} \quad \boldsymbol{\beta} \geq \mathbf{0}, \end{aligned} \tag{28}$$

where

$$S_W^A = \sum_{i,j=1}^{N} w_{ij}(\mathbb{K}^{(i)} - \mathbb{K}^{(j)})^\top AA^\top(\mathbb{K}^{(i)} - \mathbb{K}^{(j)}), \tag{29}$$

$$S_{W'}^A = \sum_{i,j=1}^{N} w'_{ij}(\mathbb{K}^{(i)} - \mathbb{K}^{(j)})^\top AA^\top(\mathbb{K}^{(i)} - \mathbb{K}^{(j)}). \tag{30}$$

The additional constraints $\boldsymbol{\beta} \geq \mathbf{0}$ cause the optimization to (28) no longer be formulatable as a generalized eigenvalue

Algorithm 1: *The Training Procedure of MKL-DR*

Input : A DR method specified by two affinity matrices W and W' (cf. (2));
Data described by various visual features in form of base kernels $\{K_m\}_{m=1}^{M}$ (cf. (9));

Output: Sample coefficient vectors $A = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2 \cdots \boldsymbol{\alpha}_P]$;
Kernel weight vector $\boldsymbol{\beta}$;

Make an initial guess for A or $\boldsymbol{\beta}$;
for $t \leftarrow 1, 2, \ldots, T$ **do**
1. Compute $S_W^{\boldsymbol{\beta}}$ in (25) and $S_{W'}^{\boldsymbol{\beta}}$ in (26);
2. A is optimized by solving the generalized eigenvalue problem (27);
3. Compute S_W^A in (29) and $S_{W'}^A$ in (30);
4. $\boldsymbol{\beta}$ is optimized by solving optimization problem (31) via semidefinite programming;

return A and $\boldsymbol{\beta}$;

Fig. 2. Algorithm 1.

problem. Indeed, it now becomes a nonconvex *quadratically constrained quadratic programming* (QCQP) problem, and is known to be hard to solve. We instead consider solving its convex relaxation by adding an auxiliary variable B of size $M \times M$:

$$\min_{\boldsymbol{\beta}, B} \quad \operatorname{trace}(S_W^A B) \tag{31}$$

$$\text{subject to} \quad \operatorname{trace}(S_{W'}^A B) = 1, \tag{32}$$

$$\mathbf{e}_m^\top \boldsymbol{\beta} \geq 0, \quad m = 1, 2, \ldots, M, \tag{33}$$

$$\begin{bmatrix} 1 & \boldsymbol{\beta}^\top \\ \boldsymbol{\beta} & B \end{bmatrix} \succeq 0, \tag{34}$$

where $\mathbf{e}_m$ in (33) is a column vector whose elements are 0 except that its mth element is 1, and the constraint in (34) means that the square matrix is a positive semidefinite. The optimization problem (31) is a *semidefinite programming (SDP) relaxation* of the nonconvex QCQP problem (28), and can be efficiently solved by SDP. One can verify the equivalence between the two optimization problems (28) and (31) by replacing the constraint (34) with $B = \boldsymbol{\beta}\boldsymbol{\beta}^\top$. In view of that the constraint $B = \boldsymbol{\beta}\boldsymbol{\beta}^\top$ is nonconvex, it is relaxed to $B \succeq \boldsymbol{\beta}\boldsymbol{\beta}^\top$. Applying the Schur complement lemma, $B \succeq \boldsymbol{\beta}\boldsymbol{\beta}^\top$ can be equivalently expressed by the constraint in (34). (Refer to [49] for the details.) Concerning the computational complexity, we note that the numbers of constraints and variables in (31) are, respectively, linear and quadratic to M, the number of the adopted descriptors. In practice, the value of M is often small. ($M = 4 \sim 10$ in our experiments.) Thus, like most of the other DR methods, the computational bottleneck of MKL-DR is still in solving the generalized eigenvalue problems, whose complexity is $\mathcal{O}(N^3)$.

Listed in Algorithm 1 (Fig. 2), the procedure of MKL-DR requires an initial guess to either A or $\boldsymbol{\beta}$ in the alternating optimization. We have tried two possibilities: 1) $\boldsymbol{\beta}$ is initialized by setting all of its elements as 1 to equally weight base kernels; 2) A is initialized by assuming $AA^\top = I$. In our empirical testing, the second initialization strategy gives more stable performances and is thus adopted in the experiments. Pertaining to the convergence of the optimization procedure, since SDP relaxation has been used, the values of the objective function are not guaranteed to monotonically decrease throughout the iterations. Still, the optimization procedures rapidly converge after only a few iterations in all of our experiments.

3.4 Novel Sample Embedding

After accomplishing the training procedure of MKL-DR, we are ready to project a testing sample, say $\mathbf{z}$, into the learned space of lower dimension by

$$\mathbf{z} \mapsto A^\top \mathbb{K}^{(\mathbf{z})} \boldsymbol{\beta}, \quad \text{where} \tag{35}$$

$$\mathbb{K}^{(\mathbf{z})} \in \mathbb{R}^{N \times M} \quad \text{and} \quad \mathbb{K}^{(\mathbf{z})}(n, m) = k_m(\mathbf{x}_n, \mathbf{z}). \tag{36}$$

Depending on the applications, some postprocessing, such as the nearest neighbor rule for classification or k-means clustering for data grouping, is then applied to the projected sample(s) to complete the task. In the remainder of this paper, we specifically discuss three sets of experimental results to demonstrate the effectiveness of MKL-DR, including supervised learning for object categorization, unsupervised learning for image clustering, and semi-supervised learning for face recognition.

4 Experimental Results: Supervised Learning for Object Categorization

Applying MKL-DR to object categorization is appropriate as the complexity of the task often requires the use of multiple feature descriptors. And in our experiments, the effectiveness of MKL-DR will be investigated through a supervised learning formulation.

4.1 Data Set

The Caltech-101 data set [14], collected by Fei-Fei et al., is used in our experiments for object categorization. It consists of 101 object categories and one additional class of background images. The total number of categories is 102, and each category contains roughly 40 to 800 images. Although each target object often appears in the central region of an image, the large class number and the substantial intraclass variations still make the data set very challenging. Indeed, the data set provides a good test bed to demonstrate the advantage of using multiple image descriptors for complex recognition tasks. Note that as the

Fig. 3. The Caltech-101 data set. One example comes from each of the 102 categories. All of the 102 categories are used in the experiments of supervised object recognition, while the 20 categories marked by the red bounding boxes are used in the following experiments of unsupervised image clustering.

images in Caltech-101 are not of the same size, we resize them to around 60,000 pixels, without changing their aspect ratio. Fig. 3 shows an image example from each category of the data set.

To implement Algorithm 1 for object recognition, we need to decide a set of descriptors for depicting the diverse objects and the underlying graph-based DR method to be generalized. Based on them, we can then derive a set of base kernels and a pair of affinity matrices, respectively. The details are described as follows.

