Semisupervised Online Multikernel Similarity Learning for Image Retrieval

Jianqing Liang, Qinghua Hu, *Senior Member, IEEE*, Wenwu Wang, and Yahong Han

Abstract—Metric learning plays a fundamental role in the fields of multimedia retrieval and pattern recognition. Recently, an online multikernel similarity (OMKS) learning method has been presented for content-based image retrieval (CBIR), which was shown to be promising for capturing the intrinsic nonlinear relations within multimodal features from large-scale data. However, the similarity function in this method is learned only from labeled images. In this paper, we present a new framework to exploit unlabeled images and develop a semisupervised OMKS algorithm. The proposed method is a multistage algorithm consisting of feature selection, selective ensemble learning, active sample selection, and triplet generation. The novel aspects of our work are the introduction of classification confidence to evaluate the labeling process and select the reliably labeled images to train the metric function, and a method for reliable triplet generation, where a new criterion for sample selection is used to improve the accuracy of label prediction for unlabeled images. Our proposed method offers advantages in challenging scenarios, in particular, for a small set of labeled images with high-dimensional features. Experimental results demonstrate the effectiveness of the proposed method as compared with several baseline methods.

Index Terms—Image retrieval, metric learning, multikernel learning, online multikernel similarity (OMKS), similarity learning, semisupervised, semisupervised online multikernel similarity (SSOMKS).

I. INTRODUCTION

WITH the rapid growth of multimedia data such as images and videos, measuring the similarity between visual objects becomes an increasingly important task in a variety of applications including classification, clustering and retrieval [1]–[3]. Conventionally, this can be achieved by using pre-defined functions, such as the Euclidean distance and cosine similarity. With these functions, however, the underlying distribution of the data is often implicitly assumed. As a result, the complex intrinsic structures within the data may not be well captured by these functions.

To address this problem, an increasing amount of effort has been made to learn an appropriate metric directly from the data, for applications such as content-based image retrieval (CBIR), which is our focus here. In the pioneering work by Xing *et al.* [4], metric learning is formulated as a convex optimization problem with a set of similarity and dissimilarity constraints, where a global Mahalanobis distance is learned by keeping similar pairs of objects close to each other while dissimilar pairs apart from each other [5]. This earlier work has inspired the development of a number of methods for learning global linear metrics, such as the information-theoretic method [6], [7], nearest neighbor classification method [8], Laplace regularized metric learning (LRML) [9], and more recently, the geometric mean metric learning (GMML) method [10].

These global metric learning techniques, however, are often limited for large-scale problems due to their high computational complexity. They may also suffer from the issue of the so-called curse of dimensionality [11]. To overcome these limitations, a number of algorithms have been presented for learning local metrics [12]–[16], which are deemed to be more flexible for capturing the variations across multiple feature spaces, and offering better performance, as compared with global metrics. However, the local metrics tend to be prone to the problem of overfitting [5].

The aforementioned algorithms aim to learn linear metrics which may have limitations in characterising the relations between the different modalities in multi-modal data, since they often have non-linear relations, and are in different spaces and dimensions. To address these issues, multiple kernel techniques have been introduced [17]–[20], by mapping the images to a high-dimensional feature space with a nonlinear kernel matrix. In [17], an optimal ensemble of kernel transformations is learned for integrating features of multiple modalities into a unified space. However, it is computationally expensive, and consequently not applicable to high-dimensional and large-scale datasets. In [18], a multi-modal distance metric learning framework is proposed by projecting data from different modalities into a latent feature space based on the multi-wing harmonium model. In [19], a weighted kernel embedding technique is presented for metric learning, which is shown to be flexible in combining multiple features. Using multiple kernel techniques, the complementary nature of different features extracted from an image can be better exploited. For this reason, multi-kernel learning techniques are also considered in our work.

J. Liang, Q. Hu, and Y. Han are with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: liangjianqing@tju.edu.cn; huqinghua@tju.edu.cn; yahong@tju.edu.cn).

W. Wang is with the Center for Vision Speech and Signal Processing, Department of Electronic Engineering, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: w.wang@surrey.ac.uk).

In applications with large-scale data, however, the algorithms discussed above are often limited in their scalability. The computational complexity and memory requirement of these algorithms may increase significantly when dealing with large-scale data [5]. To address this challenge, online techniques have been proposed in e.g. [21]–[23]. In [23], an online multiple kernel similarity learning (OMKS) algorithm is presented, where a flexible nonlinear proximity function with multiple kernels is learned in a supervised manner and applied to visual search. In [21], an online multimodal deep similarity learning (OMDSL) framework is proposed to improve multimedia similarity search by integrating multiple deep networks with a scalable online scheme. In [22], an online multi-modal distance metric learning (OMDML) scheme is proposed, where the optimal metrics are learned in individual modality space and the weights for combining different modalities are obtained with a joint formulation. These algorithms rely overwhelmingly on the availability of labeled data in their training. In practice, however, labelling data by human is time consuming and costly. In addition, the labels provided by different labellers are not always consistent and could be noisy. Therefore, it is highly desirable if the large-scale unlabelled data could be directly used in metric learning.

The use of unlabelled data in metric learning has been considered in previous work e.g. [9], [24], [25]. However, these methods were proposed for learning global Mahalanobis metrics, but not for local metrics. In this paper, inspired by the OMKS algorithm in [23], we propose a novel multi-stage semi-supervised online multi-kernel similarity (SSOMKS) learning framework for using the unlabelled data in metric learning. More specifically, we present a new method for triplet generation to allow the incorporation of the unlabelled data in the OMKS algorithm. An important challenge in the use of unlabelled data comes from the risk associated with the unreliability and noise in the training samples. To counter this problem, a new active sample selection method based on the concept of margin is proposed for measuring the classification confidence. This leads to a new method for reliable triplet generation where the labeling process is evaluated in order to select the reliably labeled images for learning the metric function. To our knowledge, such an idea has not been exploited in metric learning.

The remainder of this paper is organized as follows. Section II briefly summarises the baseline OMKS algorithm in [23]. In Section III, we introduce our proposed SSOMKS learning framework which is a multi-stage method including feature selection, selective ensemble learning, active sample selection, and triplet generation. Section IV presents experimental results on both qualitative and quantitative analysis, including the evaluation of each stage of the proposed method and its comparison with several baseline methods. We conclude the paper with an outlook for future work in Section V.

II. THE OMKS LEARNING METHOD

In this section, we give a brief introduction to the OMKS algorithm presented in [23].

Suppose there is a kernel $\kappa(\cdot, \cdot)$ and the corresponding Hilbert space $\mathcal{H}$, and consider a linear operator $L: \mathcal{H} \mapsto \mathcal{H}$ that maps a function $f \in \mathcal{H}$ to another one $L[f] \in \mathcal{H}$. Assume there is a collection of m kernel functions $\mathcal{K} = \{\kappa_i : \chi \times \chi \to \mathbb{R}, i = 1, \ldots, m\}$. A similarity function for visual search is defined as

$$f(q, p) = \sum_{i=1}^{m} \theta_i S_i(q, p) = \sum_{i=1}^{m} \theta_i \langle \kappa_i(q, \cdot), L_i[\kappa_i(p, \cdot)] \rangle_{\mathcal{H}_{\kappa_i}} \quad (1)$$

where $q \in \chi$ is a query image, and $p \in \chi$ is an image in the pooling set to be retrieved. $S_i(q, p) = \langle \kappa_i(q, \cdot), L_i[\kappa_i(p, \cdot)] \rangle_{\mathcal{H}_{\kappa_i}}$ is the similarity function based on the linear operator L_i. The goal is to learn the weights $\{\theta_i\}_{i=1}^{m}$ and the linear operators $\{L_i\}_{i=1}^{m}$ simultaneously.

Given a set of T triplets $\{(p_t, p_t^+, p_t^-)\}_{t=1}^{T}$ where p_t should be more similar to p_t^+ than to p_t^-, the objective function that needs to be optimised is given as follows:

$$\min_{\theta \in \triangle} \min_{\{L_i\}_{i=1}^{m}} \tfrac{1}{2} \sum_{i=1}^{m} \theta_i \|L_i\|_{HS}^2 + C \sum_{t=1}^{T} \ell(f(p_t, p_t^+) - f(p_t, p_t^-)) \quad (2)$$

where $\| \cdot \|_{HS}$ is the Hilbert Schmidt norm of the linear operator, $C \geq 0$ is the loss parameter, $\ell(z)$ is the hinge loss and $\triangle$ is defined as

$$\triangle = \{\theta \in \mathbb{R}_+^m \mid \theta^T e_m = 1\}. \quad (3)$$

To solve the problem (2), online learning techniques are introduced. In particular, for kernel κ_i, the corresponding weight θ_i and linear operator L_i are updated in T iterations. That is, when the tth triplet (p_t, p_t^+, p_t^-) arrives, the weight $\theta_i(t-1)$ and linear operator $L_{t-1,i}$ in kernel κ_i are updated to obtain $\theta_i(t)$ and $L_{t,i}$, respectively.

