2014

Online Multiple Kernel Similarity Learning for Visual Search

Hao Xia

Chu Hong HOI
Singapore Management University, chhoi@smu.edu.sg

Rong Jin

Peilin Zhao

Online Multiple Kernel Similarity Learning for Visual Search

Hao Xia, Steven C.H. Hoi, Rong Jin, Peilin Zhao

Abstract—Recent years have witnessed a number of studies on distance metric learning to improve visual similarity search in Content-Based Image Retrieval (CBIR). Despite their successes, most existing methods on distance metric learning are limited in two aspects. First, they usually assume the target proximity function follows the family of Mahalanobis distances, which limits their capacity of measuring similarity of complex patterns in real applications. Second, they often cannot effectively handle the similarity measure of multi-modal data that may originate from multiple resources. To overcome these limitations, this paper investigates an online kernel similarity learning framework for learning kernel-based proximity functions, which goes beyond the conventional linear distance metric learning approaches. Based on the framework, we propose a novel Online Multiple Kernel Similarity (OMKS) learning method, which learns a flexible nonlinear proximity function with multiple kernels to improve visual similarity search in CBIR. We evaluate the proposed technique for CBIR on a variety of image data sets, in which encouraging results show that OMKS outperforms the state-of-the-art techniques significantly.

Index Terms—Similarity search, kernel methods, multiple kernel learning, online learning, content-based image retrieval

1 Introduction

Similarity search plays a fundamental role in a variety of multimedia retrieval tasks [1], [2], [3], which has been extensively studied in multimedia and computer vision fields, especially for Content-Based Image Retrieval (CBIR) [4], [5], [6]. The crux of visual similarity search is to find some proximity function that can effectively measure distance/similarity between images [7], [8]. In a conventional CBIR system, given images represented in a vector space, the typical choices of such proximity functions are Euclidean distance and its variants, which are often not flexible enough to measure the proximity of images due to the nature of the fixed rigid functions.

In recent years, researchers have noticed the limitations of conventional rigid proximity functions in image similarity search. To address this issue, one group of active research studies are the Distance Metric Learning (DML) algorithms [5], [9], [10], [11], which usually learn to optimize the distance metric of proximity measure to improve image similarity search in CBIR. Despite the success of various DML algorithms to improve similarity search of CBIR, most existing DML algorithms are limited in two aspects. First, they typically assume the target proximity function follows the form of general Mahalanobis distances and restricts the DML task in finding an optimal linear distance metric, which may limit its capacity of measuring similarity of complex image patterns in real applications. Second, even though there are few existing works [12], [13] for learning nonlinear similarity function with kernel, they usually do not handle the similarity with multiple kernels.

To overcome the limitations of existing work, in this paper, we propose a novel Online Multiple Kernel Similarity (OMKS) learning scheme, which ranks images by learning pairwise similarity of images that are represented in multiple modalities using multiple kernels. Unlike the conventional DML techniques, the target similarity function learned by OMKS can be any nonlinear function in some reproducing kernel Hilbert spaces induced by some predefined kernels. And different from some batch kernel-based DML algorithms, OMKS learns with multiple kernels in an online learning fashion [14]. Thus, OMKS is able to learn a much more flexible and powerful proximity function to improve image similarity search in CBIR.

The OMKS task is however very challenging because it must on one hand learn an optimal kernel-based similarity function for each kernel in each modality, and on the other hand determine an optimal combination of multiple kernels in building the final similarity function for similarity search with all modalities. To attack the challenges, we propose a unifying online learning scheme for OMKS, which learns both the optimal similarity function with each individual kernel and the optimal combination of multiple kernels in a coherent and scalable online learning framework. In particular, we apply the online passive aggressive learning technique [15] to learn the kernel-based similarity function for each individual kernel, and the well-known Hedging online learning technique to learn the optimal combination weights of multiple kernels, from a sequence of triplet training data. As a summary, our key contributions include:

- We propose a novel framework of learning kernel-

- *H. Xia, S.C.H. Hoi, and P. Zhao are with the School of Computer Engineering, Nanyang Technological University, Singapore.*
  R. Jin is with the Department of Computer Science and Engineering, Michigan State University, United States.
  Corresponding E-mail: chhoi@ntu.edu.sg

based proximity functions with multiple kernels for visual similarity search. To the best of our knowledge, this is the first online learning work in CBIR that learns to rank images based on kernel-based similarity function using multiple kernels.
- We present an online learning algorithm for OMKS, which learns both the optimal kernel-based similarity function with an individual kernel and the optimal combination of multiple kernels.
- We conduct an extensive set of experiments to evaluate the performance of the proposed technique for CBIR on several image data sets.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 gives some preliminaries of the related techniques. Section 4 introduces the problem definition and presents the proposed online learning algorithm for OMKS, followed by theoretical analysis in Section 5. Section 6 discusses the experimental results, and section 7 sets out the conclusion of this work.

2 Related Work

This section reviews related work which can be generally grouped into three major categories as follows.

2.1 Distance Metric Learning

Distance Metric Learning (DML) from side information (e.g., relevance feedback logs [16] or user-generated contents of social images [17]) has been actively studied in CBIR for several years. In general, most DML works aim to learn an optimal distance metric in the family of Mahalanobis distances, which can be viewed as an equivalent problem of learning an optimal linear projection of original data into a new space, where Euclidean distance is adopted to measure proximity between objects.

In literature, various DML techniques have been proposed in both machine learning [18], [19], [20] and multimedia [5], [9], [10], [21], [22]. Some well-known techniques include Relevant Component Analysis (RCA) [18], Discriminative Component Analysis (DCA) [5] using the idea of Fisher's Linear Discriminant Analysis, Large Margin Nearest Neighbor (LMNN) [20], Metric Learning by Collapsing Classes [23], learning globally-consistent local distance functions [24], Regularized Distance Metric Learning [9] and Laplacian Regularized Metric Learning (LRML) [11], and so on.

In general, the DML task is cast as a Semi-Definite Programming (SDP) problem due to the impose of Positive Semi-Definite (PSD) constraint on the solution, which is computationally intensive, especially when data is of high dimensionality. Some recent work has attempted to resolve the challenge, such as ITML [25] and OASIS [26]. ITML uses LogDet divergence which can enforce positive semi-definiteness automatically to bypass the PSD constraint. OASIS drops the PSD constraint and resolves the DML task by an online learning algorithm to maximize the large margin criterion. OASIS is in general a linear metric learning method. Our technique is partially inspired to overcome the limitations of OASIS by studying kernel-based learning techniques.

2.2 Kernel-based Learning for Image Retrieval

Kernel-based learning techniques are not new for image retrieval. Our technique differs from the existing kernel-based learning techniques proposed for image retrieval in literature. For example, kernel SVM algorithms have been proposed for active learning in CBIR [27], which however address different types of problems as ours.

In literature, several kernel-based distance metric learning algorithms [28], [29] were proposed for learning similarity functions in CBIR. Also, a family of metric learning algorithms including LMNN and NCA have been shown to be able to be kernelized by KPCA trick [12]. Some recent work also reveals the connections between metric and kernel learning in [13], which may naturally provide kernelization for a larger class of metric learning methods. Our techniques differ from these approaches in two key aspects. First, they are designed to learn with a single kernel while the proposed algorithm learns with multiple kernels; and second, they usually run in a batch learning approach, which does not scale to large-scale applications. In contrast, we present online learning algorithms for learning a similarity function with multiple kernels.

We also note that our work is very different from existing kernel learning studies, such as KernelBoost [30] and nonparametric kernel learning [31], [32], which mainly aim to learn a kernel function/matrix consistent with given constraints. Unlike these studies, the proposed technique learns a kernel-based similarity function, instead of a kernel function, from constraints.

Finally, some recent studies also proposed online learning techniques to learn kernel-based similarity function for text-based image search. For example, PAMIR [3] proposed to learn a discriminative model for the image retrieval from text queries using online learning techniques. Our study differs from PAMIR because PAMIR is specially designed for text-based queries and thus cannot be directly applied to the CBIR task.

2.3 Multiple Kernel Learning

Our work is also closely related to Multiple Kernel Learning (MKL) studies [33], [34], which aim to find the optimal combination of multiple kernels for learning classifiers towards a given classification task. Exemplar algorithms include the convex optimization [33], the Semi-Infinite Linear Program (SILP) approach [34], and the level method [35]. In addition, several recent studies [36], [37] address multiple kernel learning for multi-class and multi-labeled data, and some other works aim at improving its efficiency and generality [38], [39], [40].

Despite sharing the common goal of finding the optimal combination of multiple kernels, our technique differs significantly from the existing MKL studies in two key aspects. First, we aim to learn kernel-based proximity functions for image ranking tasks while conventional MKL studies often address classification tasks. Second, the training data used by conventional MKL studies are in the regular form of single data instances with class label, while the training data in our problem are in the form of triplet instances.

We should note that this work was also inspired by our previous work on Online Multiple Kernel Learning (OMKL) [41], [42]. The OMKL technique was proposed to learn classifiers by finding an optimal combination of multiple kernels for classification tasks, while the goal of this work is to learn the image similarity function from triplets for retrieval tasks in CBIR. In particular, special care is needed in the design of algorithms to handle the triple constraints. Finally, although MKL has successfully been applied to several computer vision applications [43], [44], to the best of our knowledge, this is the first online learning work that learns a similarity function using multiple kernels for CBIR.