4.2 Image Descriptors and Base Kernels

Ten different image descriptors are considered and they, respectively, yield the following base kernels (denoted below in bold and in abbreviation):

- **GB-Dist**: For a given image, we randomly sample 400 edge pixels, and apply *geometric blur* descriptor [3] to them. With these image features, we adopt the distance function, as is suggested in (2) of the work by Zhang et al. [58], to obtain the dissimilarity-based kernel.
- **GB**: The base kernel is constructed in the same way to that of GB-Dist, except that the geometric distortion term is excluded in evaluating the distance.
- **SIFT-Dist**: The base kernel is analogously constructed as in GB-Dist, except that now the *SIFT* descriptor [33] is used to extract features.
- **SIFT-SPM**: We apply the SIFT descriptor with three different scales to an evenly sampled grid of each image, and use k-means clustering to generate *visual words* from the resulting local features of all images. Then, the base kernel is built by matching *spatial pyramids*, which is proposed in [30].
- **SS-Dist/SS-SPM**: The two base kernels are, respectively, constructed as in SIFT-Dist and SIFT-SPM, except that the SIFT descriptor is replaced with the *self-similarity* descriptor [43]. Note that for the latter descriptor, we set the size of each patch to 5×5, and the radius of the window to 40.
- **C2-SWP/C2-ML**: Biologically inspired features are also adopted. Specifically, both the *C2* features derived by Serre et al. [42] and by Mutch and Lowe [35] have been considered. For each of the two kinds of C2 features, an RBF kernel is obtained.
- **PHOG**: We also adopt the *PHOG* descriptor [5] to capture image features, and limit the pyramid level up to 2. Together with the χ^2 distance, it yields the resulting base kernel.
- **GIST**: The images are resized to 128×128 pixels prior to applying the *gist* descriptor [37]. Then, an RBF kernel is established.

The parameters in the above descriptors and distance functions are tuned independently. Namely, for each descriptor, we sample a set of parameter values and try to find a *good* way to linearly combine the corresponding pairwise distance matrices. To that end, we begin with an initial weight distribution focusing solely on the one yielding the best performance. We then separately and sequentially adjust each individual weight, and repeat the process until no further improvement can be attained. Such a scheme is to ensure that the resulting base kernels individually achieve their best performances.

4.3 Dimensionality Reduction Methods

We investigate two supervised DR techniques, namely, LDA and LDE [9], and show how MKL-DR can generalize them. Both LDA and LDE perform discriminant learning on a fully labeled data set $\Omega = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, but make different assumptions about the data distribution: While, in LDA, data of each class are supposed to form a Gaussian, in LDE, they are assumed to spread as a submanifold. Nevertheless, both techniques can be specified by a pair of affinity matrices to fit the formulation of graph embedding (2). For convenience, the resulting MKL dimensionality reduction schemes are, respectively, termed as *MKL-LDA* and *MKL-LDE*.

4.3.1 Affinity Matrices for LDA

The two affinity matrices $W = [w_{ij}]$ and $W' = [w'_{ij}]$ are defined as

$$w_{ij} = \begin{cases} 1/n_{y_i}, & \text{if } y_i = y_j, \\ 0, & \text{otherwise}, \end{cases} \tag{37}$$

$$w'_{ij} = \frac{1}{N}, \tag{38}$$

TABLE 1
Recognition Rates of LDA-Based Classifiers on the Caltech-101 Data Set

method	kernel(s)	number of training data per class (N_{train})				
		5	10	15	20	25
Kernel LDA (KFD)	GB-Dist	44.0 ± 1.1 %	55.2 ± 0.9 %	60.6 ± 1.1 %	65.1 ± 1.1 %	69.7 ± 1.6 %
	GB	40.7 ± 1.2 %	51.1 ± 0.8 %	56.2 ± 0.6 %	62.0 ± 1.4 %	66.0 ± 1.1 %
	SIFT-Dist	36.6 ± 1.1 %	46.7 ± 0.5 %	53.8 ± 0.9 %	57.8 ± 1.1 %	63.1 ± 1.9 %
	SIFT-SPM	34.9 ± 0.6 %	44.3 ± 1.0 %	50.3 ± 0.6 %	55.1 ± 1.2 %	60.0 ± 2.4 %
	SS-Dist	35.3 ± 1.0 %	44.9 ± 0.8 %	51.1 ± 1.0 %	56.0 ± 1.6 %	61.5 ± 2.3 %
	SS-SPM	37.0 ± 0.7 %	47.8 ± 0.6 %	55.0 ± 0.9 %	59.1 ± 0.9 %	64.0 ± 2.0 %
	C2-SWP	18.3 ± 0.9 %	24.9 ± 0.6 %	29.6 ± 0.7 %	34.3 ± 1.1 %	37.7 ± 0.8 %
	C2-ML	30.4 ± 0.9 %	39.2 ± 0.6 %	45.1 ± 0.6 %	48.1 ± 0.7 %	52.8 ± 2.0 %
	PHOG	27.6 ± 0.8 %	34.4 ± 0.8 %	40.7 ± 0.6 %	42.5 ± 0.8 %	46.5 ± 1.5 %
	GIST	33.2 ± 0.8 %	42.1 ± 0.9 %	47.0 ± 0.7 %	51.5 ± 1.0 %	55.6 ± 1.6 %
KFD-*Voting*	-	54.4 ± 0.7 %	65.7 ± 0.8 %	69.8 ± 0.7 %	73.9 ± 1.1 %	76.8 ± 1.6 %
KFD-*Concatenate*	-	55.4 ± 1.2 %	65.6 ± 0.9 %	71.7 ± 0.8 %	75.3 ± 0.8 %	78.2 ± 1.3 %
KFD-*AvgKernel*	-	55.9 ± 1.0 %	66.1 ± 0.6 %	72.0 ± 1.0 %	75.6 ± 0.7 %	78.2 ± 1.5 %
KFD-*SAMME*	-	56.4 ± 1.1 %	67.4 ± 0.9 %	72.3 ± 0.6 %	75.7 ± 1.0 %	77.7 ± 2.1 %
MKL-LDA	All	**58.4 ± 1.0 %**	**68.8 ± 0.9 %**	**74.5 ± 1.0 %**	**77.5 ± 1.0 %**	**79.8 ± 1.7 %**

[Mean ± std] *percent.*

where n_{y_i} is the number of data points that belong to class y_i. See [54] for the derivation.

4.3.2 Affinity Matrices for LDE

In LDE, not only the data labels but also the neighborhood relationships are simultaneously considered, namely,

$$w_{ij} = \begin{cases} 1, & \text{if } y_i = y_j \wedge [i \in \mathcal{N}_k(j) \vee j \in \mathcal{N}_k(i)], \\ 0, & \text{otherwise}, \end{cases} \tag{39}$$

$$w'_{ij} = \begin{cases} 1, & \text{if } y_i \neq y_j \wedge [i \in \mathcal{N}_{k'}(j) \vee j \in \mathcal{N}_{k'}(i)], \\ 0, & \text{otherwise}, \end{cases} \tag{40}$$

where $i \in \mathcal{N}_k(j)$ means that sample $\mathbf{x}_i$ is one of the k nearest neighbors of sample $\mathbf{x}_j$. The definitions of the affinity matrices are faithful to those in LDE [9]. However, there are now multiple image descriptors and each of them would yield an affinity matrix. Since we typically do not know/assume in advance which would be more important to a given task, we simply average the resulting affinity matrices to derive a unified one.