Starting with $L_{0,i} = I$, $L_{t,i}$ for the tth triplet is updated as

$$L_{t,i} = L_{t-1,i} + \tau_{t,i} Z_t \quad (4)$$

where $h \in \mathcal{H}$, $Z_t[h](\cdot) = \kappa(p_t, \cdot)(h(p_t^+) - h(p_t^-)) \in \mathcal{L}$ ($\mathcal{L} = \{L : \mathcal{H} \mapsto \mathcal{H}, L \text{ is a linear operator}\}$ is the space including linear operators in $\mathcal{H}$) and $\tau_{t,i}$ is computed as

$$\tau_{t,i} = \min \left\{ C, \frac{\max\{0, 1 - S_{L_{t-1,i}}(p_t, p_t^+) + S_{L_{t-1,i}}(p_t, p_t^-)\}}{\kappa(p_t, p_t)(\kappa(p_t^+, p_t^+) - 2\kappa(p_t^+, p_t^-) + \kappa(p_t^-, p_t^-))} \right\}. \quad (5)$$

Then, the weight of kernel κ_i is updated as

$$\theta_i(t) = \theta_i(t-1) \beta^{z_i(t)} \quad (6)$$

where $\beta \in (0, 1)$ is a discounting parameter which is used to penalize the kernel that makes incorrect predictions in each iteration, and $z_i(t)$ equals to 1 when $S_{L_{t-1,i}}(p_t, p_t^+) - S_{L_{t-1,i}}(p_t, p_t^-) \leq 0$, and 0 otherwise.

The OMKS algorithm is a supervised algorithm trained with labelled data. The triplet generation does not consider the use of unlabeled data. To address this issue, we propose a new semi-supervised multi-stage learning framework, by extending the OMKS algorithm to the scenario where only a small amount of training data is labelled while the majority of the data are unlabeled, as discussed next.
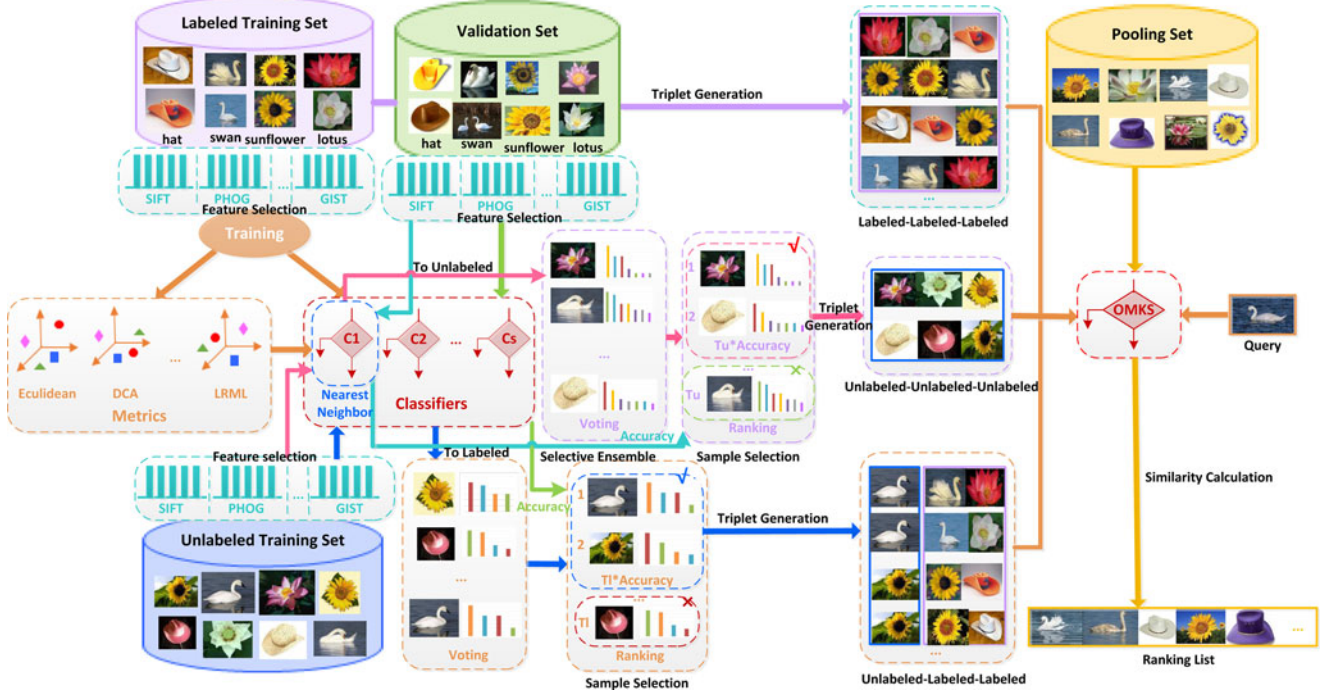
Fig. 1. Flow chart of the semisupervised online multiple kernel similarity framework for image retrieval. For each image in the dataset, we extract nine types of features (e.g., SIFT, PHOG, etc.) and then select several dimensions in each modality. Then, we learn metrics (e.g., DCA, LRML, etc.) as well as classifiers (e.g., random forest, subspace, etc.) for each modality with the labeled training set. Specifically, we apply the learned metrics for constructing the nearest neighbor classifier. To select appropriate classifiers, selective ensemble learning is performed on the validation set. Unlabeled training set is viewed as test data. By searching the nearest and farthest class of these images with the selected classifiers, we obtain unlabeled-labeled-labeled triplets. Unlabeled-unlabeled-unlabeled triplets can be generated by finding the nearest and farthest samples, while labeled-labeled-labeled triplets are produced by supervision information of labeled training set. For the first two types of triplets, we also make the first attempt to perform active sample selection with margin. Please refer to Section IV for details.

III. PROPOSED SEMISUPERVISED OMKS LEARNING METHOD

Our new framework of SSOMKS is a multi-stage method consisting of feature selection, selective ensemble learning, active sample selection, and triplet generation. The key contribution in this framework is a new method for generating the triplets, as well as a new approach for controlling the potential risk in using unlabelled data with active sample selection based on the concept of margin.

The diagram of the proposed method is shown in Fig. 1. First, feature selection is performed to obtain discriminative feature space. Then, ensemble learning is introduced to train the classifiers for each type of features, and the classifiers that offer better classification performance are selected. Third, an active sample selection method is proposed to ensure that the samples with correctly predicted labels are used. Finally, the triplets with these selected samples are generated to perform metric learning for visual search. The details of each stage are discussed below.

A. Feature Selection

High-dimensional multiple features extracted from images may contain redundant information. Feature selection is helpful for choosing the discriminative dimensions in the feature space. Here, we apply the Multi-Cluster Feature Selection (MCFS) [26] method, as it is computationally efficient and also independent of the choice of classifiers.

Given an image training set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_i \in \mathbb{R}^D$ of N images with D dimensions in K clusters. Suppose we want to select d dimensions and the number of nearest neighbors is set as p. For each image $\boldsymbol{x}_i$, we construct a p nearest neighbor graph by finding its p nearest neighbors and form an edge between $\boldsymbol{x}_i$ and its neighbors. We define the weight matrix $\boldsymbol{W}$ on the graph and a diagonal matrix $\boldsymbol{D}$ based on $\boldsymbol{W}$, $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. The graph Lapalcian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$. Solve the generalized eigen-problem [27]

$$\boldsymbol{L}\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}. \tag{7}$$

Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K]$ be the top K eigenvectors corresponding to the smallest eigenvalues $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]$. For each cluster, we solve the equivalent formulation of LASSO using Least Angle Regression (LARs) algorithm [28] by specifying the cardinality as d

$$\min_{\boldsymbol{a}_k} \|\boldsymbol{y}_k - \boldsymbol{X}^T \boldsymbol{a}_k\|^2$$
$$\text{s.t. } \|\boldsymbol{a}_k\|_1 = d. \tag{8}$$

Then we get K sparse coefficient vectors $\{\boldsymbol{a}_k\}_{k=1}^K \in \mathbb{R}^D$. The MCFS score for each feature j is computed as

$$MCFS(j) = \max_k |\boldsymbol{a}_{k,j}|. \tag{9}$$

We obtain the top d features according to the ranking.

B. Selective Ensemble Learning

Following feature selection, classification is performed to predict the labels of unlabelled samples. Here, ensemble learning is employed due to its advantage over a single classifier in its generalization ability to unseen data [29]. The performance of ensemble learning algorithms may vary. It was shown that ensembling many of the available learners can be better than ensembling all of them [30]. Therefore, selective ensemble learning is introduced to remove the under-performed learners. Here, we adopt the Margin based Pruning (MP) [31] algorithm to select proper classifiers. An advantage with the MP algorithm is that the distribution of the sample intervals can be further optimised during the process of ensemble selection. The MP procedure is discussed as follows.

Suppose $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is the training image set, $h_1, \ldots, h_L$ are the base classifiers, y_i is the true class label of $\boldsymbol{x}_i$, and $\hat{y}_{ij}$ $(j = 1, 2, \ldots, L)$ is the classification decision of $\boldsymbol{x}_i$ estimated by the classifier h_j. The margin of $\boldsymbol{x}_i$ is defined as

$$m(\boldsymbol{x}_i) = \sum_{j=1}^{L} w_j \Lambda_{ij} \qquad (10)$$

where w_j is the weight of h_j

$$\Lambda_{ij} = \begin{cases} 1, & \text{if } y_i = \hat{y}_{ij} \\ -1, & \text{if } y_i \neq \hat{y}_{ij}. \end{cases}$$

For $\boldsymbol{x}_i \in \boldsymbol{X}$, its classification loss is defined as

$$l(\boldsymbol{x}_i) = [1 - m(\boldsymbol{x}_i)]^2. \qquad (11)$$

The loss of classification is computed as

$$l(\boldsymbol{X}) = \sum_{i=1}^{N} l(\boldsymbol{x}_i) = \|\boldsymbol{u} - \boldsymbol{D}\boldsymbol{w}\|_2^2 \qquad (12)$$

where $\boldsymbol{u} = [1, \cdots, 1]_{N \times 1}^T$, $\boldsymbol{w} = [w_1, \cdots, w_L]_{L \times 1}^T$, $\boldsymbol{D} = \{\Lambda_{ij}\}_{N \times L}$.