3 PRELIMINARIES

To better motivate our work, we introduce some preliminaries of two closely related techniques: (1) online multiple kernel learning, and (2) large scale online learning of image similarity through ranking.

3.1 OMKL

We briefly review the recent work on Online Multiple Kernel Learning (OMKL) [41], [42]. This technique in general aims to address a Multiple Kernel Learning (MKL) problem.

Specifically, given a set of training examples $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, n\}$ where $y_i \in \{-1, +1\}, i = 1, \ldots, n$, and a collection of m kernel functions $\mathcal{K} = \{\kappa_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, i = 1, \ldots, m\}$, the goal of an MKL task is to identify the optimal combination of the kernel matrices, denoted by $\theta = (\theta_1, \ldots, \theta_m)$, which minimizes the margin-based classification error. This can be formulated as the following optimization task:

$$\min_{\theta \in \Delta} \min_{f \in \mathcal{H}_{K(\theta)}} \frac{1}{2} \|f\|^2_{\mathcal{H}_{K(\theta)}} + C \sum_{i=1}^{n} l(f(x_i), y_i) \qquad (1)$$

where $K(\theta)(\cdot,\cdot) = \sum_{i=1}^{m} \theta_i \kappa_i(\cdot,\cdot)$, $l(f(x_i), y_i) = \max(0, 1 - y_i f(x_i))$, and Δ is defined by

$$\Delta = \{\theta \in \mathbb{R}^m_+ | \theta^T e_m = 1\}. \qquad (2)$$

The OMKL technique simplifies the MKL problem by learning the following classification function with multiple kernels:

$$f(x) = \sum_{i=1}^{m} \theta_i \mathrm{sign}(f_i(x)) \qquad (3)$$

The OMKL algorithm iteratively updates the prediction function by the perceptron algorithm, i.e.,

$$f_{t+1,i}(x) = f_{t,i}(x) + z_i(t) y_t \kappa_i(x_t, x) \qquad (4)$$

and learns the combination weights by applying the Hedging online learning technique, i.e.,

$$\theta_i(t+1) = \theta_i(t) \beta^{z_i(t)} \qquad (5)$$

where $\beta \in (0, 1)$ is a discount weight parameter, which is employed to penalize the kernel classifier that performs incorrect prediction at each learning step, and $z_i(t)$ indicates if the i-th kernel classifier makes a mistake on the prediction of the example x_t.

3.2 OASIS

Below we briefly introduce another related work of large scale online learning of image similarity through ranking [26]. Specially, the goal of this problem is to learn a similarity function $S(p_i, p_j)$ that assigns higher similarity scores to pairs of more relevant images, i.e.,

$$S(p_i, p_i^+) > S(p_i, p_i^-), \quad \forall p_i, p_i^+, p_i^- \in P \qquad (6)$$

$$\text{such that} \quad r(p_i, p_i^+) > r(p_i, p_i^-) \qquad (7)$$

where $r(\cdot,\cdot)$ reflects the relevance between two images.

Consider a parametric similarity function that has a bilinear form, $S_W(p_i, p_j) = p_i^T W p_j$ where $W \in \mathbb{R}^{d \times d}$, the goal is to find a parametric similarity function S such that all triplets obey the following constraints:

$$S_W(p_i, p_i^+) > S_W(p_i, p_i^-) + 1 \qquad (8)$$

One can define the following hinge loss function for the triplet:

$$l_W(p_i, p_i^+, p_i^-) = \max\{0, 1 - S_W(p_i, p_i^+) + S_W(p_i, p_i^-)\} \qquad (9)$$

As a result, the *batch* optimization problem of this task can be formulated as:

$$W = \arg\min_{W} \|W\|^2_{Fro} + C \sum_{i} \ell_W(p_i, p_i^+, p_i^-) \qquad (10)$$

The *online* optimization problem is formulated as:

$$W_i = \arg\min_{W} \frac{1}{2} \|W - W_{i-1}\|^2_{Fro} + C \ell_W(p_i, p_i^+, p_i^-) \qquad (11)$$

By initializing $W_0 = I$, the online solution of updating W is given as:

$$W_i = W_{i-1} + \tau_i V_i \qquad (12)$$

where $\tau_i = \min\{C, \frac{l_{W_{i-1}}(p_i, p_i^+, p_i^-)}{\|V_i\|^2}\}$ and $V_i = p_i(p_i^+ - p_i^-)^T$.

4 ONLINE MULTIPLE KERNEL SIMILARITY

We first present our framework for online kernel similarity learning, and then extend it to online multiple kernel similarity learning.

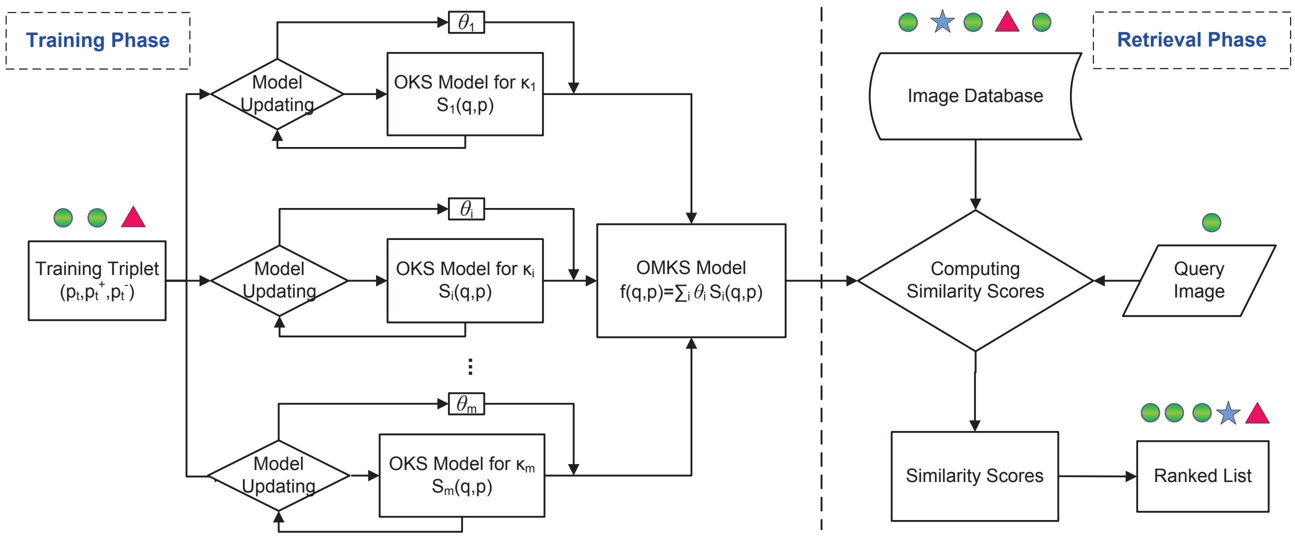


Fig. 1. The system flow of the proposed online multiple kernel similarity learning scheme for visual search.

4.1 Online Kernel Similarity

Our challenge is how to extend the linear similarity function used in OASIS to its kernel version. To this end, for a given kernel $\kappa(\cdot,\cdot)$ and the corresponding Hilbert space $\mathcal{H}$, we introduce a linear operator $L : \mathcal{H} \mapsto \mathcal{H}$ that maps a function $f \in \mathcal{H}$ to another function $L[f] \in \mathcal{H}$. Given the linear operator L, we define the similarity function $S_L(p, q)$ as

$$S_L(q,p) = \langle \kappa(q,\cdot), L[\kappa(p,\cdot)] \rangle_{\mathcal{H}} \quad (13)$$

$q \in \mathcal{X}$ is a query image and $p \in \mathcal{X}$ is an image in database to be retrieved. Compared to the similarity function $S_W(p,q) = p^\top W q$, we observe that the linear operator L plays the same role as matrix W. Let $\mathcal{L}$ be the space that includes all the linear operators in $\mathcal{H}$, i.e.,

$$\mathcal{L} = \{L : \mathcal{H} \mapsto \mathcal{H}, L \text{ is a linear operator}\} \quad (14)$$

Following the framework of OASIS, we develop a framework of kernel similarity learning based on the linear operator. It searches for the optimal linear operator by minimizing the rank loss, i.e.,

$$L_* \;=\; \arg\min_{L\in\mathcal{L}} \|L\|_{HS}^2 + C\sum_i \ell_L(p_i, p_i^+, p_i^-) \quad (15)$$

where $\|\cdot\|_{HS}$ is the Hilbert Schmidt norm of the linear operator, and $\ell_L(p_i, p_i^+, p_i^-) = \max\left(0, 1 - S_L(p_i, p_i^+) + S_L(p_i, p_i^-)\right)$.

Next, we develop an online learning algorithm for efficiently solving (15) based on the online Passive Aggressive (PA) learning [15]. Similar to the OASIS algorithm, in the proposed online learning algorithm, at each trial t, given triplet p_t, p_t^+, p_t^-, we solve the following simple optimization problem

$$L_t = \arg\min_{L\in\mathcal{L}} \frac{1}{2}\|L - L_{t-1}\|_{HS}^2 + C\ell_L(p_t, p_t^+, p_t^-) \quad (16)$$

where we initialize L_0 to be an identity operator at the beginning of online learning. The following proposition gives the closed-form solution to the above optimization.