4.4 Quantitative Results

Like in [2], [50], [58], we randomly pick 30 images from each of the 102 categories, and split them into two disjoint subsets: One contains N_{train} images per category, and the other consists of the rest. The two subsets are, respectively, used as the training and testing data. Via MKL-DR, the data are projected to the learned space, and the recognition task is accomplished there by enforcing the *nearest-neighbor* rule. To relieve the effect of sampling, the whole process of performance evaluation is redone 20 times by using different random splits between the training and testing subsets. The recognition rates are measured in the cases where the value of N_{train} is, respectively, set as 5, 10, 15, 20, and 25.

Coupling the 10 base kernels with the affinity matrices of LDA and LDE, we can, respectively, derive MKL-LDA and MKL-LDE using Algorithm 1. Their effectiveness is investigated by comparing with KFD (kernel Fisher discriminant) [34] and KLDE (kernel LDE) [9]. Since KFD considers only one base kernel at a time, we implement four strategies to take account of the information from the 10 resulting KFD classifiers, including

1. KFD-*Voting*: It is constructed based on the voting result of the 10 KFD classifiers. If there is any ambiguity in the voting result, the next nearest neighbor in each KFD classifier will be considered, and the process is continued until a decision on the class label can be made.
2. KFD-*Concatenate*: For each sample, we concatenate its separately learned feature vectors, each of which is normalized by dividing the standard deviation of the pairwise distances among the projected training data.
3. KFD-*AvgKernel*: KFD is reapplied to the average kernel of the 10 base ones.
4. KFD-*SAMME*: By viewing each KFD classifier as a multiclass weak learner, we boost them by *SAMME* [59], which is a multiclass generalization of AdaBoost.

Analogously, the four strategies are also adopted for the KLDE classifiers.

The values of parameters $\{\sigma_m\}$ in (9) are critical to the performance of MKL-DR. However, it is almost infeasible to find their optimal values exhaustively. We instead adopt the following procedure that would give satisfactory results. Observe that the larger the σ_m, the more evenly the entries in K_m would distribute. Fixing some values of, say, s and t, we adjust the value of σ_m by *binary search* such that the largest s entries in K_m will take up t percent of the sum of all entries. Given s and t, the values of $\{\sigma_m\}$ can thus be determined. We set s as a constant and exhaustively seek the optimal t. The resulting $\{\sigma_m\}$ will then serve as the initialization to the procedure described in the end of Section 4.2.

Table 1 summarizes the mean recognition rates and the standard deviations of KFD classifiers and MKL-LDA classifiers when different amounts of training data are available. By focusing on $N_{train} = 15$, we observe that MKL-LDA achieves a significant performance gain of 13.9 percent $(=74.5\% - 60.6\%)$ over the best recognition rate

TABLE 2
Recognition Rates of LDE-Based Classifiers on the Caltech-101 Data Set

method	kernel(s)	number of training data per class (N_{train})				
		5	10	15	20	25
Kernel LDE (KLDE)	GB-Dist	44.5 ± 1.1 %	54.8 ± 0.7 %	60.7 ± 1.1 %	66.2 ± 1.4 %	69.7 ± 1.3 %
	GB	41.0 ± 1.3 %	51.3 ± 1.1 %	55.9 ± 0.5 %	62.2 ± 1.2 %	66.3 ± 1.3 %
	SIFT-Dist	37.0 ± 1.1 %	46.6 ± 0.7 %	53.4 ± 0.7 %	58.0 ± 1.4 %	63.7 ± 2.1 %
	SIFT-SPM	35.3 ± 0.8 %	44.4 ± 0.9 %	50.2 ± 0.7 %	55.0 ± 1.0 %	61.3 ± 2.4 %
	SS-Dist	35.1 ± 1.0 %	44.1 ± 0.8 %	50.6 ± 1.0 %	55.2 ± 1.1 %	61.2 ± 1.9 %
	SS-SPM	37.1 ± 0.7 %	47.1 ± 0.4 %	55.4 ± 1.3 %	59.2 ± 1.0 %	64.1 ± 1.9 %
	C2-SWP	18.7 ± 0.9 %	24.9 ± 0.5 %	29.7 ± 0.7 %	34.3 ± 1.1 %	38.2 ± 1.1 %
	C2-ML	30.7 ± 0.9 %	39.5 ± 0.8 %	45.9 ± 0.8 %	48.3 ± 0.6 %	54.2 ± 2.2 %
	PHOG	28.0 ± 1.0 %	34.8 ± 0.9 %	39.5 ± 0.8 %	42.3 ± 1.0 %	46.1 ± 1.0 %
	GIST	33.4 ± 0.8 %	41.8 ± 0.7 %	47.0 ± 0.6 %	51.5 ± 0.9 %	56.7 ± 1.5 %
KLDE-*Voting*	-	53.9 ± 1.2 %	64.3 ± 0.8 %	70.3 ± 1.0 %	74.3 ± 0.7 %	76.6 ± 1.6 %
KLDE-*Concatenate*	-	55.4 ± 0.9 %	65.7 ± 0.7 %	71.4 ± 0.9 %	75.5 ± 0.9 %	78.5 ± 1.4 %
KLDE-*AvgKernel*	-	55.8 ± 0.7 %	66.3 ± 0.7 %	71.8 ± 0.9 %	75.7 ± 0.8 %	78.4 ± 1.5 %
KLDE-*SAMME*	-	56.1 ± 1.3 %	66.8 ± 0.7 %	72.6 ± 1.0 %	75.7 ± 0.9 %	77.8 ± 1.3 %
MKL-LDE	All	**59.2 ± 1.5 %**	**68.9 ± 0.8 %**	**74.9 ± 1.1%**	**77.2 ± 0.9%**	**79.2 ± 1.6%**

[Mean ± std] *percent.*

by the 10 KFD classifiers. It suggests that the 10 base kernels tend to complement each other, and our approach can effectively fuse them to result in a more powerful classifier. On the other hand, while KFD-*Voting*, KFD-*Concatenate*, and KFD-*SAMME* try to combine the *separately* trained KFD classifiers, MKL-LDA *jointly* integrates the 10 kernels into the learning process. The quantitative results show that MKL-LDA can make the most of fusing various feature descriptors, and improve the recognition rates from 69.8, 71.7, and 72.3 percent to 74.5 percent. Besides, MKL-LDA outperforms KFD-*AvgKernel*. That is, the ensemble kernel based on the learned kernel weight vector $\boldsymbol{\beta}$ by MKL-DR is more effective than the average kernel. Similar improvements can be observed in cases where different numbers of training data per class are used.