The L2-norm regularization is added to the loss function [32]

$$F_{\boldsymbol{w}} = \|\boldsymbol{u} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2. \qquad (13)$$

The weights w_j $(j = 1, 2, \ldots, L)$ can be obtained by minimizing $F_{\boldsymbol{w}}$ using open software packages such as [33]. Then, the classifiers h_{sj} $(sj = 1, 2, \ldots, L)$ are ranked in terms of the descending order of the weights w_j $(j = 1, 2, \ldots, L)$. After this, we compute the average precision ψ_j with $\{h_{s1}, h_{s2}, \ldots, h_{sj}\}$. Finally, $\{h_{s1}, h_{s2}, \ldots, h_{sB}\}$ are the selected classifiers with $B = \max_{j \in \{1, \ldots, L\}} \psi_j$.

C. Sample Selection With Classification Confidence

For unlabelled data, the labels predicted by voting in the above section may not be reliable. As a result, the triplet could be wrongly generated, which can have negative impact on both the computational efficiency and learning performance of metric learning. To address this problem, we propose a new technique to select samples, based on the concept of margin, which has been previously used to measure the confidence of classification. If a trained model gives a large margin, it will have a higher degree of confidence and reliability. Inspired by the work in [34]–[37], we introduce the concept of classification confidence to sample
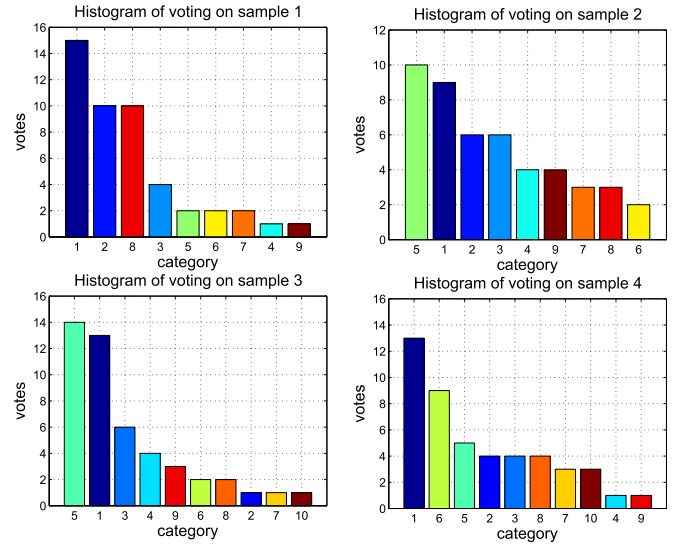

Fig. 2. Voting results of samples in Class 1. Each classifier outputs a label for each sample. The final predicted class is the one which gets the most votes. Samples 1 and 4 fall into Class 1, while samples 2 and 3 fall into Class 5.

selection. Our method is based on three hypotheses. First, each selected classifier has considerable classification ability, which means it is better than random guess. Second, the accuracy is positively related to the votes of the largest class. Third, the accuracy is positively related to the margin between the first and the second largest class.

Assume the L classifiers are independent, where $L = 2k + 1$ is odd. Let X_i be a variable indicating whether the classification by the ith classifier is correct or not. If the prediction accuracy of each classifier is p, then we have $X_i \sim \text{Bernouli}(p)$, and the number of correct classifications with the ensemble majority voting method is $Y = \sum_{i=1}^{L} X_i \sim \text{binomial}(L, p)$ [38]. The majority vote accuracy is

$$P_{\text{majority}(L)} = \sum_{i=k+1}^{L} \binom{L}{i} p^i (1-p)^{L-i}. \qquad (14)$$

It has been shown that the sequence $\{P_{\text{majority}(2k+1)}\}$ strictly increase when $p > 0.5$ [39]. In addition, $\lim_{k \to \infty} P_{\text{majority}(2k+1)} = 1$, and the prediction accuracy of the ensemble voting method converges to 1 when $p > 0.5$. As the probability for the largest number of votes may be smaller than half, (14) is the lower bound of the actual probability.

Suppose there are Nu unlabeled training images, B classifiers and K classes. We introduce a parameter c to balance between the maximum and the margin, and define the criterion for selection as $cmax + (1-c)margin$. We denote the voting accuracy of validation images with these B classifiers as $Accv$. After ranking the Nu images in a descending order of $cmax + (1-c)margin$, we select the top $Nu * Accv$ unlabeled images to generate triplets.

Fig. 2 gives an example of the voting results for four unlabeled samples from Class 1. We observe that only sample 1 and 4 are correctly classified. Two incorrectly labeled samples will be introduced with all classifiers. The accuracy on validation set

is 37.5%. Thus we may select 2 out of 4 samples with relevant strategies. If we adopt the max criterion, then sample 1 and 3 are chosen, introducing one mistake. If we use the $margin$ or $max + margin$ criterion, then sample 1 and 4 are selected without a mistake. This illustrates that our sample selection strategy can improve classification accuracy.

D. Triplet Generation

To exploit a certain unlabeled image x_i for metric learning, it is necessary to find its nearest neighbor x_j and farthest image x_k to generate the triplet (x_i, x_j, x_k).

Given an unlabeled sample x_i, the triplets can be divided into two types, i.e., unlabeled-labeled-labeled and unlabeled-unlabeled-unlabeled ones according to whether x_j and x_k are labeled or not. For the former, we first label x_i with the selected base classifiers. The farthest class is the one to which the farthest sample found with learned metrics belongs. Then, x_js can be the labeled training samples that belong to the same class with x_i, whereas x_ks are the labeled training samples that belong to its farthest class. For the latter, we are required to find the nearest and farthest samples to x_i in unlabeled training set. Without supervision information, we consider exploiting the learned metrics. Therefore, x_js are the nearest unlabeled samples while x_k s are the farthest unlabeled ones obtained by the metrics.

E. Summary of the Proposed SSOMKS Learning Framework

The implementation steps of the proposed SSOMKS method are summarised in Algorithm 1. It can be seen that the proposed SSOMKS differs from the OMKS algorithm in the process on how the triplet is generated. In the proposed method both the labeled and unlabeled images are used to learn a metric, while in the baseline OMKS method, only the labeled images are considered.

In constrast to [23], where a theoretical analysis is presented for the OMKS method, we do not yet have a theoretical proof for the convergence property of the proposed SSOMKS method. However, a simulation study is provided in Section IV-A for the analysis of its convergence.

IV. EXPERIMENTS

In this section, we evaluate the performance of SSOMKS and compare it with several baseline methods. We first introduce the experimental setting and then show the results as well as the analysis to these results.

A. Experimental Setting

1) Datasets and Experiment: We conduct the experiments on image datasets including Corel [40], ImageCLEF,[1] Indoor,[2] Caltech256,[3] Flickr[4] and Oxford Buildings.[5] We pick 10, 20 and

[1][Online]. Available: http://imageclef.org/
[2][Online]. Available: http://web.mit.edu/torralba/www/indoor.html
[3][Online]. Available: http://www.vision.caltech.edu/Image_DataSets/Caltech256/
[4][Online]. Available: http://press.liacs.nl/mirflickr/mirdownload.html
[5][Online]. Available: http://www.robots.ox.ac.uk/vgg/data/oxbuildings/index.html

---

Algorithm 1: The SSOMKS algorithm

Input:

    Labeled training set with M modalities
    $D^l = \{D_j^l\}_{j=1}^M$;
    Unlabeled training set $D^u = \{D_j^u\}_{j=1}^M$;
    Validation set $D^v = \{D_j^v\}_{j=1}^M$;
    Trade-off parameter c;

Output:

    $f(q, p)$;

1:   **for** $j = 1$ to M **do**
2:       Feature selection from the jth modality D_j^l (**Section III-A**), then we get the labeled, unlabeled training set and validation set $D_j^{sl}, D_j^{su}, D_j^{sv}$;
3:       Learn metrics and train classifiers with D_j^{sl};
4:   **end for**
5:   Select classifiers with D^{sv} (**Section III-B**);
6:   Vote for D^{su}, compute max and $margin$ for each sample;
7:   Rank D^{su} in a descending order of $cmax + (1 - c)$ $margin$, then perform sample selection (**Section III-C**);
8:   Exploit the selected samples to generate triplets (**Section III-D**);
9:   Input these triplets to the OMKS framework;
10:   Output $f(q, p)$.

---

50 classes in Caltech256 to form three subsets, i.e. Caltech10, Caltech20 and Caltech50, respectively. In other datasets, we pick 10 classes. For each dataset, the number of images for each class equals the number of images of the class that has the minimum size in its sample set. We select half of the images for training, 10% for validation, 10% for query, and the remaining 30% for retrieval evaluation. The experiment is performed on a machine with 3.40 GHz Intel processor, 8 GB memory, and the Matlab software.

2) Descriptors and Kernels: Both global and local feature descriptors are extracted to represent images. The global features we tested include: 1) color histogram (256 dimensions for gray images and 768 dimensions for color images); 2) GLCM coefficients (16 dimensions); 3) Local Binary Pattern (59 dimensions); and 4) GIST features (512 dimensions). The local features we used include: 1) SIFT; 2) dense-SIFT; 3) SURF; 4) Geometric Blur; and 5) PHOG (680 dimensions). We set the vocabulary size as 200 to represent Bag-of-Words (BOW) features except for the PHOG descriptor. Since CNN is effective for image content representation and is trained with color images, we extract DCNN feature (4096 dimensions) using CaffeNet, except for the dataset ImageCLEF. Then we apply PCA to each type of features and retain the first 50 principle components. The full dimension of the original features is retained if it is smaller than 50.