Proposition 1: The optimal solution to the optimization problem in (16) can be expressed as:

$$L_t = L_{t-1} + \tau_t Z_t \quad (17)$$

where $Z_t \in \mathcal{L}$ is a rank one linear operator and is given by $Z_t[h](\cdot) = \kappa(p_t,\cdot)\left(h(p_t^+) - h(p_t^-)\right)$ for any $h \in \mathcal{H}$. The coefficient τ_t in (17) is calculated as

$$\tau_t = \min\left\{C, \frac{\max\{0, 1 - S_{L_{t-1}}(p_t, p_t^+) + S_{L_{t-1}}(p_t, p_t^-)\}}{\kappa(p_t,p_t)(\kappa(p_t^+,p_t^+) - 2\kappa(p_t^+,p_t^-) + \kappa(p_t^-,p_t^-))}\right\} \quad (18)$$

Proof: We rewrite the problem into a constrained form:

$$\begin{aligned} \min_{L\in\mathcal{L},\xi\geq 0} \quad & \frac{1}{2}\|L - L_{t-1}\|_{HS}^2 + C\xi \\ \text{s. t.} \quad & 1 - \langle \kappa(p_t,\cdot), L[\kappa(p_t^+,\cdot) - \kappa(p_t^-,\cdot)]\rangle_{\mathcal{H}} \leq \xi \end{aligned}$$

We define the Lagrangian as

$$\begin{aligned} g(L,\xi,\tau,\lambda) \;=\; & \frac{1}{2}\|L - L_{t-1}\|_{HS}^2 + C\xi - \lambda\xi \\ & + \tau\left(1 - \xi - \text{tr}(LZ_t^\dagger)\right) \end{aligned}$$

where $\tau \geq 0$ and $\lambda \geq 0$ are Lagrangian multipliers, and $Z_t : \mathcal{H} \mapsto \mathcal{H}$ is a rank one linear operator defined as

$$Z_t[h](\cdot) = \kappa(p_t,\cdot)\langle \kappa(p_t^+,\cdot) - \kappa(p_t^-,\cdot), h\rangle_{\mathcal{H}}$$

and $Z_t^\dagger$ is the adjoint of Z_t. By setting $\frac{\partial g(L,\xi,\tau,\lambda)}{\partial L} = 0$, we have the following

$$\frac{\partial g(L,\xi,\tau,\lambda)}{\partial L} = L - L_{t-1} - \tau Z_t = 0$$

and therefore $L = L_{t-1} + \tau Z_t$. Next, by setting $\frac{\partial g(L,\xi,\tau,\lambda)}{\partial \xi} = 0$, we have

$$C - \tau - \lambda = 0$$

Since $\lambda \geq 0$, we have $\tau \leq C$. Thus, we have:

$$\begin{aligned} g(\tau) &= \frac{1}{2}\tau^2\|Z_t\|_{HS}^2 + \tau(1 - \text{tr}(LZ_t^\dagger)) \\ &= -\frac{1}{2}\tau^2\|Z_t\|_{HS}^2 + \tau(1 - \text{tr}(L_{t-1}Z_t^\dagger)) \end{aligned}$$

Further, by setting $\frac{\partial g(\tau)}{\partial \tau} = 0$, we have

$$\begin{aligned} \frac{\partial g(\tau)}{\partial \tau} &= -\tau\|Z_t\|_{HS}^2 + 1 - \text{tr}(LZ_t^\dagger) \\ &= -\tau\|Z_t\|_{HS}^2 + \ell_{L_{t-1}}(p_t, p_t^+, p_t^-) = 0 \end{aligned}$$

Thus, we have

$$\tau = \frac{\ell_{L_{t-1}}(p_t, p_t^+, p_t^-)}{\|Z_t\|_{HS}^2}$$

Combining the fact that $\tau \leq C$, we prove the proposition by using the fact

$$\begin{aligned} \|Z_t\|_{HS}^2 &= \|\kappa(p_t, \cdot)\|_{HS}^2\|\kappa(p_t^+, \cdot) - \kappa(p_t^-, \cdot)\|_{HS}^2 \\ &= \kappa(p_t, p_t)\left(\kappa(p_t^+, p_t^+) + \kappa(p_t^-, p_t^-) - 2\kappa(p_t^+, p_t^-)\right) \end{aligned}$$

□

Using the above result, we can rewrite $S_{L_t}(q, p)$ as

$$\begin{aligned} & S_{L_t}(q,p) = \langle \kappa(q,\cdot), L_t[\kappa(p,\cdot)]\rangle_{\mathcal{H}} \\ &= \kappa(q,p) + \sum_{l=1}^{t} \tau_l \kappa(q, p_l)(\kappa(p_l^+, p) - \kappa(p_l^-, p)) \end{aligned} \quad (19)$$

Similar to the function learned by support vector machines (SVM), we slightly abuse the concept of Support Vectors (SV) to define each triplet (p_l, p_l^+, p_l^-) of nonzero coefficient $\tau_l > 0$ as a support vector for the learned linear operator. Thus, during the whole training process, we should only keep trace of the support vectors and their coefficients. Algorithm 1 summarizes the proposed algorithm for Online Kernel Similarity (OKS).

Algorithm 1 Online Kernel Similarity (OKS)

INPUT: parameter C, training triplets: (p_t, p_t^+, p_t^-), an input kernel $\kappa(\cdot,\cdot) : \chi\times\chi \to \mathbb{R}$

1: Initialization: $L_0 = I$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Receive a training triplet: (p_t, p_t^+, p_t^-)
4: Compute τ_t in (18)
5: Update L_t as (17)
6: **end for**

OUTPUT: $S(p, q) = \langle \kappa(p, \cdot), L_T[\kappa(q, \cdot)]\rangle$

4.2 Online Multiple Kernel Similarity

We now extend the above online kernel similarity learning problem to the setting of learning with multiple kernels, i.e., the Online Multiple Kernel Similarity (OMKS) learning task. Figure 1 shows the system flow of the proposed OMKS scheme.

Let $\mathcal{K} = \{\kappa_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, i = 1, \ldots, m\}$ be a collection of m kernel functions. Our goal is to identify the optimal combination of the m kernels, denoted by $\theta = (\theta_1, \ldots, \theta_m)$, and consequentially learn the combined kernel similarity function that can be used effectively for image similarity search, i.e.,

$$f(q,p) = \sum_{i=1}^{m} \theta_i S_i(q,p) = \sum_{i=1}^{m} \theta_i \langle \kappa_i(q,\cdot), L_i[\kappa_i(p,\cdot)]\rangle_{\mathcal{H}_{\kappa_i}} \quad (20)$$

where $L_i \in \mathcal{L}_i = \{L : \mathcal{H}_{\kappa_i} \mapsto \mathcal{H}_{\kappa_i}, L \text{ is a linear operator}\}$ and $S_i(q,p) = \langle \kappa_i(q,\cdot), L_i[\kappa_i(p,\cdot)]\rangle_{\mathcal{H}_{\kappa_i}}$ is the similarity function based on the linear operator L_i. To simultaneously learn both the combination weights $\{\theta_i\}_{i=1}^m$ and the linear operators $\{L_i\}_{i=1}^m$, we cast multiple kernel similarity learning into the following optimization problem

$$\min_{\theta\in\Delta} \min_{\{L_i\}_{i=1}^m} \frac{1}{2}\sum_{i=1}^{m} \theta_i\|L_i\|_{HS}^2 + C\sum_{t=1}^{T} \ell(f(p_t, p_t^+) - f(p_t, p_t^-)) \quad (21)$$

where $f(p, q)$ is given in (20), Δ is defined in (2) and $\ell(z)$ is the hinge loss.

Remark. At the first glance, the formulation of multiple kernel similarity learning in (21) may be very different from that for multiple kernel learning in (1). This difference is in fact superficial. According to [41], [42], the problem in (1) is equivalent to the following optimization problem

$$\min_{\theta\in\Delta} \min_{\{f_i\}_{i=1}^m} \frac{1}{2}\sum_{i=1}^{m} \theta_i\|f_i\|_{HS}^2 + C\sum_{t=1}^{T} \ell(f(x_i), y_i) \quad (22)$$

where $f(x) = \sum_{i=1}^m \theta_i f_i(x)$. By comparing (22) to (21), we can find that multiple kernel similarity learning is almost identical to multiple kernel learning except that the kernel prediction functions f_i in (1) is replaced with the linear operator L_i in (21), and the loss functions are somewhat different.

There are two sets of target variables to be learned in the OMKS task, i.e., the combination weights of multiple kernels, and the set of linear operators with respect to each of different kernels. Following the idea of the online multiple kernel learning [41], [42], we apply the Hedging algorithm to online learn the combination weights of multiple kernels, and then apply the online kernel similarity learning algorithm to learn the similarity function of each individual kernel.

Specifically, for each of the m kernels, e.g., κ_i, on every online learning trial, we apply Proposition 1 to find the optimal coefficient for learning the similarity function with respect to kernel κ_i, and then apply the Hedging algorithm to update the combination weights as follows:

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

where $\beta \in (0, 1)$ is a discounting parameter, 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.