The quantitative results of KLDE and MKL-LDE are reported in Table 2. Like MKL-LDA, MKL-LDE achieves similar degrees of improvements over the KLDE classifiers and their combinations. In addition, we explore the effect of the dimensions of the unified feature space by MKL-DR. Illustrated with MKL-LDE and KLDE, we evaluate their recognition rates over a range of embedding dimensions, i.e., P in (20). The results are plotted in Fig. 4. We see that the recognition rates by MKL-LDE and KLDE all converge around the dimensions of $90 \sim 110$. Compared with KLDE, MKL-LDE can achieve similar degrees of accuracy with fewer dimensions.

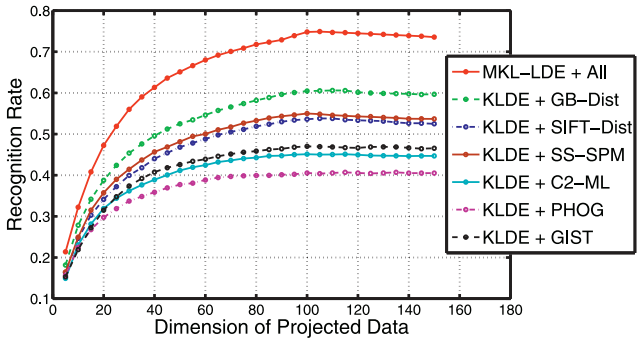


Fig. 4. Recognition rates versus different dimensions of the projected data when $N_{train} = 15$.

When the number of training data per class from Caltech-101 is set as 15, the recognition rate of 74.5 percent by MKL-LDA and 74.9 percent by MKL-LDE are favorably comparable to those by most existing approaches. In [2], Berg et al. report a recognition rate of 48 percent based on deformable shape matching. Using the pyramid matching kernel over data in the bag-of-features representation, the recognition rate by Grauman and Darrell [21] is 50 percent. Subsequently, Lazebnik et al. [30] improve it to 56.4 percent by considering the spatial pyramid matching kernel. In [58], Zhang et al. combine the geometric blur descriptor and spatial information to achieve 59.05 percent. Our related work [31] that performs adaptive feature fusing via locally combining kernel matrices has a recognition rate of 59.8 percent, while merging 12 kernel matrices from the *support kernel machines* (SKMs) [1] by Kumar and Sminchisescu [28] yields 57.3 percent. Frome et al. [16] propose to learn a local distance for each training sample, and derive 60.3 percent. To tackle the weakly labeled attribute of Caltech-101, Bosch et al. [5] suggest finding the *ROIs* of images before performing recognition, and report an accuracy rate of 70.4 percent. By exploring the features from *subcategories*, Todorovic and Ahuja [48] report a recognition rate around 73 percent. Similar results are obtained by Christoudias et al. [10] using localized Gaussian processes with multiple kernels. Gehler and Nowozin [19] carry out multiple kernel learning in a boosting manner, and achieve 74.6 percent. In Fig. 5, we summarize the recognition rates of our approach and several published techniques, including [2], [4], [5], [14], [17], [19], [21], [24], [26], [30], [35], [42], [55], [58], under different sizes of training data.

We complete the section by discussing the convergence property of our algorithm. Take, for example, learning MKL-LDA with different sizes of training data per class. The values of the objective function (21) through the iterative optimization are, respectively, shown in Figs. 6a, 6b, 6c, 6d, and 6e. In each iteration, two such values are plotted to account for updating either A or $\boldsymbol{\beta}$. It can be

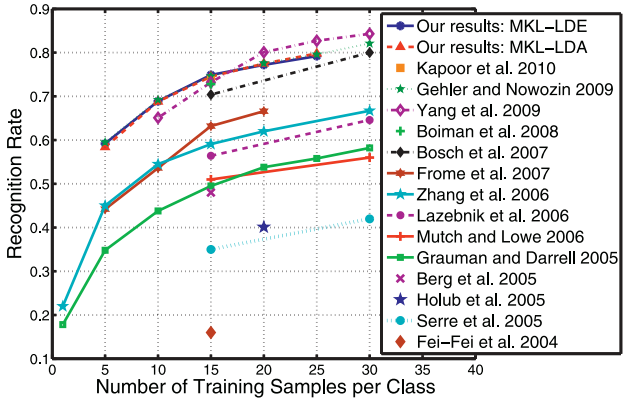


Fig. 5. Recognition rates of several published systems on Caltech-101 versus different amounts of training data.

observed that all of the optimization procedures rapidly converge after a few iterations. Also, increasing the size of training data tends to speed up the convergence, which is reasonable since sufficient information generally facilitates solving an optimization task.

5 Experimental Results: Unsupervised Learning for Image Clustering

To explain the link between MKL-DR and unsupervised learning, we investigate the problem of image clustering. In this case, MKL-DR can be viewed as a *preprocessing* tool to enrich the capacity of an existing clustering technique. There are two main advantages for so doing. First, since MKL-DR can learn a unified space for image data in multiple representations, it enables the underlying clustering algorithm to simultaneously consider characteristics captured by distinct descriptors. Second, a majority of clustering algorithms, e.g., k-means, are designed to work only in the euclidean space. With MKL-DR, no matter what the original spaces the data reside in, they all can be projected to the learned euclidean space, and consequently our formulation can extend the applicability of such a clustering method.

5.1 Data Set

We follow the setting in [12], where *affinity propagation* [15] is used for unsupervised image categorization, and select the same 20 categories from Caltech-101 for the image clustering experiments. Examples from the 20 image categories are shown in Fig. 3, and each is marked with a bold red bounding box. Due to the category-wise differences in the number of images, we randomly select 30 images from each category to form a data set of 600 images.

5.2 Image Descriptors and Base Kernels

Since the data set is now a subset of Caltech-101, it is convenient to use the same 10 descriptors and distance functions that are discussed in Section 4.2 to establish the base kernels for MKL-DR.

5.3 Dimensionality Reduction Method

For image clustering, we consider implementing MKL-DR with LPP [23], and denote it as MKL-LPP. The LPP technique is known to be an unsupervised DR scheme that can uncover a low-dimensional subspace by preserving the neighborhood structures. The property is particularly useful since respecting the locality information often plays a key factor in the clustering outcomes. To carry out MKL-LPP, we need to reduce LPP to the formulation of graph embedding (2). This is accomplished by defining $W = [w_{ij}]$ and $D = [d_{ij}]$ as

$$w_{ij} = \begin{cases} 1, & \text{if } i \in \mathcal{N}_k(j) \vee j \in \mathcal{N}_k(i), \\ 0, & \text{otherwise}, \end{cases} \tag{41}$$

$$d_{ij} = \begin{cases} \sum_{n=1}^{N} w_{in}, & \text{if } i = j, \\ 0, & \text{otherwise}. \end{cases} \tag{42}$$

Note that LPP is specified by an affinity matrix W and a diagonal matrix D, instead of a pair of affinity matrices W and W'. In Section 3.3, although we only discuss how to derive MKL-DR with DR methods that can be expressed by a pair of W and W', the derivation for those by a pair of W and D is indeed analogous, and the details are omitted here for the sake of space.