Based on these features, we construct 4 kernels [23]:

RBF kernel: $\kappa(x, x') = \exp(-\frac{\|x-x'\|^2}{r\sigma^2})$, where the parameter r is the mean of the pairwise distance and $\sigma \in \{10^{-2}, 2 * 10^{-2}, 4 * 10^{-2}\}$ is the scale parameter.

Cosine similarity: $\kappa(x, x') = \frac{\langle x, x' \rangle}{\|x\|_2 \|x'\|_2}$. To ensure the similarity value in the range of [0, 1], we adopt $\kappa(x, x') = 0.5 \frac{\langle x, x' \rangle}{\|x\|_2 \|x'\|_2} + 0.5$.

3) Base Classifiers: We perform the feature selection algorithm MCFS on each feature and select 50 dimensions. All dimensions are kept if the original feature is less than 50. To make predictions for unlabeled training images, we construct a series of base classifiers for each kind of feature, i.e. Adaboost_M1 [41] + CART [42], discriminative analysis [43], random forest [44], subspace as well as nearest neighbor with Euclidean distance, RCA [6], DCA [40], LRML [9] and SERAPH [15] metrics.

4) Evaluation Criteria: For each query image, we can rank all of the test images according to their similarities. Here we use the mean Average Precision (mAP) to evaluate the performance of retrieval. Given a query and its R retrieved images, the Average Precision is defined as

$$AP = \frac{1}{L} \sum_{r=1}^{R} prec(r)\delta(r) \qquad (15)$$

where L is the size of the relevant images in the retrieved set, $prec(r)$ is the precision at the rth position, and $\delta(r)$ represents whether the rth retrieved image is relevant to the query or not. $\delta(r) = 1$ when they are relevant; $\delta(r) = 0$, otherwise. The mAP is defined based on the average AP values of all the queries. R is set as the number of images for each class in the pooling set.

5) Compared Methods and Parameter Setting: We compare SSOMKS with the following state-of-the-art metric learning algorithms. For each metric, we concatenate all types of features, and then report the retrieval result.

1) DCA: An efficient supervised metric learning scheme which can exploit both positive and negative constraints [40].

2) LRML: A semi-supervised distance metric learning technique that integrates both labeled and unlabelled samples into an effective graph regularization framework [9].

3) OASIS: A supervised online dual approach that learns a bilinear similarity measure over sparse representations [45].

4) EMR: A scalable graph-based manifold ranking algorithm for image retrieval [46].

5) ITML: An information-theoretic method which minimizes the differential relative entropy between two multivariate Gaussians with constraints [7].

6) DML-eig: An efficient eigenvalue optimization framework for metric learning [16].

7) OMKS: An efficient online metric learning algorithm which learns a flexible nonlinear proximity function with multiple kernels for improving visual search [23].

8) SERAPH: An information-theoretic metric learning approach that does not rely on the manifold assumption [15].

9) HDS: A deep learning framework to learn hash codes and image representations in a point-wise manner [47].

Table I shows the parameter setting of SSOMKS. It was observed that MCFS performs well when the number of selected features is smaller than 50 [26]. Therefore, we set d as 50. The parameter k of kNN in LRML controls the number of nearest neighbors linked in a KNN graph. Commonly, it is tuned in 5-20.

TABLE I
PARAMETER SETTING OF SSOMKS

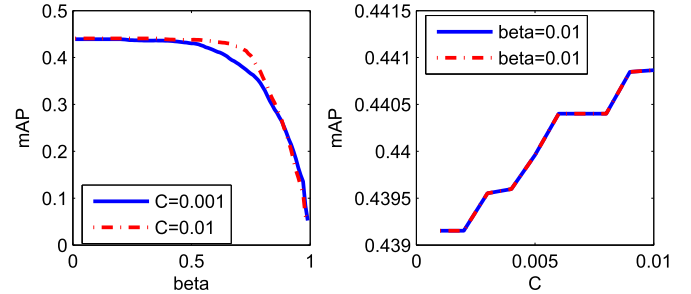| Parameter	d	k	num	λ	C	β
Value	50	20	50	10000	[0,1]	(0,1)

Fig. 3. Retrieval performance versus parameter C and β for the Indoor dataset. The solid lines represent the first round of parameter tuning and dashed lines represent the second round. In each round, first we fix parameter C while tuning β, and then we fix parameter β while tuning parameter C. The algorithm converges after two rounds. Noticing that the curves in the right column are not smooth, and this seems to suggest that the algorithm converges to a local optimum.

As the number of labeled images per class is greater than 20, we set k as 20. For ensemble learning methods including Adaboost, random forest and subspace, we set num as 50. The trade-off parameter λ in MP is used to avoid overfitting. By tuning it in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$ on the validation set, we set it as 10000. The choices of C and β follow from OMKS.

Fig. 3 gives an example of parameter tuning. We only tune several key parameters and set all the remaining to default values. In particular, we set the regularization parameters γ_s, γ_d as 1 due to the lack of prior information and vary the parameter k of kNN in LRML in the range of 5-20. We set the number of the landmarks picked p in EMR as 50 after tuning it on the validation set. For DML-eig, we tune the parameter k in kNN from 1 to the number of the labeled training images per class minus one. All these intervals are chosen as in their released source codes. As for HDS, we use the trained network due to the limited number of labeled images.

B. Performance Analysis

In this section, we conduct a series of experiments on feature selection, classifier and feature analysis, classifier selection, sample selection as well as performance comparisons between the proposed method and other methods.

1) Feature Selection: We compare MCFS [26] with PCA, Laplacian score (LS) [48], discriminative least squares regression (DLSR) [49], CfsSubetEval + GreedyStepwise (CSE+GS) and the original ones (i.e. without feature selection). The feature selection methods select the dimension $d = 10, 20, \ldots, 100, 120, \ldots, 200 \,(15 \,\text{sets})$, except CSE+GS, which can search the optimal number automatically. We perform k-means with the selected features for clustering and use the normalized mutual information (NMI) for evaluation.
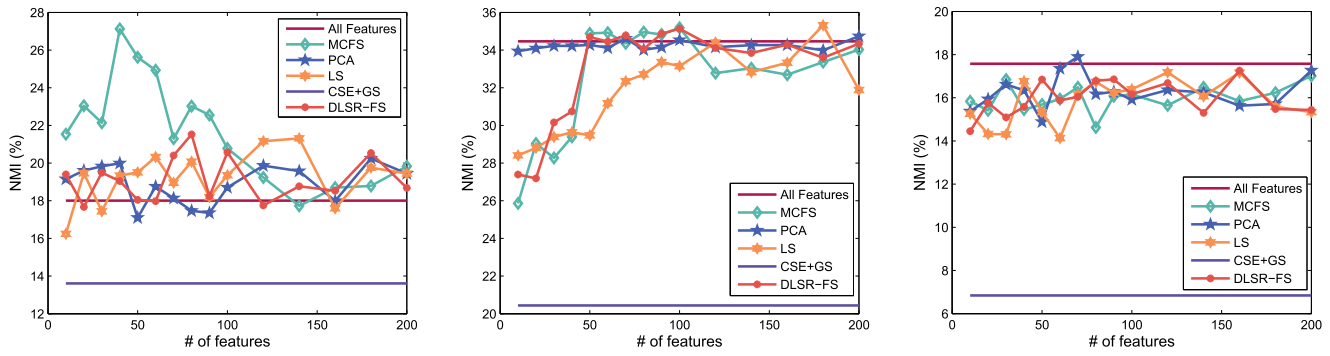
Fig. 4. Clustering performance comparison in terms of NMI versus the number of selected dimensions.
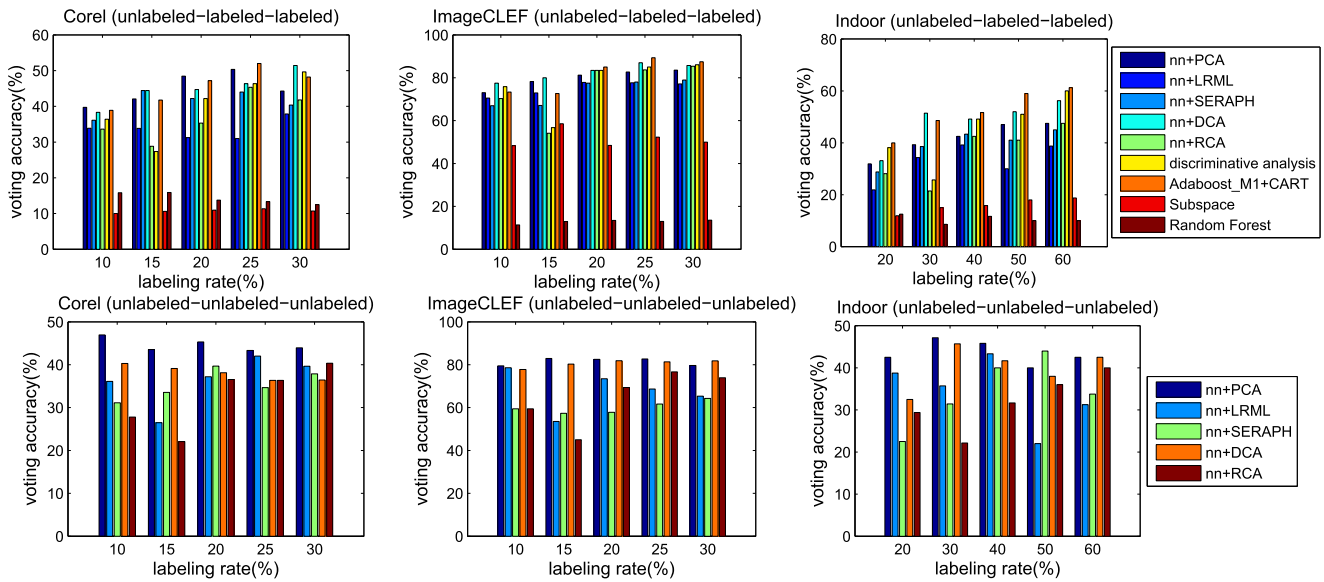

Fig. 5. Performance comparison of different methods on three datasets with varying labeling rates.