Algorithm 2 summarizes the proposed algorithm for Online Multiple Kernel Similarity (OMKS). Finally, we

Algorithm 2 Online Multiple Kernel Similarity (OMKS)

INPUT:

- Kernels: $\kappa_i(\cdot,\cdot): \chi \times \chi \to \mathbb{R}, i = 1, \cdots, m$
- Combination weights $\theta_i(0) = 1, i = 1, \dots, m$
- Discount weight $\beta \in (0, 1)$

```
Initialize L_{0,i} = I, i ∈ [m]
for t = 1, 2, ..., T do
  Receive a training triplet: (p_t, p_t^+, p_t^-)
  for i = 1, 2, ..., m do
    Compute τ_{t,i} in (18) using L_{t-1,i} for L_{t-1}
    Update L_{t,i} by (17) with Z_{t,i} given by Z_{t,i}[h](·) =
    κ_i(p_t, ·) (⟨κ_i(p_t^+, ·), h⟩_{H_{κ_i}} − ⟨κ_i(p_t^-, ·), h⟩_{H_{κ_i}})
    if S_{L_{t-1,i}}(p_t, p_t^+) − S_{L_{t-1,i}}(p_t, p_t^-) ≤ 0 then
      Set z_i(t) = 1
    else
      Set z_i(t) = 0
    end if
    Update θ_i(t) = θ_i(t − 1)β^{z_i(t)}
  end for
end for
```

OUTPUT: $f(q,p) = \sum_{i=1}^m \theta_i(T)\langle \kappa_i(q,\cdot), L_{T,i}[\kappa_i(p,\cdot)]\rangle_{\mathcal{H}_{\kappa_i}}$.

also analyze the theoretical bounds of the two proposed OKS and OMKS algorithms in Section 5.

Remark. It is not difficult to analyze the time complexity of the proposed OKS and OMKS algorithms. Specifically, the time complexity of OKS is $\mathcal{O}(T|SV|)$, where $|SV|$ denotes the size of support vectors of the similarity function, and the time complexity of OMKS is $\mathcal{O}(T|SV|m)$. By assuming small $|SV|$ and m values, both algorithms are generally linear w.r.t. the number of training instances, making the proposed learning scheme efficient and scalable for large applications.

5 Theoretical Analysis

First of all, we analyze the mistake bound of the proposed Online Kernel Similarity (OKS) algorithm as shown in Algorithm 1 in the following theorem.

Theorem 1: Let $(p_1, p_1^+, p_1^-), \dots, (p_T, p_T^+, p_T^-)$ be a sequence of triplet examples, where $p_t, p_t^+, p_t^- \in \mathbb{R}^n$, and assume $\|Z_t\|_{HS}^2 = \kappa(p_t,p_t)(\kappa(p_t^+,p_t^+) - 2\kappa(p_t^+,p_t^-) + \kappa(p_t^-,p_t^-)) \le R$ for all t. Then the number of prediction mistakes M made by OKS on this sequence of examples is bounded by:

$$M \le \min_L \left\{ \frac{1}{\min(1/R,C)} \left[\|L - I\|_{HS}^2 + 2C \sum_{t=1}^T l_L(p_t, p_t^+, p_t^-) \right] \right\}$$

Proof:

$$\begin{aligned}
\Delta_t &= \|L_{t-1} - L\|_{HS}^2 - \|L_t - L\|_{HS}^2 \\
&= \|L_{t-1} - L\|_{HS}^2 - \|L_{t-1} - L + \tau_t Z_t\|_{HS}^2 \\
&= -2\tau_t[(S_{L_{t-1}}(p_t,p_t^+) - S_{L_{t-1}}(p_t,p_t^-)) \\
&\quad -(S_{L_t}(p_t,p_t^+) - S_{L_t}(p_t,p_t^-))] - \tau_t^2\|Z_t\|_{HS}^2 \\
&\ge \tau_t(2\ell_t - \tau_t\|Z_t\|_{HS}^2 - 2\ell_t^*)
\end{aligned}$$

where $\ell_t = \ell_{L_{t-1}}(p_t, p_t^+, p_t^-)$ and $\ell_t^* = \ell_L(p_t, p_t^+, p_t^-)$, thus

$$\sum_{t=1}^T \tau_t(2\ell_t - \tau_t\|Z_t\|_{HS}^2 - 2\ell_t^*) \le \sum \Delta_t \le \|L - I\|_{HS}^2 \quad (23)$$

Since $\tau_t = \min(\ell_t/\|Z_t\|_{HS}^2, C)$, $\tau_t\|Z_t\|_{HS}^2 \le \ell_t$ and $\tau_t \le C$.

$$\sum_{t=1}^T \tau_t(2\ell_t - \tau_t\|Z_t\|_{HS}^2 - 2\ell_t^*) \ge \sum_{t=1}^T (\tau_t\ell_t - 2C\ell_t^*) \quad (24)$$

Combining Equation (23) and (24), we have

$$\sum_{t=1}^T \tau_t\ell_t \le \|L - I\|_{HS}^2 + 2C\sum_{t=1}^T \ell_t^* \quad (25)$$

If the algorithm makes a mistake on round t then $\ell_t \ge 1$. In addition, according to the assumption, we have $\tau_t = \min(\ell_t/\|Z_t\|_{HS}^2, C) \ge \min(1/R, C)$. Thus, we have:

$$\sum_{t=1}^T \tau_t\ell_t \ge \min(1/R, C)M$$

Plugging the above result into the previous equation will result in the conclusion stated in the theorem. □

Secondly, we analyze the mistake bound of the proposed Online Multiple Kernel Similarity (OMKS) algorithm in Algorithm 2. For the convenience of discussions, we define the following notations:

$$\begin{aligned}
\theta_t &= \sum_{i=1}^m \theta_i(t), \quad q_i(t) = \frac{\theta_i(t)}{\theta_t} \\
z_i(t) &= I\left(S_{L_{t-1,i}}(p_t,p_t^+) - S_{L_{t-1,i}}(p_t,p_t^-) \le 0\right)
\end{aligned}$$

where $I(x)$ is an indicator function that outputs 1 when x is true and 0 otherwise. Here, $q_i(t)$ essentially defines the mixture of kernel similarity functions, and $z_i(t)$ indicates if training example (p_t, p_t^+, p_t^-) is misclassified by the ith kernel similarity function at trial t. Finally, we define the optimal margin error $g(\kappa_i, l, \mathcal{L})$ for the kernel $\kappa_i(\cdot,\cdot)$ with respect to a collection of training examples $\mathcal{L} = \{(p_t, p_t^+, p_t^-), t = 1, \dots, T\}$ satisfying $\kappa_i(p_t,p_t)(\kappa_i(p_t^+,p_t^+) - 2\kappa_i(p_t^+,p_t^-) + \kappa_i(p_t^-,p_t^-)) \le R_i$ as

$$g(\kappa_i, l, \mathcal{L}) = \min_L \left\{ \frac{\left[\|L - I\|_{HS}^2 + 2C\sum_{t=1}^T \ell_L(p_t,p_t^+,p_t^-)\right]}{\min(1/R_i, C)} \right\}$$

Theorem 2: After receiving a sequence of T training examples, denoted by $\mathcal{L} = \{(p_t, p_t^+, p_t^-), t = 1, \dots, T\}$ satisfying $\kappa_i(p_t,p_t)(\kappa_i(p_t^+,p_t^+) - 2\kappa_i(p_t^+,p_t^-) + \kappa_i(p_t^-,p_t^-)) \le R_i$, the number of mistakes M made by running the algorithm in Algorithm 2, denoted by

$$\begin{aligned}
M &= \sum_{t=1}^T I(S_{L_{t-1}}(p_t,p_t^+) - S_{L_{t-1}}(p_t,p_t^-) \le 0) \\
&= \sum_{t=1}^T I\left(\sum_{i=1}^m q_i(t-1)z_i(t) \ge 0.5\right)
\end{aligned}$$

is bounded as follows

$$M \leq \frac{2\ln(1/\beta)}{1-\beta} \min_{1\leq i\leq m} \sum_{t=1}^{T} z_i(t) + \frac{2\ln m}{1-\beta} \quad (26)$$

$$\leq \frac{2\ln(1/\beta)}{1-\beta} \min_{1\leq i\leq m} g(\kappa_i, l, \mathcal{L}) + \frac{2\ln m}{1-\beta} \quad (27)$$

By choosing the value of β as $\beta = \frac{\sqrt{T}}{\sqrt{T}+\sqrt{\ln m}}$, we have

$$M \leq 2\left(\left(1+\sqrt{\frac{\ln m}{T}}\right) \min_{1\leq i\leq m} g(\kappa_i, l, \mathcal{L}) + \ln m + \sqrt{T \ln m}\right)$$

The proof to the above theorem essentially combines the results of the passive aggressive learning algorithm [15] and the Hedge learning algorithm. We omit the details of our proof due to space limitation.

6 Experiments

In this section, we conduct an extensive set of experiments to evaluate the efficacy of the proposed algorithms for visual similarity search in CBIR. The data sets and code of our experiments can be found in our project web site http://OMKS.stevenhoi.org/.

6.1 Experimental Testbed

We adopt five publicly available image data sets [1]. These five data sets have been widely used for the benchmark of image retrieval, classification and recognition tasks.

The first testbed is the "Indoor" database[2], which was used for the research of recognizing indoor scenes [45]. This data set consists of 67 indoor categories, and a total of 15620 images. The number of images contained in different categories are diverse, but each category contains at least 100 images. It is further divided into five subsets: *store, home, public spaces, leisure, working place*. We evaluate the performance of different algorithms individually on a randomly picked subset: *public spaces* (we name it "Public", it is also used to evaluate the effect of parameters) as well as the whole indoor collection.