5.4 Quantitative Results

Coupling LPP with the base kernels, MKL-LPP would project the given image data to a learned space, where clustering algorithms will be performed. In the experiments, we restrict the number of clusters to the number of classes in the data set, i.e., 20, for all of the tested clustering algorithms, and evaluate their performances with the following two criteria: *normalized mutual information* (NMI) [46], and *clustering accuracy* (ACC). (Refer to [53] for their definitions.)

For the purpose of comparison, we first consider affinity propagation [15] for clustering (without any data preprocessing). The clustering technique is devised to detect representative exemplars (clusters) by taking the similarities between data pairs as input. When image data are represented by each of the 10 feature representations, the pairwise similarities are set to the negative distances measured by the corresponding distance function. The

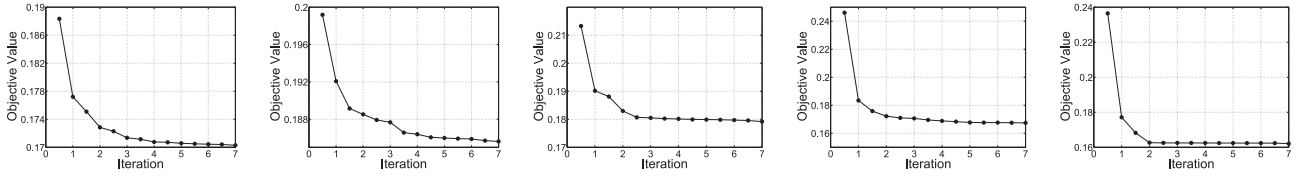


Fig. 6. The values of the objective function of MKL-LDA through the iterative optimization procedure when N_{train} is set as (a) 5, (b) 10, (c) 15, (d) 20, and (e) 25, respectively.

TABLE 3
Clustering Performances on the 20-Class Image Data Set

kernel(s)	preprocessing method	affinity propagation		k-means clustering	
		without data preprocessing	with data preprocessing	without data preprocessing	with data preprocessing
GB-Dist	Kernel LPP (KLPP)	0.580 / 50.7%	0.617 / 54.5%	–	0.634 / 56.3%
GB		0.531 / 46.2%	0.598 / 55.2%	–	0.612 / 54.7%
SIFT-Dist		0.638 / 59.8%	0.621 / 57.0%	–	0.630 / 54.0%
SIFT-SPM		0.590 / 55.8%	0.612 / 59.2%	–	0.630 / 54.7%
SS-Dist		0.527 / 45.3%	0.557 / 49.5%	–	0.565 / 50.2%
SS-SPM		0.573 / 55.5%	0.586 / 56.8%	–	0.603 / 57.7%
C2-SWP		0.400 / 32.8%	0.374 / 32.0%	0.383 / 31.5%	0.380 / 31.8%
C2-ML		0.494 / 44.2%	0.490 / 44.0%	0.525 / 47.0%	0.509 / 44.7%
PHOG		0.455 / 42.7%	0.490 / 46.3%	–	0.504 / 47.2%
GIST		0.515 / 49.2%	0.491 / 48.8%	0.494 / 44.7%	0.502 / 42.8%
-	KLPP-*Concatenate*	–	0.626 / 68.8%	–	0.667 / 62.3%
-	KLPP-*AvgKernel*	–	0.702 / 77.7%	–	0.708 / 62.5%
All	MKL-LPP	–	**0.737** / **78.3%**	–	**0.759** / **71.2%**

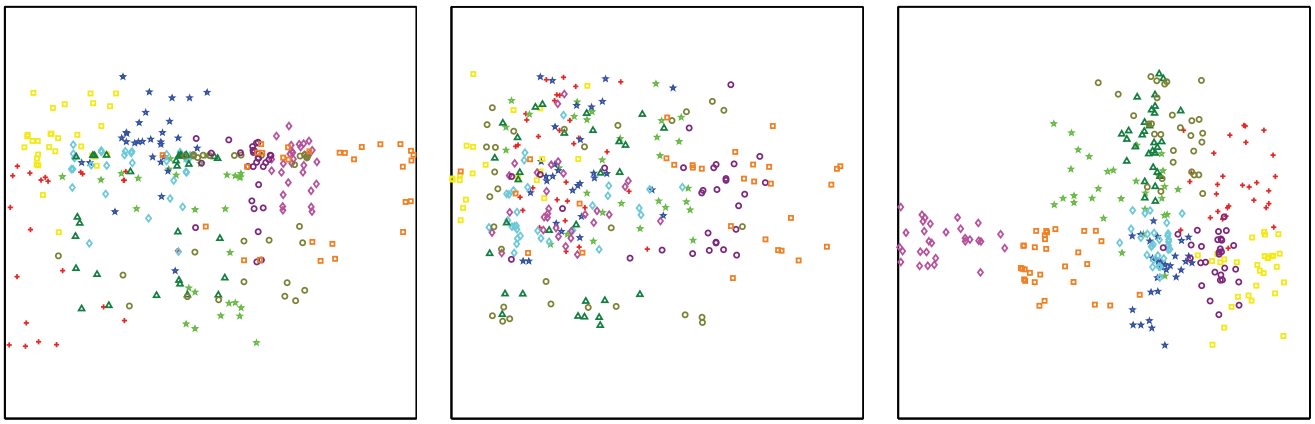
[NMI/ACC] percent.

clustering results evaluated based on NMI and ACC are reported in the third column of Table 3.

We then, respectively, adopt kernel LPP and MKL-LPP to preprocess the data. The main difference between the two is that kernel LPP learns a projection by taking one base kernel into account at a time, while MKL-LPP considers the 10 base kernels simultaneously. In the fourth column of Table 3, we show the clustering results of applying affinity propagation to the projected data by both schemes. It can be observed that with the advantage of exploring data characteristics from various aspects, MKL-LPP can achieve significant improvements in the clustering outcomes: NMI is increased from 0.621 to 0.737 and ACC is improved from 59.2 to 78.3 percent. Furthermore, owing to its better use of complementary image descriptors, MKL-LPP also outperforms KLPP-*Concatenate* and KLPP-*AvgKernel*.

The same experiments are repeated by replacing affinity propagation with k-means, and the results are given in the last two columns of Table 3. Note that if no additional preprocessing is performed, k-means is only applicable to data under the representations of C2-SWP, C2-ML, and GIST in that the others lead to non-euclidean spaces. Again, considerable performance gains with MKL-LPP can be concluded from the clustering results.

Besides demonstrating the usefulness of MKL-LPP with the quantitative results, it would be insightful if the projected data can be visually compared. However, for kernel LPP and MKL-LPP, directly embedding the data into a 2D space for visualization is not practical since both would not yield good clustering results in such a low-dimensional space. Instead, we first use kernel LPP and MKL-LPP to embed the data to low-dimensional spaces in which they, respectively, achieve their best clustering performance. Then, we apply *multidimensional scaling* (MDS) [11] to find the 2D projections. In Fig. 7, we show 2D visualizations of the projected data, respectively, obtained by kernel LPP with the base kernels GB-Dist and GIST, and by MKL-LPP with all the 10 base kernels. Each point in the figures represents a data sample, and its color indicates its class label. For the purpose of better illustration, only data from



(a) (b) (c)

Fig. 7. The 2D visualizations of the projected data. Each point represents a data sample, and its color indicates its class label. The projections are learned by (a) kernel LPP with base kernel GB-Dist (KLPP + GB-Dist), (b) kernel LPP with base kernel GIST (KLPP + GIST), and (c) MKL-LPP with all the 10 base kernels (MKL-LPP + All kernel).