Fig. 4 shows the plots of clustering performance versus the number of selected dimensions. In general, PCA offers performance comparable with the original features without performing feature selection. It is clear that MCFS performs well most cases, while CSE+GS has a poor performance due to the lack of supervision information. It can be seen that MCFS significantly outperforms the original ones with SIFT features. This can be attribute to its strong ability in selecting discriminative information in high-dimensional feature spaces.

2) Classifier and Feature Analysis: Fig. 5 shows the voting accuracy of each classifier versus the labeling rate, which represents the proportion of labeled images in the training set. For unlabeled-labeled-labeled triplets, it is clear that the performance of the subspace and random forest is lower than that of other classifiers. In contrast, nn+DCA and Adaboost_M1 + CART consistently exhibit significant advantages, which is partly due to the utilization of supervision information as well as error adaptive adjustment.

Fig. 6 shows the voting accuracy of each feature versus the labeling rate. It can be observed that the performance for each specific feature is data dependent. For instance, the color

histogram outperforms almost all the other features on the Corel dataset, however, it has poor performance on other datasets, which is caused by the single color of the background (i.e. the sunrise is golden, the sky is blue and so on) in Corel. In contrast, as a kind of local features, SIFT is more sensitive to subtle variation in the complex scene, thus having excellent performance on the ImageCLEF and Indoor datasets.

We learn the weight of each feature-classifier pair by minimizing (13), and then report the results in Fig. 7. Generally speaking, it is much easier for the feature-classifier pairs to find the farthest class than the correct class. Therefore, we may observe that these pairs get comparative weights in searching the farthest class. Intuitively, pairs having better performance receive larger weights. We can see that on the ImageCLEF dataset, the weights of Adaboost_M1 + CART and nn+DCA are much larger, while those of subspace and random forests are close to zero in terms of unlabeled-labeled-labeled triplets, which is consistent with the results in Fig. 5.

3) Selective Ensemble Learning Analysis: The process of selective ensemble with different labeling rates is shown in Fig. 8. Intuitively, it is not necessary to exploit all the classifiers for achieving the optimal performance because a part of them may
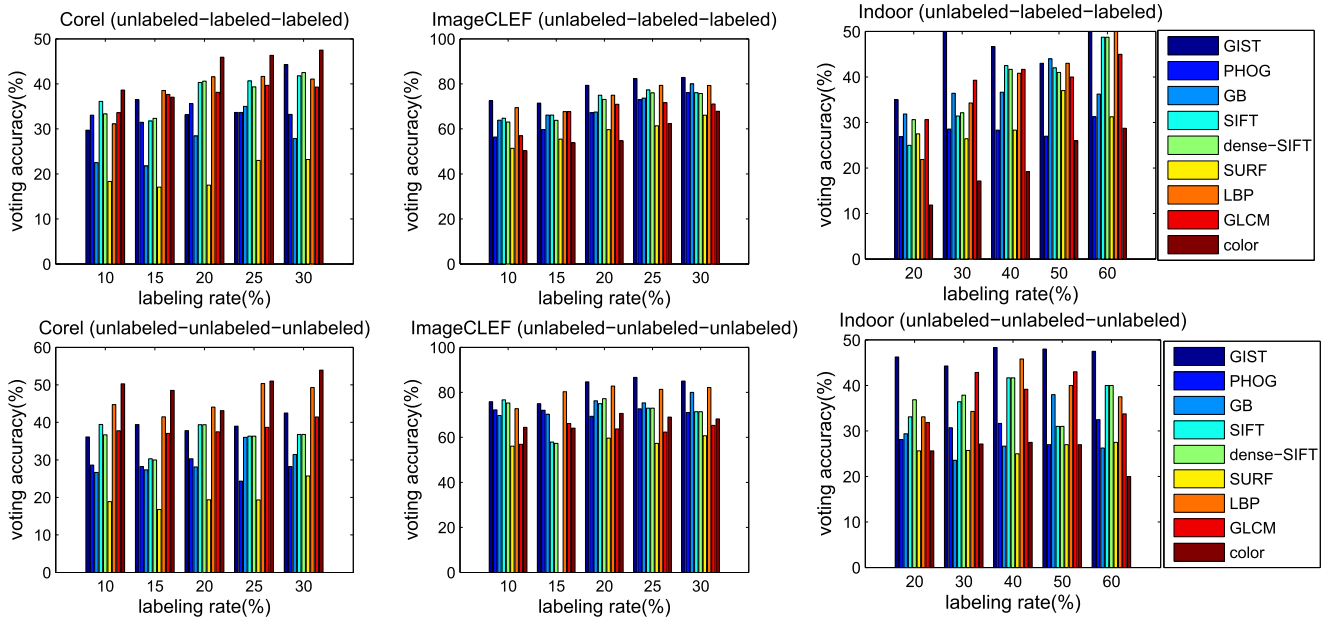
Fig. 6.   Performance comparison of nine different features on three datasets with varying labeling rates.
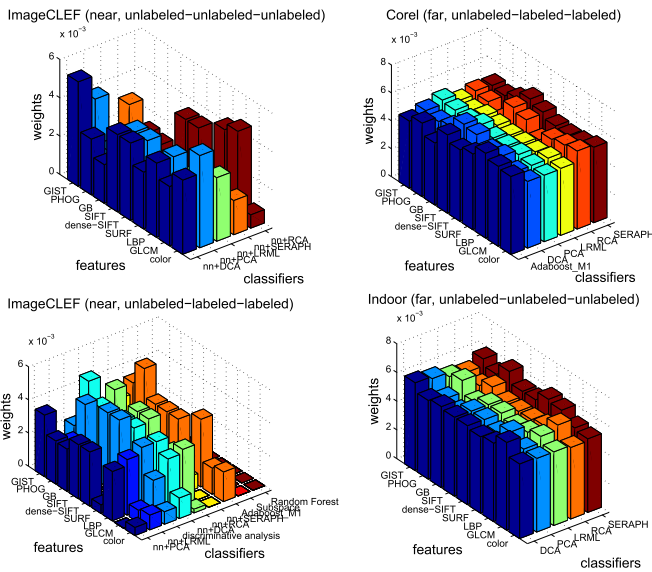

Fig. 7.   Obtained weights by different features and classifiers.

not be necessary. For unlabeled-labeled-labeled triplets, the required number of classifiers is closely related to the complexity of the scene. For instance, the best performance can be obtained with fewer than 50 base classifiers on the Corel and Image-CLEF datasets. However, due to the complicated background in the Indoor dataset, almost all the classifiers are needed when the labeling rate equals 20%. In general, for a certain dataset, the optimal number of fused classifiers decreases with the increase in the labeling rate.

4) Sample Selection Evaluation: Fig. 9 summarizes the performance comparison of different strategies. On the whole, the introduction of sample selection significantly improves the performance. Commonly, voting achieves a higher accuracy than using the best one. As the distribution of the decision-making on

the test and validation set may differ, MP does not always outperform using all the samples. The performance of a certain strategy is data dependent. For instance, as for unlabeled-unlabeled-unlabeled triplets, MP+max performs better than MP+margin on the Corel dataset when the labeling rate is 10%, while the latter performs better on the Indoor dataset when the labeling rate is 20%.

The classification accuracy under a much smaller training ratio is listed in Table II. It is clear that the performance is still acceptable with a relatively small training ratio, especially when sample selection is adopted. The performance is also data-dependent. In particular, a higher accuracy is obtained for ImageCLEF, which is a gray medical dataset having simpler background structure.