The second testbed is the "Caltech256" database[3], which has been widely adopted for object recognition and image retrieval tasks [46], [26]. This database contains 256 object categories (excluding the background category) and a total of 30607 images. Following the similar experiments as the previous work [26], we pick 10, 20 or 50 out of the 256 classes to form three subsets (the same sets as used in [26]), which are named as "Caltech10", "Caltech20", and "Caltech50", respectively.

The third testbed is the "Corel5000" database [5]. The image testbed consists of real-world photos from COREL image CDs. It has 50 categories, with each category contains exactly 100 images that are randomly selected from relevant examples in the COREL image CDs.

The fourth testbed is the "ImageCLEF" database[4], which was also used in [22]. It is a medical image data set. We also combine "ImageCLEF" with a collection of 100,000 social photos crawled from Flickr, this larger set is named "ImageCLEF+". For the Flickr photos, we treat all of them as the background noisy photos, which are mainly used to test the scalability of our algorithms.

The fifth testbed is the "Oxford Buildings" database[5], which was first used in [47]. It consists of 5062 images collected from Flickr by searching for particular Oxford landmarks. The query set contains 55 queries from 11 different landmarks. We name it "Oxford" for short.

6.2 Experimental Setup

For each data set (except "Oxford" which has no categorial info), we randomly select a subset from each class to make sure that all classes have the same number of images as the one has least images in the original data set. This can avoid the performance being dominated by some single class of large number of images. Based on the data set, we then randomly select 50% examples from each class to form a training set, 10% examples to form a validation set, 10% examples to form a query set, and the rest 30% examples to form the test set for retrieval evaluation. The validation set is mainly used to determine the best parameters and the best cases of the compared algorithms. The final results are averages over 5 splits. We measure both mean and standard deviation of the results, and highlight the best case by performing student t-tests with the significance level $\alpha = 0.05$.

We need to generate side information in the forms of triplet training instances for learning the similarity functions by OASIS and the proposed algorithms, and also pairwise training data instances for the kernelized ITML algorithm (KITML) [13]. In our approach, we generate the side information by sampling triplet constraints from the images in the training set according to their ground truth class labels. Specifically, we generate all positive pairs (two images belong to the same class), and for each positive pair we randomly select another image from another class to form a triplet. Then two pairwise constraints $(p, p^+, +1)$ and $(p, p^-, -1)$ can be derived from the triplet (p, p^+, p^-). After that we randomly sample 20% (i.e., $RatioTrain = 20\%$) of all training instances to form the training set in order to speed up the experiments (as we can see in section 6.6, the performance improves along with the number of training instances increases and then arrives at a saturated value).

For the "Oxford" data set, we randomly split the query set into 5 portions. Then for each split, we test the algorithms 5 times, each time select a different portion as query set, one portion as validation set, the others are left as training set. This can make sure that the average of these 5 runs is the evaluation of the whole query set. The same as the other data sets, the final results are averaged

1. The data sets used in our experiments are all available in our project website:http://www.cais.ntu.edu.sg/~chhoi/OMKS/
2. http://web.mit.edu/torralba/www/indoor.html
3. http://www.vision.caltech.edu/Image_Datasets/Caltech256/
4. http://imageclef.org/
5. http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/index.html

over 5 splits. The side information can be generated easily from the ground truth, we use all positives, and for each positive we randomly select a negative.

We evaluated the performance of all algorithms using some standard performance metrics for ranking in multimedia retrieval. Specifically, for each query image in the query set, all the test images are ranked according to their similarities to the query image, which returns a set of top n similar images for the query. We can measure the precision at top n returned images by computing the number of top-n images of the same class label as the query image. We also adopt the standard mean Average Precision (mAP) to evaluate the retrieval result. In particular, the Average Precision (AP) value is the area under precision-recall curve for a query. The mAP value is calculated based on the average AP value of all the queries. The precision value is the ratio of relevant examples over the total retrieved examples, while recall is the ratio of the relevant examples retrieved over the total relevant examples in the database.

Finally, all of the experiments were run in MATLAB environment on a Linux machine with 3GHz Intel CPU and 16GB RAM.

6.3 Image Descriptors and Kernel Functions

Here we describe how to to extract features from images by different descriptors, and how to compute different kernel functions based on different kinds of features.

6.3.1 Image Descriptors

We adopt both global and local feature descriptors to extract features for representing images in our experiments. We have done some preprocessing of resizing all the images to the scale of 500×500 pixels while keeping the aspect ratio unchanged.

For global features, we extract five kinds of features, including (1) color histogram and color moments (81 dimensions), (2) edge direction histogram (37 dimensions), (3) Gabor wavelets transform (120 dimensions), (4) Local Binary Pattern (59 dimensions), and (5) GIST features (512 dimensions). These global features have been widely used in previous CBIR studies.

For local features, we extract the bag-of-visual-words features using two types of descriptors: (i) the SIFT descriptor — we adopt the Hessian-Affine interest region detector with threshold 500; and (ii) the SURF descriptor — we adopt the SURF detector with threshold 500. For the clustering step, we adopt a forest of 16 kd-trees and search 2048 neighbors to speed up the clustering task. Finally, we adopt the TF-IDF weighing scheme to generate the final bag-of-visual-words representation. By choosing different descriptors (SIFT/SURF) and vocabulary sizes (200/1000), we totally extracted four kinds of local features: SIFT200, SIFT1000, SURF200 and SURF1000. For the "Oxford" data set, we use larger vocabulary sizes (20,000 and 100,000) instead, because it prefers larger code book size.

We apply PCA to all kinds of features and keep the first 50 dimensions (if the original dimension is less than 50, we keep all dimensions) to improve the efficiency of the experiment. For those features whose dimension is larger than 10,000, Singular Value Decomposition (SVD) is performed to keep the first 1,000 dimensions. After dimension reduction, we normalize all feature vectors to unit length.

6.3.2 Kernel Functions

In the above, we represent each image in our database by a total of 9 types of different features. Based on these features, we can build a series of kernel functions on these features. To facilitate the learning tasks, we normalize all the kernel values to the range of [0,1].

We adopt 4 kernel functions to build kernels on each kind of feature, which thus results in a total of 36 different kernels. The 4 kernels used in our approach are described as follows:

- RBF kernel: $\kappa(x, x') = \exp(-\frac{d(x,x')}{\gamma\sigma^2})$, where $d(\cdot,\cdot)$ is the Euclidean distance, the kernel parameter γ is selected as the mean of the pairwise distance, σ is used to control the bandwidth, we select $\sigma \in \{2^{-1}, 2^0, 2^1\}$.
- Kernel using cosine similarity: $\kappa(x, x') = \frac{<x,x'>}{\|x\|_2\|x'\|_2}$. We normalize the kernel $\kappa(x, x') = 0.5\frac{<x,x'>}{\|x\|_2\|x'\|_2} + 0.5$ to the range of [0,1].

6.4 Comparison Algorithms

To extensively examine the efficacy of the proposed algorithms, we have implemented the following algorithms:

- Eucl.-Best: We test the retrieval performance of all kinds of features on the validation set by ranking with Euclidean distance, and then select the best feature of the highest mAP. We report the result of this feature by ranking with Euclidean distance.
- RCA-Best: We train the RCA [48] model on the training set for all the features and test the retrieval performance of all kinds of features on the validation set by RCA, and then select the best feature of the highest mAP value. We report the result of this feature by RCA.
- LMNN-Best: We train the LMNN [20] model on the training set for all the features and test the retrieval performance of all kinds of features on the validation set by LMNN, and then select the best feature of the highest mAP value. We report the result of this feature by LMNN.
- OASIS-Best: We train the OASIS [26] model on the training set for all the features and test the retrieval performance of all kinds of features on the validation set by OASIS, and then select the best feature of the highest mAP value. We report the result of this feature by OASIS.
- KRCA-Best: We train the KRCA [49] model on the training set for all the kernels and test the retrieval performance of all kinds of kernels on the validation

TABLE 1
Experimental results of mAP performance.