Fig. 8. Four kinds of intraclass variations caused by (a) different lighting conditions, (b) in-plane rotations, (c) partial occlusions, and (d) out-of-plane rotations.

10 of the 20 classes (i.e., even-numbered classes) are plotted. Fig. 7 reveals that MKL-LPP can effectively utilize the data characteristics extracted by different descriptors and results in a more meaningful projection. That is, data of the same class tend to gather together, while data from different classes are kept apart. This property facilitates a clustering algorithm to identify representative clusters and achieve better performances.

6 Experimental Results: Semi-Supervised Learning for Face Recognition

Our last set of experiments focuses on evaluating the performance gains of applying MKL-DR to semi-supervised learning tasks. In solving such problems, the given data set is often partially labeled, and one is required to make use of both the class information of the labeled data and the intrinsic relationships of the unlabeled ones to accomplish the learning tasks. Specifically, we consider the face recognition problem to demonstrate the advantages of our approach, and exploit the property that the face images of an identity spread as a submanifold if they are sufficiently sampled to guide and regularize the optimization process in learning a more effective face recognition system.

6.1 Data Set

The CMU PIE database [44] is used in our experiments of face recognition. It is comprised of face images of 68 subjects. For a practical setting, we divide the 68 people into four equal-size disjoint groups, each of which contains face images from 17 subjects characterized by a certain kind of variations. (See Fig. 8 for an overview.) Specifically, for each subject in the first group, we consider only the images of the frontal pose (C27) taken in varying lighting conditions (those under the directory "`lights`"). For subjects in the second and third groups, the images with near frontal poses (C05, C07, C09, C27, and C29) under the directory "`expression`" are used. While each image from the second group is rotated by a randomly sampled angle within $[-45°, 45°]$, each from the third group is instead occluded by a nonface patch whose area is about 10 percent of the face region. Finally, for subjects in the fourth group, the images with out-of-plane rotations are selected under the directory "`expression`" and with the poses (C05, C11, C27, C29, and C37). All images are cropped and resized to 51×51 pixels.

Fig. 9. Images obtained by applying the delighting algorithm [22] to the five images in Fig. 8a. Clearly, variations caused by different lighting conditions are alleviated.

Performing face recognition over the resulting data set is challenging because the distances among data of the same class (identity) could be even larger than those among data of distinct classes if improper descriptors are used. On the other hand, adding the aforementioned variations to the data set is useful for emulating the practical situations, which are often caused, say, by imperfect face detectors or in uncontrolled environments.

6.2 Image Descriptors and Base Kernels

Again our objective is to select a set of visual features that can well capture subjects' characteristics as well as tolerate the large intraclass variations. Totally, we consider using four image descriptors and their respective distance function. Via (9), they result in four dissimilarity-based kernels described as follows:

- **RsL2**: Each sample is represented by its pixel intensities in raster scan order. Also, the euclidean (L^2) distance is used to correlate two images. This is a widely used representation for face images.
- **RsLTS**: The base kernel is similar to RsL2, except that the distance function is now based on the *least trimmed squares* (LTS) with 20 percent outliers allowed. It is designed to take account of the partial occlusions in a face image.
- **DeLight**: The underlying feature representation is obtained from the delighting algorithm [22], and the corresponding distance function is set as $1 - \cos\theta$, where θ is the angle between a pair of samples under the representation. Some delighting results are shown in Fig. 9. It can be seen that variations caused by different lighting conditions are significantly alleviated under the representation.
- **LBP**: As is illustrated in Fig. 10, we divide each image into $96 = 24 \times 4$ regions, and use a rotation-invariant

Fig. 10. Each image is divided into 96 regions. The distance between the two images is obtained when circularly shifting causes ψ' to be the new starting radial axis.

TABLE 4
Recognition Rates of Several Classifiers on the CMU PIE Data Set

method	kernel(s)	dataset (number of classes)				
		All (68)	Lighting (17)	Rotation (17)	Occlusion (17)	Profile (17)
Kernel SDA (KSDA)	RsL2	37.8 ± 2.0 %	56.1 ± 8.0 %	29.9 ± 3.3 %	25.5 ± 2.6 %	39.7 ± 6.4 %
	RsLTS	54.5 ± 2.0 %	46.8 ± 5.0 %	48.5 ± 9.7 %	68.9 ± 5.9 %	53.7 ± 6.6 %
	DeLight	44.7 ± 1.5 %	98.8 ± 2.3 %	15.9 ± 4.0 %	42.6 ± 5.3 %	21.3 ± 3.7 %
	LBP	62.0 ± 2.1 %	88.0 ± 6.3 %	52.2 ± 7.8 %	58.1 ± 8.3 %	49.8 ± 8.6 %
KSDA-*Voting*	-	65.7 ± 1.5 %	94.1 ± 4.0 %	54.9 ± 5.2 %	66.7 ± 3.7 %	47.2 ± 6.3 %
KSDA-*Concatenate*	-	68.1 ± 2.1 %	94.4 ± 6.2 %	47.8 ± 8.4 %	73.0 ± 5.7 %	57.1 ± 8.6 %
KSDA-*AvgKernel*	-	70.8 ± 2.4 %	93.6 ± 4.2 %	55.1 ± 5.7 %	79.4 ± 5.0 %	55.1 ± 8.9 %
KSDA-*SAMME*	-	67.2 ± 1.2 %	96.1 ± 3.3 %	52.9 ± 5.9 %	68.6 ± 5.0 %	51.1 ± 5.3 %
MKL-LDA	All	69.7 ± 1.5 %	97.5 ± 2.9 %	48.0 ± 7.4 %	72.3 ± 6.9 %	61.0 ± 7.1 %
MKL-LDE	All	68.1 ± 2.6 %	**99.8 ± 0.7 %**	45.6 ± 6.3 %	71.3 ± 7.3 %	55.6 ± 10.2 %
MKL-SDA	All	**78.8 ± 2.5 %**	97.8 ± 2.4 %	**65.4 ± 8.0 %**	**82.8 ± 3.3 %**	**69.1 ± 5.8 %**

[Mean $\pm$ std] *percent*.

local binary pattern (LBP) operator [36] (with operator setting $LBP^{riu2}_{8,1}$) to detect 10 distinct binary patterns. Thus, an image can be represented by a 960-dimensional vector, where each dimension records the number of occurrences that a specific pattern is detected in the corresponding region. To achieve rotation invariance, the distance between two such vectors, say, $\mathbf{x}_i$ and $\mathbf{x}_j$, is the minimal one among the 24 values computed from the distance function $1 - \mathrm{sum}(\min(\mathbf{x}_i, \mathbf{x}_j))/\mathrm{sum}(\max(\mathbf{x}_i, \mathbf{x}_j))$ by circularly shifting the starting radial axis for $\mathbf{x}_j$. Clearly, the base kernel is constructed to deal with variations resulting from rotations.