5) Performance Comparisons: Tables III–VI summarize the comparison results on the nine datasets, where mAP-9 means mAP with 9 kinds of features, while mAP-10 means adopting DCNN feature as well. Tables III–V imply that SSOMKS significantly outperforms other algorithms with 9 features, while as the labeling rate increases, the supervised algorithm DCA gradually shows its superiority. In the beginning, SSOMKS improves the most, around 15% over the OMKS. With the increase in labeling rate, the improvement decreases. Comparing mAP-9 with mAP-10, it is clear that the utilization of DCNN feature obtains significant improvement. In fact, the performance differences become smaller with better image representation. From Table VI, the deep learning method HDS does not reveal superiority in that the generalization capability is limited without parameter tuning. In terms of computational efficiency, the Euclidean metric takes the least time. Owing to the massive triplets production as well as the time-consuming multiple kernel learning, the time cost of OMKS/SSOMKS grows rapidly as the labeling rate increases, whereas the test process only takes a few seconds. Several techniques such as distributed parallel learning [50] and mini-batch processing [51] could be applied to further
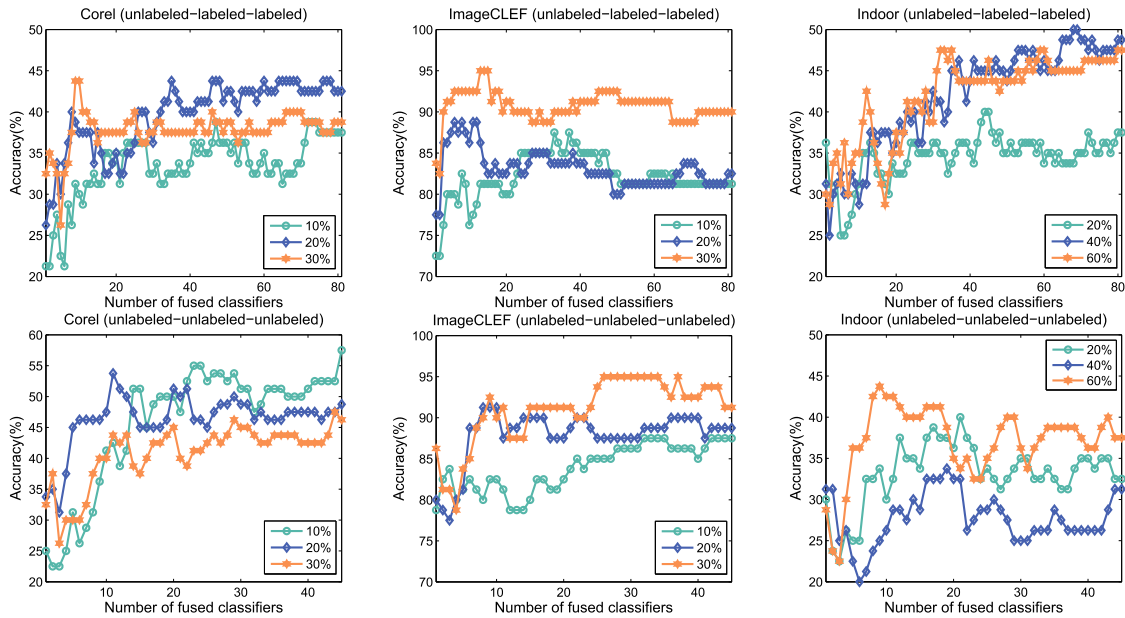
Fig. 8.   Performance comparison in terms of classification accuracy versus the number of fused classifiers.
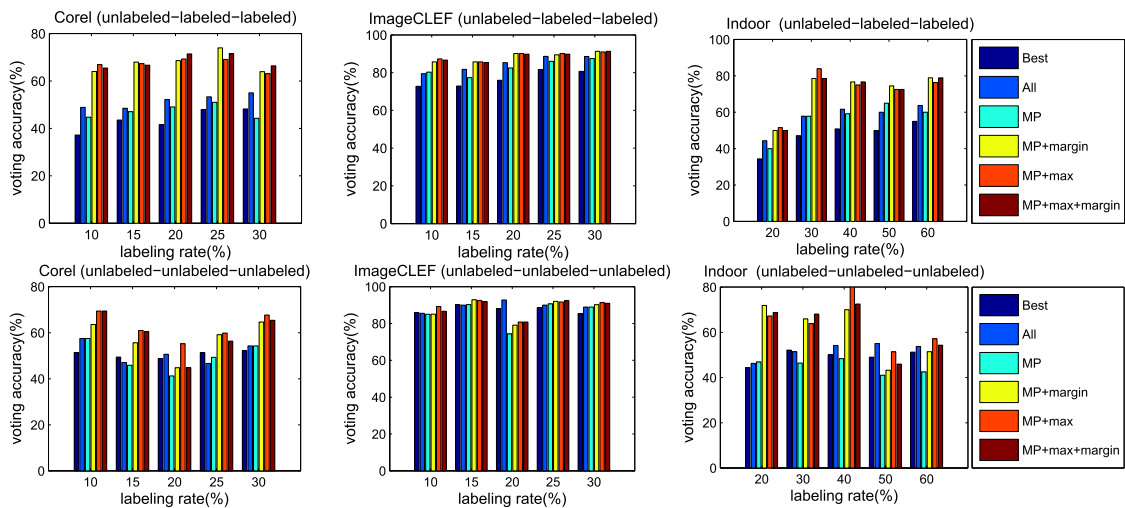
Fig. 9.   Performance comparison of six different strategies in terms of classification accuracy with different labeling rates on three datasets.

TABLE II
CLASSIFICATION ACCURACY(%) ON COREL, IMAGECLEF, AND INDOOR DATASETS WHEN THE TRAINING RATIO IS 20%

Datasets	Triplets	MP	MP+margin	MP+max	MP+max+margin
Corel	unlabeled-labeled-labeled	54.67	71.01	76.81	75.36
	unlabeled-unlabeled-unlabeled	54.67	60.71	69.05	69.05
ImageCLEF	unlabeled-labeled-labeled	85.00	94.06	93.07	94.06
	unlabeled-unlabeled-unlabeled	82.50	81.13	81.13	81.13
Indoor	unlabeled-labeled-labeled	50.00	60.00	60.00	57.78
	unlabeled-unlabeled-unlabeled	58.89	62.22	64.44	62.22

reduce the time cost. Furthermore, as 50% is a relatively high ratio, we can also reduce the proportion of the training set. The average ranks of mAP demonstrate that our proposed method outperforms most of the baseline methods.

Finally, we randomly pick up several query images and compare the top 5 ranked images retrieved with different metric

learning algorithms. Fig. 10 shows the qualitative comparisons of six distinct queries obtained by four diverse algorithms, including OMKS, SSOMKS-max, SSOMKS-margin and SSOMKS-max+margin. From the visual results, it can be observed that SSOMKS retrieves more relevant images than OMKS. For example, for query 1, SSOMKS obtains 4 relevant

TABLE III
PERFORMANCE COMPARISON IN TERMS OF mAP (9 AND 10 TYPES OF FEATURES) AND TIME COST (10 TYPES OF FEATURES) ON
COREL DATASET WITH LABELING RATE EQUALS TO 10%, 15%, 20%, 25%, AND 30%. HDS USES DCNN FEATURE EXTRATED
BY ITSELF. THE BEST AND THE SECOND BEST RESULTS ARE SHOWN IN BOLD AND UNDERLINED, RESPECTIVELY

Labeling Rate	Metric	SSOMKS-max	SSOMKS-max+margin	SSOMKS-margin	OMKS	Euclidean	DCA	LRML	SERAPH	OASIS	ITML	EMR	DML-eig	HDS
10%	mAP-9	0.1589	0.1597	**0.1604**	0.1378	0.0427	0.1284	0.1078	0.0626	0.0280	0.0413	0.0419	0.0579	
	mAP-10	0.3883	0.3881	0.3886	0.3837	0.0855	0.2011	0.1413	0.2997	0.2618	0.3166	0.0529	0.0255	**0.4368**
	time(s)	99.77	97.90	98.15	67.38	**0.40**	0.60	2.96	3.02	21.68	336.46	0.41	1.04	
15%	mAP-9	**0.1541**	0.1535	**0.1541**	0.1374	0.0427	0.1320	0.1065	0.0576	0.0428	0.0697	0.0419	0.0486	
	mAP-10	0.3924	0.3924	0.3924	0.3872	0.0855	0.2842	0.1368	0.3078	0.2732	0.2773	0.0529	0.0799	**0.4401**
	time(s)	377.80	377.80	377.80	137.78	0.52	0.61	3.06	3.07	26.80	311.47	**0.43**	2.24	
20%	mAP-9	0.1561	0.1561	**0.1571**	0.1452	0.0427	0.1280	0.0965	0.0600	0.0389	0.0424	0.0419	0.0638	
	mAP-10	0.3980	0.3980	0.3980	0.3915	0.0855	0.3093	0.1354	0.3093	0.3071	0.2516	0.0529	0.2042	**0.4435**
	time(s)	1330.55	1330.55	1289.27	522.66	**0.34**	0.88	4.56	2.14	144.84	274.28	0.43	4.42	
25%	mAP-9	0.1628	0.1624	**0.1642**	0.1535	0.0427	0.1274	0.0860	0.0602	0.0565	0.0426	0.0419	0.0651	
	mAP-10	0.4032	0.4032	0.4032	0.3986	0.0855	0.3054	0.1406	0.3128	0.3044	0.2582	0.0529	0.2602	**0.4470**
	time(s)	3892.07	3892.07	3892.07	1787.16	0.45	0.59	2.89	1.98	22.18	373.87	**0.37**	6.86	
30%	mAP-9	**0.1665**	0.1651	0.1644	0.1607	0.0427	0.1287	0.0882	0.0427	0.0429	0.0654	0.0419	0.0658	
	mAP-10	0.4102	0.4102	0.4102	0.3978	0.0855	0.3077	0.1402	0.3116	0.3116	0.2553	0.0529	0.2359	**0.4513**
	time(s)	8789.79	8164.26	8164.26	5443.61	**0.41**	0.59	2.94	1.74	22.21	377.62	**0.41**	10.24	
average rank	mAP-9	1.8	2.2	**1.4**	3.6	9.4	4.6	5.6	7.8	10	9	10.8	7.2	
	mAP-10	1.2	1.4	**1**	2.4	8.6	4.6	7.6	3.6	5	5.4	9.8	7.8	–
	time	10.4	9.8	9.8	8.4	**1.4**	2.8	5	4.6	6.8	8.8	**1.4**	4.8	

TABLE IV
PERFORMANCE COMPARISON IN TERMS OF mAP (9 TYPES OF FEATURES) AND TIME COST (9 TYPES OF FEATURES) ON IMAGECLEF
DATASET WITH LABELING RATE EQUALS TO 10%, 15%, 20%, 25%, AND 30%. HDS USES DCNN FEATURE EXTRATED
BY ITSELF. THE BEST AND THE SECOND BEST RESULTS ARE SHOWN IN BOLD AND UNDERLINED, RESPECTIVELY