Alg.	Metric	Public	Indoor	Caltech10	Caltech20	Caltech50	Corel5000	ImageCLEF	ImageCLEF+	Oxford
Eucl.-Best	mAP	0.1628	0.0439	0.2220	0.1668	0.0982	0.1833	0.4125	0.1224	0.4360
	std	±0.0017	±0.0007	±0.0068	±0.0017	±0.0028	±0.0047	±0.0131	±0.0105	±0.0000
RCA-Best	mAP	0.1595	0.0435	0.2203	0.1656	0.0995	0.1831	0.4563	0.1413	-
	std	±0.0073	±0.0006	±0.0085	±0.0011	±0.0031	±0.0040	±0.0130	±0.0129	
LMNN-Best	mAP	0.1618	0.0447	0.2281	0.1639	0.1021	0.1958	0.4840	0.1367	-
	std	±0.0096	±0.0005	±0.0040	±0.0085	±0.0029	±0.0072	±0.0092	±0.0124	
OASIS-Best	mAP	0.1681	0.0462	0.2365	0.1777	0.0997	0.1841	0.4530	0.1388	0.6078
	std	±0.0047	±0.0009	±0.0061	±0.0062	±0.0023	±0.0056	±0.0138	±0.0121	±0.0422
KRCA-Best	mAP	0.1657	0.0446	0.2451	0.1855	0.1114	0.2292	0.5520	0.1222	-
	std	±0.0040	±0.0007	±0.0052	±0.0038	±0.0016	±0.0052	±0.0165	±0.0133	
KITML-Best	mAP	0.1754	0.0460	0.2513	0.1774	0.0993	0.1909	0.5434	0.1962	-
	std	±0.0048	±0.0022	±0.0047	±0.0079	±0.0030	±0.0075	±0.0108	±0.0164	
KITML-Avg	mAP	0.2083	0.0625	0.2501	0.2291	0.1298	0.3273	0.4839	0.2360	-
	std	±0.0178	±0.0038	±0.0096	±0.0145	±0.0040	±0.0059	±0.0279	±0.0197	
Eucl.-Con	mAP	0.1921	0.0568	0.2052	0.1562	0.1013	0.2542	0.3919	0.1118	0.2032
	std	±0.0033	±0.0019	±0.0093	±0.0038	±0.0032	±0.0066	±0.0184	±0.0092	±0.0000
RCA-Con	mAP	0.1916	0.0566	0.2137	0.1727	0.1061	0.2606	0.4974	0.1509	-
	std	±0.0039	±0.0019	±0.0088	±0.0046	±0.0029	±0.0059	±0.0174	±0.0128	
LMNN-Con	mAP	0.2009	0.0593	0.2265	0.1709	0.1114	0.2808	0.5251	0.1553	-
	std	±0.0049	±0.0020	±0.0103	±0.0048	±0.0034	±0.0080	±0.0150	±0.0185	
OASIS-Con	mAP	0.2004	0.0581	0.2249	0.1659	0.1023	0.2518	0.4751	0.1540	0.4580
	std	±0.0090	±0.0028	±0.0088	±0.0055	±0.0031	±0.0103	±0.0166	±0.0114	±0.0277
KRCA-Con	mAP	0.1958	0.0589	0.241	0.1971	0.1226	0.3376	0.5762	0.1624	-
	std	±0.0037	±0.0021	±0.0153	±0.0055	±0.0021	±0.0091	±0.0222	±0.0098	
KITML-Con	mAP	0.2049	0.0572	0.2468	0.1891	0.1044	0.2757	0.5597	0.1919	-
	std	±0.0090	±0.0055	±0.0164	±0.0096	±0.0034	±0.0073	±0.0215	±0.0104	
OKS-Best	mAP	0.1897	0.0513	0.2653	0.1972	0.1192	0.2407	0.5785	0.2804	0.6698
	std	±0.0044	±0.0014	±0.0088	±0.0099	±0.0058	±0.0091	±0.0099	±0.0114	±0.0142
OKS-Avg	mAP	0.2152	0.0677	0.2601	0.2335	0.1433	0.3542	0.5036	0.2802	0.2312
	std	±0.0088	±0.0024	±0.0160	±0.0041	±0.0049	±0.0060	±0.0150	±0.0104	±0.0051
OMKS-U	mAP	0.218	0.0678	0.252	0.224	0.1382	0.3633	0.5766	0.3794	0.2187
	std	±0.0085	±0.0023	±0.0088	±0.0056	±0.0035	±0.0085	±0.0316	±0.0234	±0.0192
OMKS-W	mAP	0.2187	0.068	0.2559	0.2255	0.1389	0.3654	0.5966	0.3991	0.2709
	std	±0.0084	±0.0024	±0.0084	±0.0056	±0.0035	±0.0085	±0.0289	±0.0214	±0.2709
OMKS	mAP	**0.2453**	**0.0794**	**0.3253**	**0.2690**	**0.1592**	**0.3858**	**0.6681**	**0.4442**	**0.7411**
	std	**±0.0082**	**±0.0022**	**±0.0035**	**±0.0089**	**±0.0045**	**±0.0091**	**±0.0176**	**±0.0084**	**±0.0151**

Note: KITML algorithms are too computationally intensive to run on large data sets, so we report the results by a low rank scheme proposed in [13] instead; some algorithms cannot run on "Oxford" dataset without class labels.

set by KRCA, and then select the best kernel of the highest mAP value. We report the result of this kernel by KRCA.

- KITML-Best: We train the kernelized ITML (KITML) [13] model on the training set for all the kernels, then test the retrieval performance of all kinds of kernels by KITML on the validation set, and finally select the best kernel of the highest mAP value. We report the result of this scheme using KITML with this kernel.
- KITML-Avg: We build an average kernel at first by $\kappa(x,x') = \sum_{i=1}^{m} \frac{1}{m}\kappa_i(x,x')$. Then we report the result of this kernel by KITML.
- Eucl.-Con: We first concatenate all kinds of features together, and then report the result of this feature by ranking with Euclidean distance.
- RCA-Con: We first concatenate all kinds of features together, and then report the result of this feature by RCA [48].
- LMNN-Con: We first concatenate all kinds of features together, and then report the result of this feature by LMNN [20].
- OASIS-Con: We first concatenate all kinds of features together, and then report the result of this feature by OASIS [26].
- KRCA-Con: We first concatenate all kinds of features together, then train the KRCA [49] model on the training set for all 4 kernels of this feature and test the retrieval performance of theses 4 kinds of kernels on the validation set by KRCA, and select the best kernel of the highest mAP value. We report the result of this kernel by KRCA.
- KITML-Con: We first concatenate all kinds of features together, then train the KITML [13] model on the training set for all 4 kernels of this feature, then test the retrieval performance of theses 4 kinds of kernels by KITML on the validation set, and finally select the best kernel of the highest mAP value. We report the result by KITML using this kernel.
- OKS-Best: We train the OKS model by Algorithm 1 on the training set for all the kernels and test the retrieval performance of all kinds of kernels on the validation set by OKS, and then select the best kernel of the highest mAP value. We report the result of this kernel by OKS.
- OKS-Avg: We build an average kernel at first by

$\kappa(x, x') = \sum_{i=1}^{m} \frac{1}{m}\kappa_i(x, x')$. Then we report the result of this kernel by Algorithm 1.
- OMKS-U: Use $f(q, p) = \sum_{i=1}^{m} \frac{1}{m} S_i(q, p)$ instead of $f(q, p) = \sum_{i=1}^{m} \theta_i S_i(q, p)$ for OMKS.
- OMKS-W: Use $f(q, p) = \sum_{i=1}^{m} \theta_i S_i(q, p)$ for OMKS, but the weight is computed as $\theta_i = e^{mAP_i}$. mAP_i is obtained by training the OKS model on the training set and test it on the validation set.
- OMKS: The proposed OMKS algorithm as shown in Algorithm 2.

6.5 Experimental Results

We now present the experimental results of performance evaluations on the data sets. We measure the performance in terms of top-n ($n = 1, 2, \cdots, 5$) precision and the mAP values. We summarize in TABLE 1 the experimental results, measured by mAP, of the compared algorithms on all data sets. Fig. 2 illustrate the details of the top-n precision results on two sampled data sets. In TABLE 1, we highlight the best result in each group in bold font by conducting student t-tests with the significance level $\alpha = 0.05$. We draw several empirical observations from the experimental results as follows.

First of all, by comparing the linear methods based on the best feature, we notice that RCA-Best and LMNN-Best are not guaranteed to outperform Eucl.-Best, while OASIS-Best can achieve consistent improvements over Eucl.-Best.

Second, by comparing the linear methods based on the best feature (RCA-Best and OASIS-Best) with the kernel-based methods using the best kernel (KRCA-Best and OKS-Best), we observed that the kernel methods can improve the performance of the linear methods significantly. OKS-Best consistently outperforms all the other methods based on either single feature or single kernel. We also got the results of full rank KIMTL-Best (KITMLFR-Best) on three small scale data sets ”Public”, ”Caltech10” and ”Caltech20”, their mAP values are 0.1911, 0.2754, 0.1989 respectively. It seems that KITMLFR-Best performs slightly better than OKS-Best (though their difference is not statistically significant). We believe this result is fairly encouraging since OKS-Best is an online learning method while KITMLFR-Best is a batch learning method. Despite their comparable performance, we emphasize OKS-Best is empirically more attractive due to its significant advantage in efficiency and scalability over KITMLFR-Best. In particular, as the time efficiency evaluation shown in TABLE 2, the running time of KITMLFR-Best is at least 700 times of that of OKS-Best, and the gain becomes more significant when the data set size increases. Because of the extremely high computational cost, KITMLFR-Best simply cannot run on large data sets as it will take several months to run these data sets on the same machine, so we report the results by a low rank scheme proposed in [13] instead. The dimension of KITML is set to $1/5$ that of the original dimension and no more than 200. It can be observed that though this low rank scheme can improve the efficiency, it causes some loss in performance, such as for ”Oxford” data set when the dimension is reduced from bout 5,000 to 200, KITML failed to beat the baseline, so they were not include in TABLE 1. These promising results show that the proposed OKS algorithm is able to learn the similarity function more effectively and efficiently than the state-of-the-art techniques.

TABLE 2
Training time (seconds) of KITML versus OKS.

	Public	Caltech10	Caltech20
KITMLFR-Best	887.76	63.54	888.80
KITML-Best	15.07	2.34	15.16
OKS-Best	0.59	0.09	0.43

Third, we found that the methods based on the concatenated feature do not always outperform those based on the best feature. For example, consider the “Euclidean” distance, the feature concatenation approach outperforms the best feature only on dataset ”Public”, ”Indoor”, ”Caltech50”, ”Corel5000”, but fails on the other datasets. This observation implies that the feature concatenation is not optimal for combining different kinds of features.