6.3 Dimensionality Reduction Method

We adopt SDA [6] as the semi-supervised DR technique to be generalized by MKL-DR. SDA carries out discriminant learning over labeled data while preserving the geometric structure of unlabeled data. Analogously to LDA and LDE in Section 4.3, SDA can be specified by two affinity matrices $W = [w_{ij}]$ and $W' = [w'_{ij}]$, when a partially labeled data set, $\Omega = \{\mathbf{x}_p, y_p\}_{p=1}^{N_\ell} \cup \{\mathbf{x}_q\}_{q=N_\ell+1}^{N_\ell+N_u}$, is available

$$w_{ij} = \begin{cases} 1/n_{y_i} + \alpha \cdot s_{ij}, & \text{if } y_i = y_j, \\ \alpha \cdot s_{ij}, & \text{otherwise}, \end{cases} \tag{43}$$

$$w'_{ij} = \begin{cases} 1/N_\ell, & \text{if both } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are labeled}, \\ 0, & \text{otherwise}, \end{cases} \tag{44}$$

where

$$s_{ij} = \begin{cases} 1, & \text{if } i \in \mathcal{N}_k(j) \vee j \in \mathcal{N}_k(i), \\ 0, & \text{otherwise}, \end{cases} \tag{45}$$

and α is a positive parameter to adjust the relative importance between the label information and the neighborhood relationships.

6.4 Quantitative Results

In the experiments, we randomly select 12 images from each subject. Three of them serve as the labeled training data, while the other nine as the unlabeled ones. In total, we have 816 (i.e., 12×68) images for training (204 of them are labeled), and the remainder for testing. Since the numbers of images for subjects in different groups may not be the same, an accuracy rate is first computed for each subject. And the reported recognition rate is the average of them. The overall procedure is repeated eight times to reduce the effect of sampling.

We report the mean recognition rates and the standard deviation in the third column of Table 4. It can be observed that MKL-SDA achieves a significant performance gain of 16.8 percent ($=78.8\% - 62.0\%$) over the best recognition rate by the four kernel SDA (or KSDA for short) classifiers. It also outperforms KSDA-*Voting*, KSDA-*Concatenate*, KSDA-*AvgKernel*, and KSDA-*SAMME*, and improves the recognition rates from 65.7 to 70.8 percent to 78.8 percent.

To evaluate the effect of using unlabeled training data in SDA, we compare MKL-SDA with MKL-LDA and MKL-LDE. The main difference among them is that MKL-SDA considers both labeled and unlabeled training data, while MKL-LDA and MKL-LDE use only the labeled ones. The quantitative results in Table 4 show that MKL-SDA can boost the recognition rate about 10 percent by making use of the additional information from the unlabeled training data.

We also provide the recognition rates with respect to each of the four groups in the last four columns of Table 4. (We name each group according to the type of its intraclass variation.) Note that each of such recognition rates is computed by considering only the data in a particular group. And no new classifiers are trained. As expected, the four base kernels generally result in classifiers that produce good performances in dealing with some specific kinds of intraclass variations. For example, the base kernel DeLight achieves a near perfect result for subjects in the Lighting group, and RsLTS yields satisfactory results in the Occlusion group. However, none of them is good enough for dealing with the whole data set. On the other hand, MKL-DR can effectively combine the four base kernels to complement them, and leads to a remarkable increase in accuracy.

Finally, we discuss the learned combination weights over the four base kernels, i.e., $\boldsymbol{\beta}$ in Algorithm 1. In Fig. 11, we plot the averages and the standard deviations of the learned $\boldsymbol{\beta}$ by MKL-SDA. Observe first that the weights are not directly in proportion to the individual performances of the base kernels. This is mostly due to that the scales of the four base kernels are different. Further, the degrees of information complement and redundancy among these base kernels should also be taken into account since they are considered

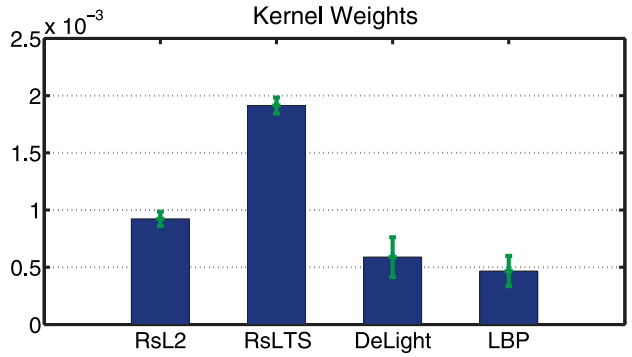


Fig. 11. The learned kernel weights by MKL-SDA.

jointly in learning the weights. The other useful property readily inferred from Fig. 11 is that the standard deviations in the combination weights are uniformly small, and it highlights the desirable stability of MKL-DR.

7 Conclusions

The proposed MKL-DR introduces a new paradigm to fortify a broad scope of existing dimensionality reduction techniques. Its main advantage lies in the ability of learning a unified space of low dimension for data in multiple feature representations. Such a flexibility is important in tackling complicated vision problems, and allows one to explore more prior knowledge for effectively analyzing a given data set, including choosing a proper set of visual features to better characterize the data and adopting a graph-based DR method to appropriately model the relationship among the data points. Throughout this work, MKL-DR has been comprehensively evaluated in three important computer vision applications, including supervised object recognition, unsupervised image clustering, and semi-supervised face recognition. The promising experimental results further consolidate the usefulness of our approach.

Also, as we have demonstrated, MKL-DR can extend the multiple kernel learning framework to address not only the supervised learning problems but also the unsupervised and semi-supervised ones. This aspect of generalization introduces a new frontier in applying MKL to solving vision and learning applications.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments. This work is supported in part by grants 95-2221-E-001-031-MY3 and 97-2221-E-001-019-MY3.

References

[1] F. Bach, G. Lanckriet, and M. Jordan, "Multiple Kernel Learning, Conic Duality, and the SMO Algorithm," *Proc. Int'l Conf. Machine Learning,* 2004.

[2] A. Berg, T. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low Distortion Correspondences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 26-33, 2005.

[3] A. Berg and J. Malik, "Geometric Blur for Template Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 607-614, 2001.

[4] O. Boiman, E. Shechtman, and M. Irani, "In Defense of Nearest-Neighbor Based Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[5] A. Bosch, A. Zisserman, and X. Muñoz, "Image Classification Using Random Forests and Ferns," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[6] D. Cai, X. He, and J. Han, "Semi-Supervised Discriminant Analysis," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[7] J. Carreira and C. Sminchisescu, "Constrained Parametric Min-Cuts for Automatic Object Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[8] C.-P. Chen and C.-S. Chen, "Lighting Normalization with Generic Intrinsic Illumination Subspace for Face Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1089-1096, 2005.

[9] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local Discriminant Embedding and Its Variants," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 846-853, 2005.