Labeling Rate	Metric	SSOMKS-max	SSOMKS-max+margin	SSOMKS-margin	OMKS	Euclidean	DCA	LRML	SERAPH	OASIS	ITML	EMR	DML-eig	HDS
10%	mAP	0.5067	**0.5076**	0.5058	0.4391	0.0636	0.3754	0.2711	0.0876	0.0307	0.0666	0.0560	0.0507	0.3959
	time(s)	126.04	120.09	123.77	51.61	5.96	0.64	3.00	2.69	20.10	352.67	**0.38**	1.48	
15%	mAP	0.5022	**0.5053**	0.5004	0.4455	0.0636	0.3951	0.2327	0.0636	0.0674	0.0710	0.0561	0.0837	0.4118
	time(s)	446.33	446.92	446.47	134.52	3.46	0.53	2.95	3.24	27.25	311.18	**0.37**	2.85	
20%	mAP	0.4977	0.5027	**0.5120**	0.4709	0.0636	0.3978	0.2797	0.0636	0.0332	0.0622	0.0561	0.0985	0.4362
	time(s)	1746.00	1973.75	1805.92	587.50	**0.36**	0.49	3.11	1.43	21.06	242.25	0.38	4.24	
25%	mAP	**0.4735**	0.4680	0.4647	0.4728	0.0636	0.4025	0.2682	0.0636	0.0453	0.0573	0.0561	0.1058	0.4535
	time(s)	5993.47	5991.07	5982.98	2088.27	0.42	0.60	3.50	1.43	35.65	300.29	**0.39**	7.11	
30%	mAP	0.5001	**0.5051**	0.4865	0.4887	0.0636	0.4105	0.2688	0.0636	0.0907	0.0630	0.0561	0.0948	0.4702
	time(s)	13298.61	13328.16	13288.83	5735.21	0.41	0.57	3.07	2.09	42.22	306.59	**0.39**	10.66	
average rank	mAP	2	**1.6**	3	3.4	8.8	5	6	8.4	10.2	8.8	10.4	7.8	–
	time	10.6	11	10.2	8.4	3.4	2.6	4.8	4.2	6.8	8.8	**1.2**	4.8	

TABLE V
PERFORMANCE COMPARISON IN TERMS OF mAP (9 AND 10 TYPES OF FEATURES) AND TIME COST (10 TYPES OF FEATURES) ON INDOOR
DATASET WITH LABELING RATE EQUALS TO 20%, 30%, 40%, 50%, AND 60%. HDS USES DCNN FEATURE EXTRATED
BY ITSELF. THE BEST AND THE SECOND BEST RESULTS ARE SHOWN IN BOLD AND UNDERLINED, RESPECTIVELY

Labeling Rate	Metric	SSOMKS-max	SSOMKS-max+margin	SSOMKS-margin	OMKS	Euclidean	DCA	LRML	SERAPH	OASIS	ITML	EMR	DML-eig	HDS
20%	mAP-9	0.1641	**0.1671**	0.1652	0.1440	0.0525	0.0937	0.0893	0.0525	0.0276	0.0504	0.0430	0.0261	
	mAP-10	**0.4461**	**0.4461**	0.4460	0.4424	0.1162	0.3107	0.1460	0.3877	0.3559	0.3337	0.0580	0.0300	0.3754
	time(s)	68.89	69.57	67.95	48.52	**0.38**	0.47	2.67	0.82	14.34	249.81	0.39	1.11	
30%	mAP-9	0.1738	0.1740	**0.1760**	0.1601	0.0525	0.1183	0.0876	0.0525	0.0197	0.0504	0.0430	0.0299	
	mAP-10	**0.4561**	**0.4561**	0.4561	0.4514	0.1162	0.3604	0.1410	0.3925	0.3245	0.3009	0.0580	0.0488	0.3919
	time(s)	220.76	222.51	222.33	154.36	0.52	0.50	2.92	0.84	171.21	284.96	**0.41**	2.76	
40%	mAP-9	0.1727	**0.1741**	0.1737	0.1673	0.0525	0.1253	0.0887	0.0525	0.0214	0.0401	0.0430	0.0289	
	mAP-10	**0.4517**	**0.4517**	**0.4517**	0.4467	0.1162	0.3372	0.1463	0.3878	0.3265	0.2729	0.0580	0.0687	0.4121
	time(s)	809.55	862.23	933.52	655.34	0.50	0.66	2.71	1.09	13.99	275.14	**0.42**	3.59	
50%	mAP-9	0.1701	0.1721	**0.1740**	0.1683	0.0525	0.1235	0.1023	0.0525	0.0533	0.0504	0.0430	0.0310	
	mAP-10	**0.4513**	**0.4513**	**0.4513**	0.4491	0.1162	0.3698	0.1623	0.4063	0.2873	0.3139	0.0580	0.0668	0.4397
	time(s)	2410.37	2410.37	2410.37	2174.43(16)	**0.36**	0.57	2.74	0.96	13.27	279.14	0.41	6.29	
60%	mAP-9	0.1630	0.1611	**0.1644**	0.1621	0.0525	0.1245	0.1026	0.0525	0.0367	0.0496	0.0430	0.0309	
	mAP-10	**0.4551**	**0.4551**	**0.4551**	0.4519	0.1162	0.3840	0.1623	0.3967	0.3854	0.2236	0.0580	0.0663	0.4512
	time(s)	7817.64	7817.64	7817.64	6509.28	**0.34**	0.56	2.77	0.82	16.21	297.96	0.42	12.21	
average rank	mAP-9	2.8	2	**1.4**	3.8	7.2	5	6	7.2	9.8	8.4	9	10.6	
	mAP-10	**1**	**1**	1.2	2.2	8.2	4.8	7.2	3.2	5	5.8	9.8	9.6	–
	time	9.8	10.6	10.2	8.4	**1.6**	2.8	5.4	4	7.2	9.6	**1.6**	5.6	

TABLE VI

PERFORMANCE COMPARISON IN TERMS OF mAP (10 TYPES OF FEATURES) AND TIME COST (10 TYPES OF FEATURES) ON PUBLIC, CALTECH10, CALTECH20, CALTECH50, FLICKR, AND OXFORD DATASETS, RESPECTIVELY. HDS USES DCNN FEATURE EXTRATED BY ITSELF. THE BEST AND THE SECOND BEST RESULTS ARE SHOWN IN BOLD AND UNDERLINED, RESPECTIVELY

Datasets	Metric	SSOMKS-max	SSOMKS-max+margin	SSOMKS-margin	OMKS	Euclidean	DCA	LRML	SERAPH	OASIS	ITML	EMR	DML-eig	HDS
Public	mAP	_0.2054_	_0.2054_	_0.2054_	0.1943	0.0657	0.1460	0.0828	0.1626	0.1306	0.1155	0.0325	0.0995	**0.2293**
	time(s)	599.98	572.79	572.79	376.71	**0.54**	_0.75_	7.57	4.61	15.69	385.58	0.96	1.24	
Caltech10	mAP	0.6074	**0.6086**	_0.6083_	0.5990	0.1683	0.4492	0.2546	0.4509	0.4296	0.3628	0.0957	0.0389	0.4513
	time(s)	119.48	119.76	124.14	61.91	**0.38**	0.63	2.95	1.69	19.56	236.13	_0.39_	2.87	
Caltech20	mAP	**0.4357**	**0.4357**	**0.4357**	0.4140	0.1013	0.3595	0.2813	0.3706	0.3048	0.1914	0.0210	0.2388	0.3340
	time(s)	1884.22	1884.22	1884.22	612.09	**0.68**	_1.43_	15.96	14.27	16.40	309.45	1.74	2.53	
Caltech50	mAP	**0.3750**	**0.3750**	**0.3750**	0.3566	0.0618	0.3038	0.0696	0.2620	0.1376	0.1080	0.0075	0.1822	0.2532
	time(s)	7642.19	7642.19	7642.19	2465.86	**1.60**	7.59	100.74	70.52	27.99	569.88	8.73	_4.28_	
Flickr	mAP	**0.2356**	**0.2356**	**0.2356**	_0.2160_	0.0821	0.1889	0.0931	0.1829	0.1161	0.0912	0.0494	0.1283	0.1000
	time(s)	1423.16	1423.16	1423.16	860.09	**0.36**	_0.56_	4.22	2.32	17.18	274.52	_0.56_	2.18	
Oxford	mAP	**0.1690**	_0.1689_	_0.1689_	0.1554	0.1052	0.1572	0.1244	0.1381	0.1275	0.0817	0.0711	0.0333	0.1421
	time(s)	922.33	979.09	925.55	579.08	**0.36**	0.54	3.01	3.72	16.76	306.90	_0.41_	1.21	
average rank	mAP	1.33	**1.17**	_1.33_	2.67	9.17	3.83	7.67	4	5.83	7.83	10.33	7.17	–
	time	9.83	10.17	10.17	8.5	**1**	_2.5_	5.83	5	6.5	8.67	2.67	3.67	

Fig. 10. Qualitative comparison of image similarity search results on the Corel data set by different algorithms. For each block, the first image is the query, and the results from the first line to the fourth line represents OMKS, SSOMKS-max, SSOMKS-margin, and SSOMKS-max+margin, respectively. The ground truth for the queries are as follows: 1 (sky, jet, plane), 2 (field, horses, mare, foals), 3 (tails, snow, coyote, light), 4 (water, tree, ships, sunset), 5 (mountain, sun, clouds, tree), and 6 (sky, water, monument). The red dots represent the images of the same semantic theme with the queries, and the yellow squares represent the images from different semantic themes.

images, while OMKS only obtains 1. For query 2, SSOMKS obtains the entire relevant images, while OMKS only obtains 3. Overall, SSOMKS outperforms OMKS in image retrieval due to the utilization of unlabeled images.