Fourth, OKS-Avg outperforms OKS-Best on dataset ”Public”, ”Indoor”, ”Caltech20”, ”Caltech50”, ”Corel5000”, but fails on the other data sets. In general, it is hard to conclude which is always better the other. We believe whether or not the average kernel outperforms the best kernel should depend on the properties of the underlying data set and individual kernels. If the best kernel significantly outperforms the other kernels or there are a number of very poor kernels for the given data set, OKS-Best would be more likely to outperform OKS-Avg on such data set.

Fifth, OMKS-U and OMKS-W outperform OKS-Best in most cases, but fail on “Caltech10” and ”Oxford”. This is primarily because some kernels have very poor performance, which in turn reduces the effectiveness of the uniform and the simple weighted combination. This result again motivates the importance of studying more advanced kernel combination approaches.

Finally, by examining the results of the proposed algorithm, we found that it consistently outperforms the other algorithms on all the data sets. This promising result shows that OMKS is able to learn an effective similarity function with multiple kernels by learning the optimal combination weights.

6.6 Evaluation with Varied-Size Training Data

In the previous experiments, we fixed training data by setting the parameter of $RatioTrain$ to 20%. In this section, we evaluate the impact of varied amounts of training triplets for the proposed algorithms, as well as OASIS-Best which also adopts the same amounts of triplets as input.

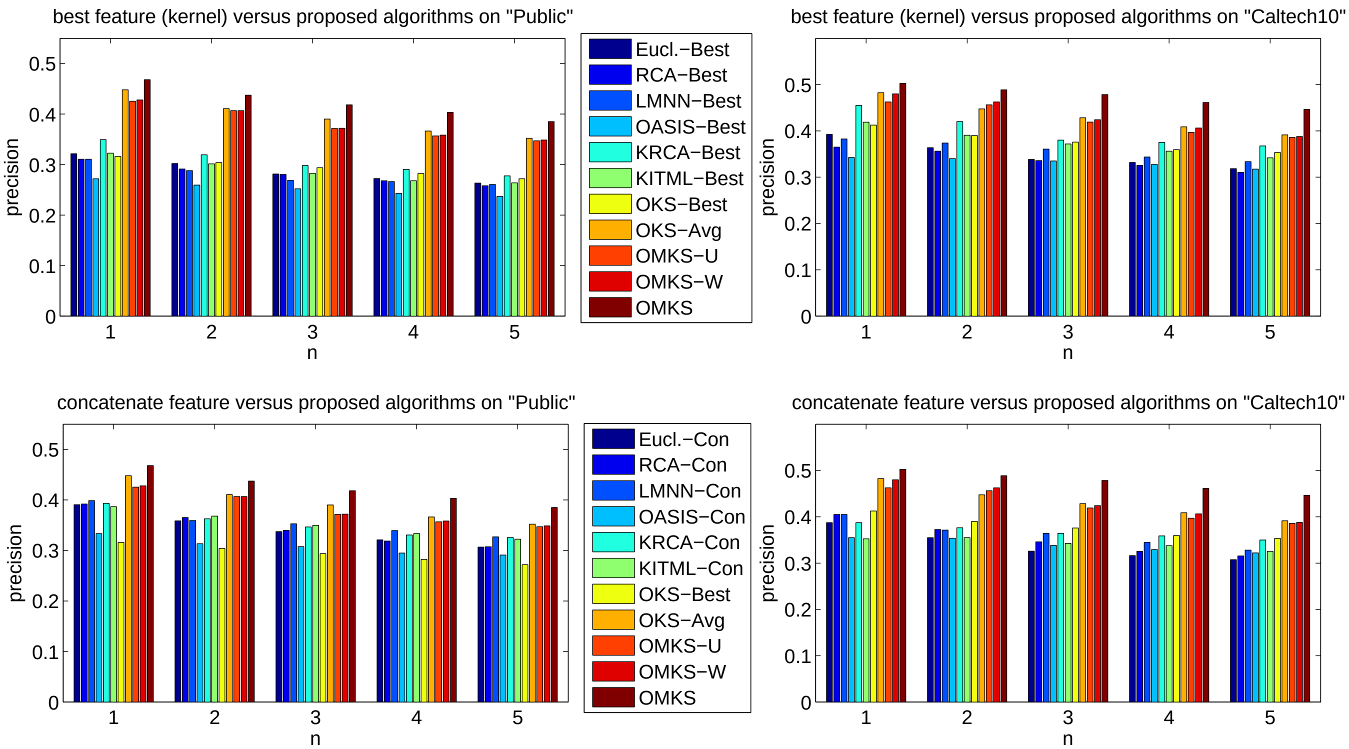


Fig. 2. Top-n precision results on "Public" and "Caltech10" data set.

Fig. 3 shows the evaluation results under varied values of $RatioTrain$ used for building the similarity functions on two sampled data sets. From the results, we observed that all the algorithms in comparison share the similar performance trend as the number of training triplets increases. In particular, the larger the value of $RatioTrain$, the better the retrieval performance can be achieved by the learning algorithms. Moreover, when $RatioTrain$ is large enough, e.g., over 40%, the improvements by most of the learning algorithms tend to become smaller, which is mainly attributed to sufficiently large amount of training data. Finally, similar to the previous experiments, for all the cases under varied values of $RatioTrain$, the proposed OMKS algorithm can perform significantly better than the other competing algorithms.

6.7 Experiments Under Another Setup

Table 3 shows the experimental results obtained by following exactly the same settings as the previous work of OASIS [26], where "OASIS-Ori" is our implementation of OASIS based on the original features used in [26], and the others are the same as those described in Section 6.4 using our own features.

First of all, by comparing "OASIS-Ori" with the results published in [26], they are very similar, where the only slight differences were caused due to the randomization issues, i.e., different random splits, random generated training triplets and cross validation ranges.

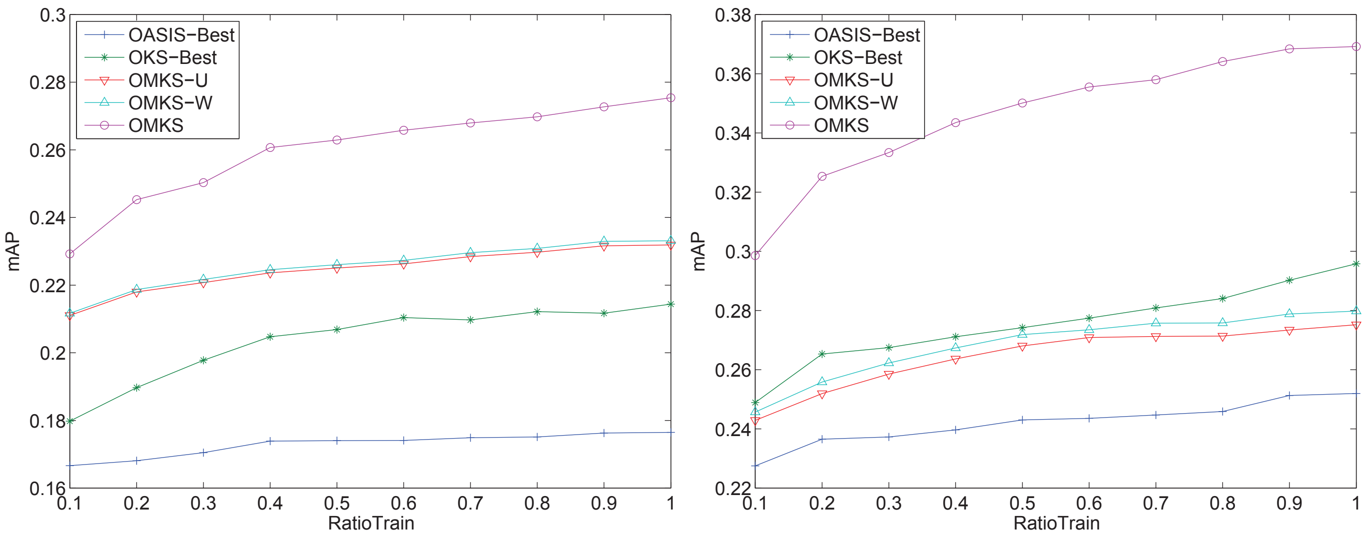
Second, as our implemented algorithms adopt different features, the results of "OASIS-Best" are different from those of "OASIS-Ori". In general, our single best feature tends to perform worse than the original features used in [26]; we conjecture this may be because they have adopted a well-design feature for this particular data set.

Third, by comparing the results of "OMKS", "OKS-Best", "OASIS-Best" and "Eucl.-Best", it is again to validate that our proposed algorithms "OMKS" and "OKS-Best" are significantly more effective by following another different experimental setup.

Finally, no matter which kinds of features used by OASIS, our algorithms can always make consistent improvements over the results of OASIS by following the same experimental setup in [26] .

6.8 Qualitative Comparison

In the last experiment, we sample several query images, and compare the top ranked images retrieved by different methods. Fig. 4 shows the qualitative comparisons of six different query examples obtained by four different algorithms, including "OASIS-Best", "OKS-Best", "OMKS-U" and "OMKS". From the visual results, we observe that in general, "OKS-Best" retrieves more relevant images than "OASIS-Best", as illustrated by the results for the first two queries. This result implies the importance of introducing nonlinear similarity functions in ranking. But at the same time, we notice that for all the other 4 queries, the results by "OKS-Best" are



(a) "Public" (b) "Caltech10"

Fig. 3. Evaluation of *RatioTrain* on both "Public" and "Caltech10" data set.