[10] M. Christoudias, R. Urtasun, and T. Darrell, "Bayesian Localized Multiple Kernel Learning," technical report, Electical Eng. and Computer Science Dept., Univ. of California, Berkeley, 2009.

[11] T. Cox and M. Cox, *Multidimentional Scaling.* Chapman & Hall, 1994.

[12] D. Dueck and B. Frey, "Non-Metric Affinity Propagation for Unsupervised Image Categorization," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[13] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, The PASCAL Visual Object Classes Challenge (VOC2007) Results, 2007.

[14] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Proc. IEEE Computer Vision and Pattern Recognition Workshop Generative-Model Based Vision,* 2004.

[15] B. Frey and D. Dueck, "Clustering by Passing Messages between Data Points," *Science,* vol. 315, pp. 972-976, 2007.

[16] A. Frome, Y. Singer, and J. Malik, "Image Retrieval and Classification Using Local Distance Functions," *Advances in Neural Information Processing Systems,* pp. 417-424, MIT Press, 2006.

[17] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[18] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet, "Multi-Class Object Localization by Combining Local Contextual Interactions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[19] P. Gehler and S. Nowozin, "On Feature Combination for Multiclass Object Classification," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[20] M. Gönen and E. Alpaydin, "Localized Multiple Kernel Learning," *Proc. Int'l Conf. Machine Learning,* pp. 352-359, 2008.

[21] K. Grauman and T. Darrell, "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1458-1465, 2005.

[22] R. Gross and V. Brajovic, "An Image Preprocessing Algorithm for Illumination Invariant Face Recognition," *Proc. Int'l Conf. Audio- and Video-Based Biometric Person Authentication,* pp. 10-18, 2003.

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

[24] A. Holub, M. Welling, and P. Perona, "Combining Generative Models and Fisher Kernels for Object Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 136-143, 2005.

[25] I. Joliffe, *Principal Component Analysis.* Springer-Verlag, 1986.

[26] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian Processes for Object Categorization," *Int'l J. Computer Vision,* vol. 88, no. 2, pp. 169-188, 2010.

[27] S.-J. Kim, A. Magnani, and S. Boyd, "Optimal Kernel Selection in Kernel Fisher Discriminant Analysis," *Proc. Int'l Conf. Machine Learning,* pp. 465-472, 2006.

[28] A. Kumar and C. Sminchisescu, "Support Kernel Machines for Object Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[29] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, "Learning the Kernel Matrix with Semidefinite Programming," *J. Machine Learning Research,* vol. 5, pp. 27-72, 2004.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2169-2178, 2006.

[31] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Local Ensemble Kernel Learning for Object Category Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.
[32] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Dimensionality Reduction for Data in Multiple Feature Representations," *Advances in Neural Information Processing Systems,* pp. 961-968, MIT Press, 2008.
[33] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.
[34] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher Discriminant Analysis with Kernels," *Proc. Workshop Neural Networks for Signal Processing,* pp. 41-48, 1999.
[35] J. Mutch and D. Lowe, "Multiclass Object Recognition with Sparse, Localized Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 11-18, 2006.
[36] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 971-987, July 2002.
[37] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *Int'l J. Computer Vision,* vol. 42, no. 3, pp. 145-175, 2001.
[38] E. Pekalska, P. Paclik, and R. Duin, "A Generalized Kernel Approach to Dissimilarity-Based Classification," *J. Machine Learning Research,* vol. 2, no. 2, pp. 175-211, 2002.
[39] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "More Efficiency in Multiple Kernel Learning," *Proc. Int'l Conf. Machine Learning,* pp. 775-782, 2007.
[40] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 2000.
[41] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, no. 5, pp. 1299-1319, 1998.
[42] T. Serre, L. Wolf, and T. Poggio, "Object Recognition with Features Inspired by Visual Cortex," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 994-1000, 2005.
[43] E. Shechtman and M. Irani, "Matching Local Self-Similarities across Images and Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.
[44] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1615-1618, Dec. 2003.
[45] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large Scale Multiple Kernel Learning," *J. Machine Learning Research,* vol. 7, pp. 1531-1565, 2006.
[46] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research,* vol. 3, pp. 583-617, 2002.
[47] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, pp. 2319-2323, 2000.
[48] S. Todorovic and N. Ahuja, "Learning Subcategory Relevances for Category Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.
[49] L. Vandenberghe and S. Boyd, "Semidefinite Programming," *SIAM Rev.,* vol. 38, pp. 49-95, 1996.
[50] M. Varma and D. Ray, "Learning the Discriminative Power-Invariance Trade-Off," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.
[51] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple Kernels for Object Detection," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.
[52] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace Ratio versus Ratio Trace for Dimensionality Reduction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.
[53] M. Wu and B. Schölkopf, "A Local Learning Approach for Clustering," *Advances in Neural Information Processing Systems,* pp. 1529-1536, MIT Press, 2006.
[54] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 1, pp. 40-51, Jan. 2007.
[55] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group-Sensitive Multiple Kernel Learning for Object Categorization," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.
[56] J. Ye, R. Janardan, and Q. Li, "GPCA: An Efficient Dimension Reduction Scheme for Image Compression and Retrieval," *Proc. ACM SIGKDD,* pp. 354-363, 2004.
[57] J. Ye, R. Janardan, and Q. Li, "Two-Dimensional Linear Discriminant Analysis," *Advances in Neural Information Processing Systems,* MIT Press, 2004.
[58] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2126-2136, 2006.
[59] J. Zhu, S. Rosset, H. Zou, and T. Hastie, "Multi-Class Adaboost," technical report, Dept. of Statistics, Univ. of Michigan, 2005.

Yen-Yu Lin received the BS degree in information management in 2001 and the MS degree in computer science and information engineering in 2003 from National Taiwan University, where he is currently a PhD candidate in the same graduate program. Since 2004, he has been working as a full-time research assistant at the Institute of Information Science, Academia Sinica, Taiwan. His research interests include computer vision, pattern recognition, and machine learning.

Tyng-Luh Liu received the bachelor's degree in applied mathematics from the National Chengchi University of Taiwan in 1986 and the PhD degree in computer science from New York University in 1997. He is a research fellow at the Institute of Information Science, Academia Sinica, Taiwan. He received the Junior Research Investigators Award, Academia Sinica, in 2006. His research interests include computer vision, pattern recognition, and machine learning. He is a member of the IEEE.

Chiou-Shann Fuh received the BS degree in computer science and information engineering from National Taiwan University, Taipei, in 1983, the MS degree in computer science from Pennsylvania State University, University Park, in 1987, and the PhD degree in computer science from Harvard University, Cambridge, Massachusetts, in 1992. He was with AT&T Bell Laboratories, where he was engaged in performance monitoring of switching networks from 1992 to 1993. He was an associate professor in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, from 1993 to 2000, and was then promoted to a full professor. His current research interests include digital image processing, computer vision, pattern recognition, mathematical morphology, and their applications to defect inspection, industrial automation, digital still camera, digital video camcorder, and camera module such as color interpolation, auto exposure, auto focus, auto white balance, color calibration, and color management. He is a member of the IEEE.