V. CONCLUSION AND FUTURE WORK

We have presented a semi-supervised online multi-kernel similarity (SSOMKS) learning framework to learn a local metric in image retrieval when the supervision information is limited. We have focused on the use of supervision information to estimate the labels of the unlabeled images. The main new aspect of our work lies in the use of classification confidence to evaluate the labeling process and select the reliably labeled images to train the metric function. Experiments with real-world tasks have shown the effectiveness of the proposed method.

Our work is different from the current trend that encourages learning a globally linear metric and focuses on fully supervised kernel similarity learning. Based on the characteristics in visual tasks, we have analyzed why it is necessary to introduce unlabeled images to metric learning. We have proposed a new method for reliable triplet generation, and also designed a criterion for triplet selection to improve the accuracy and efficiency in similarity learning. The proposed method for triplet generation could also be used in other algorithms that take triplets as their inputs.

Although online approaches are more scalable than the batch processing techniques, they suffer from high computational cost in projections. To further improve the efficiency, reducing the

number of projections and performing distributed learning could be considered. In the future, we will explore the potentials of the techniques such as mini-batch and adaptive sampling for computationally efficient metric learning.

ACKNOWLEDGMENT

REFERENCES

[1] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, 2006.

[2] Y. Jing and S. Baluja, "Visualrank: Applying pagerank to large-scale image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1877–1890, Nov. 2008.

[3] N. Jiang, W. Liu, and Y. Wu, "Order determination and sparsity-regularized metric learning adaptive visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1956–1963.

[4] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *Adv. Neural Inf. Process. Syst.*, vol. 15, pp. 505–512, 2003.

[5] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *CoRR*, 2013. [Online]. Available: http://arxiv.org/abs/1306.6709

[6] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, vol. 3, pp. 11–18.

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 209–216.

[8] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[9] S. C. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 3, pp. 1–26, 2010, Art. no. 18.

[10] P. H. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2464–2471.

[11] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *J. Mach. Learn. Res.*, vol. 13, pp. 519–547, 2012.

[12] J. Wang, A. Kalousis, and A. Woznica, "Parametric local metric learning for nearest neighbor classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1601–1609.

[13] J. Wang, K. Sun, F. Sha, S. Marchand-Maillet, and A. Kalousis, "Two-stage metric learning," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 370–378.

[14] L. Cheng, "Riemannian similarity learning," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 540–548.

[15] G. Niu, B. Dai, M. Yamada, and M. Sugiyama, "Information-theoretic semi-supervised metric learning via entropy regularization," *Neural Comput.*, vol. 26, no. 8, pp. 1717–1762, 2014.

[16] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 1–26, 2012.

[17] B. McFee and G. Lanckriet, "Learning multi-modal similarity," *J. Mach. Learn. Res.*, vol. 12, pp. 491–523, 2011.

[18] P. Xie and E. P. Xing, "Multi-modal distance metric learning," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1806–1812.

[19] X. Lu, Y. Wang, X. Zhou, and Z. Ling, "A method for metric learning with multiple-kernel embedding," *Neural Process. Lett.*, vol. 43, no. 3, pp. 905–921, 2016.

[20] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[21] P. Wu *et al.*, "Online multimodal deep similarity learning with application to image retrieval," in *Proc. 21st Int. Conf. Multimedia*, 2013, pp. 153–162.

[22] P. Wu, S. C. Hoi, P. Zhao, C. Miao, and Z.-Y. Liu, "Online multi-modal distance metric learning with application to image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 454–467, Feb. 2016.

[23] H. Xia, S. C. Hoi, R. Jin, and P. Zhao, "Online multiple kernel similarity learning for visual search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 536–549, Mar. 2014.

[24] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, "Robust distance metric learning with auxiliary knowledge," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1327–1332.

[25] G. Kunapuli and J. Shavlik, "Mirror descent for metric learning: A unified approach," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2012, pp. 859–874.

[26] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 333–342.

[27] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, vol. 14, pp. 585–591.

[28] B. Efron *et al.*, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[29] T. Liu, D. Tao, M. Song, and S. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[30] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, no. 1, pp. 239–263, 2002.

[31] L. Li, Q. Hu, X. Wu, and D. Yu, "Exploration of classification confidence in ensemble learning," *Pattern Recog.*, vol. 47, no. 9, pp. 3120–3131, 2014.

[32] T. Van Gestel *et al.*, "Benchmarking least squares support vector machine classifiers," *Mach. Learn.*, vol. 54, no. 1, pp. 5–32, 2004.

[33] J. Liu *et al.*, *SLEP: Sparse Learning With Efficient Projections*, Arizona State Univ., Tempe, AZ, USA. (2009). [Online]. Available: http://www.yelab.net/software/SLEP/

[34] V. Vapnik, *Estimation of Dependences Based on Empirical Data*. New York, NY, USA: Springer, 2006.

[35] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1537–1544.

[36] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[37] C. Xu, D. Tao, and C. Xu, "Large-margin multi-view information bottleneck," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1559–1572, Aug. 2014.

[38] H. Ahn *et al.*, "Classification by ensembles from random partitions of high-dimensional data," *Comput. Statist. Data Anal.*, vol. 51, no. 12, pp. 6166–6179, 2007.

[39] L. Lam and S. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 27, no. 5, pp. 553–568, Sep. 1997.

[40] S. C. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2006, vol. 2, pp. 2072–2078.

[41] Y. Freund *et al.*, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, vol. 96, pp. 148–156.

[42] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.

[43] C. R. Rao, "The utilization of multiple measurements in problems of biological classification," *J. Roy. Statist. Soc.*, vol. 10, no. 2, pp. 159–203, 1948.

[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[45] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010.

[46] B. Xu *et al.*, "Efficient manifold ranking for image retrieval," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 525–534.

[47] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2015, pp. 27–35.

[48] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 507–514.

[49] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, Nov. 2012.

[50] F. Wang and J. Sun, "Survey on distance metric learning and dimensionality reduction in data mining," *Data. Mining Knowl. Discovery*, vol. 29, no. 2, pp. 534–564, 2015.

[51] Q. Qian, R. Jin, J. Yi, L. Zhang, and S. Zhu, "Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (SGD)," *Mach. Learn.*, vol. 99, no. 3, pp. 353–372, 2015.

Jianqing Liang received the B.E. degree from the School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi, China, in 2013, and is currently working toward the Ph.D. degree at the College of Computer Science and Technology, Tianjin University, Tianjin, China.

Her current research interests include metric learning, semisupervised learning, and machine learning.

Qinghua Hu (M'10–SM'13) received the B.E., M.E., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He previously worked with Harbin Institute of Technology as an Assistant Professor and Associate Professor from 2006 to 2011, and as a Postdoctoral Fellow with the Hong Kong Polytechnic University, Hong Kong, China. He is currently a Full Professor with Tianjin University, Tianjin, China. He has authored or coauthored more than 100 journal and conference papers in the areas of pattern recognition, machine learning, and data mining. His research interests include intelligent modeling, data mining, knowledge discovery for classification and regression.

Prof. Hu is the PC Co-Chair of RSCTC 2010, CRSSC 2012, and ICMLC 2014, and severs as referee for a great number of journals and conferences.

Wenwu Wang received the B.Sc. degree in automatic control in 1997, the M.E. degree in control science and control engineering in 2000, and the Ph.D. degree in navigation guidance and control in 2002, all from Harbin Engineering University, Harbin, China.

He joined Kings College, London, U.K., in May 2002 as a Postdoctoral Research Associate, and transferred to Cardiff University, Cardiff, U.K., in January 2004, where he worked in the area of blind signal processing. In May 2005, he joined the Tao Group Ltd. (now Antix Labs Ltd.), Reading, U.K., as a DSP Engineer, working on algorithm design and implementation for real-time and embedded audio and visual systems. In September 2006, he joined Creative Labs, Ltd., Egham, U.K., as an Engineer, working on 3D spatial audio for mobile devices. Since May 2007, he has been with the Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, U.K., where he is currently a Reader in signal processing, and a Co-Director of the Machine Audition Laboratory. During spring 2008, he was a Visiting Scholar with the Perception and Neurodynamics Laboratory and the Center for Cognitive Science, The Ohio State University, Columbus, OH, USA. He has authored or coauthored more than 150 publications in these areas, including two books: *Machine Audition: Principles, Algorithms and Systems* (IGI Global, 2010) and *Blind Source Separation: Advances in Theory, Algorithms and Applications* (Springer, 2014). His current research interests include blind signal processing, sparse signal processing, audio-visual signal processing, machine learning and perception, machine audition (listening), and statistical anomaly detection.

Prof. Wang is a Member of the Ministry of Defence (MoD) University Defence Research Collaboration (UDRC) in Signal Processing (since 2009), a Member of the BBC Audio Research Partnership (since 2011), an Associate Member of the Surrey Centre for Cyber Security (since 2014), and a Member of the MRC/EPSRC Microphone Network (since 2015). He is currently an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is also Publication Co-Chair of ICASSP 2019 (to be held in Brighton, U.K.). He was a Tutorial Speaker on ICASSP 2013, UDRC Summer School 2014, 2015, and 2016, and SpaRTan/MacSeNet Spring School 2016.

Yahong Han received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2012.

He is currently an Associate Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His current research interests include multimedia analysis, retrieval, and machine learning.