TABLE 3
Mean average precision and precision at top 1, 10 and 50 on the Caltech object data sets.

Caltech10	OMKS	OKS-Best	OASIS-Best	OASIS-Ori
Mean avg prec.	40±1.7	35±1.3	24±1.1	31±1.1
Top 1 prec.	59±1.8	49±3.3	35±3.4	45±2.1
Top 10 prec.	49±2.5	41±1.2	29±1.4	38±1.8
Top 50 prec.	27±1.3	25±1.0	20±0.5	23±0.3
Caltech20	OMKS	OKS-Best	OASIS-Best	OASIS-Ori
Mean avg prec.	31±0.6	25±0.7	20±0.5	19±0.7
Top 1 prec.	49±2.1	39±0.9	29±1.1	28±2.5
Top 10 prec.	39±0.8	31±1.0	24±0.9	23±1.1
Top 50 prec.	22±0.3	18±0.6	16±0.4	15±0.7
Caltech50	OMKS	OKS-Best	OASIS-Best	OASIS-Ori
Mean avg prec.	20±0.5	15±0.6	10±0.3	12±0.4
Top 1 prec.	36±1.5	26±1.5	16±1.5	20±0.6
Top 10 prec.	27±0.7	20±0.9	13±0.5	16±0.5
Top 50 prec.	15±0.2	12±0.3	9±0.1	10±0.3

not so perfect, which often returns irrelevant images similar to OASIS-Best. The result of query 3 and 4 indicates that "OMKS-U" tends to perform better than "OKS-Best", validating the importance of incorporating multiple kernels built from diverse modalities. On the other hand, "OMKS-U" does not outperform "OKS-Best". For example, for query 5, "OKS-Best" obtained 3 relevant images out of 4, while "OMKS-U" only obtained 1. Overall, OMKS overcomes the limitations of OMKS-U and is able to always find relevant images for all the queries, showing the significance of appropriately weighing individual kernels.

7 Conclusions

This paper addressed a fundamental problem of learning similarity functions for ranking images towards visual similarity search. To overcome the limitations of conventional distance metric learning techniques, we proposed a novel Online Multiple Kernel Similarity (OMKS) learning scheme that can effectively improve image similarity search by learning nonlinear proximity functions beyond conventional linear distance metric learning framework. By exploring the power of multiple kernels in combining multi-modal data, OMKS learns a much more flexible and powerful kernel-based proximity function to improve image similarity search in CBIR. We developed an efficient online learning algorithm and extensively evaluated the proposed algorithms for image similarity search on a number of public image databases. Our empirical results showed that OMKS significantly surpasses the state-of-the-art linear and nonlinear metric learning techniques for image similarity search. Despite being tested on image retrieval tasks, the proposed framework is rather generic for any multimedia retrieval tasks [50]. For future work, we plan to explore more applications and address other practical challenges of the proposed OMKS framework for large-scale applications, such as the convergence rate [51] and budget online learning issues [52].

Acknowledgments

This research was supported by Singapore MOE tier 1 grant (RG33/11).

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, 2008.

[3] D. Grangier and S. Bengio, "A discriminative kernel-based approach to rank images from text queries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1371–1384, 2008.

[4] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. PAMI*, vol. 22, no. 12, pp. 1349–1380, 2000.

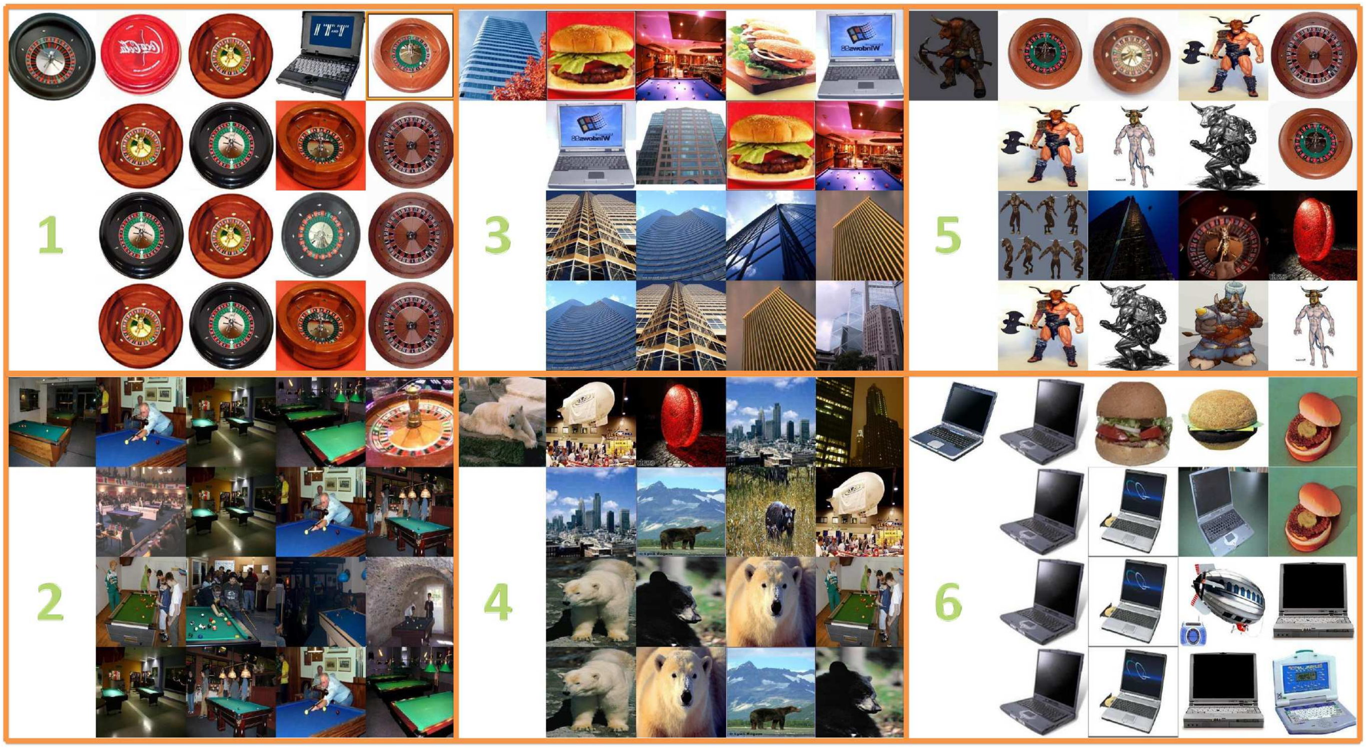


Fig. 4. Qualitative comparison of image similarity search results on the "Caltech10" database by different algorithms. For each block, the first image is the query, and the results from the first line to the fourth line represents "OASIS-Best", "OKS-Best", "OMKS-U" and "OMKS", respectively. The category names for the queries are as follows: 1 (roulette wheel), 2 (billard), 3 (skyscraper), 4 (bear), 5 (minotaur), 6 (laptop).

[5] S. C. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New York, US, Jun. 17–22 2006.

[6] R. Rahmani, S. A. Goldman, H. Zhang, S. R. Cholleti, and J. E. Fritts, "Localized content-based image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1902–1912, 2008.

[7] S. Aksoy and R. M. Haralick, "Probabilistic vs. geometric similarity measures for image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 2357–2362.

[8] A. Qamra, Y. Meng, and E. Y. Chang, "Enhanced perceptual distance functions and indexing for image replica recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 379–391, 2005.

[9] L. Si, R. Jin, S. C. Hoi, and M. R. Lyu, "Collaborative image retrieval via regularized metric learning," *ACM Multimedia Systems Journal*, vol. 12, no. 1, pp. 34–44, 2006.

[10] J.-E. Lee, R. Jin, and A. K. Jain, "Rank-based distance metric learning: An application to image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[11] 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. 18:1–18:26, Aug. 2010.

[12] R. Chatpatanasiri, T. Korsrilabutr, P. Tangchanachaianan, and B. Kijsirikul, "A new kernelization framework for mahalanobis distance learning algorithms," *Neurocomputing*, vol. 73, no. 10-12, pp. 1570–1579, 2010.

[13] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *Journal of Machine Learning Research*, vol. 13, pp. 519–547, 2012.

[14] S. C. Hoi, J. Wang, and P. Zhao, *LIBOL: A Library for Online Learning Algorithms*, Nanyang Technological University, 2012. [Online]. Available: http://LIBOL.stevenhoi.org

[15] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[16] S. C. Hoi, M. R. Lyu, and R. Jin, "A unified log-based relevance feedback scheme for image retrieval," *IEEE Trans. KDE*, vol. 18, no. 4, pp. 509–204, 2006.

[17] P. Wu, S. C.-H. Hoi, P. Zhao, and Y. He, "Mining social images with distance metric learning for automated image tagging," in *Proc. 4th ACM international conference on Web search and data mining (WSDM'11)*, Hong Kong, China, 2011, pp. 197–206.

[18] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, vol. 6, pp. 937–965, 2005.

[19] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems*, 2002.

[20] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems*, 2006, pp. 1473–1480.

[21] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu, "Distance metric learning from uncertain side information with application to automated photo tagging," in *ACM Multimedia*, 2009, pp. 135–144.

[22] L. Yang, R. Jin, L. B. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. C. H. Hoi, and M. Satyanarayanan, "A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 30–44, 2010.

[23] A. Globerson and S. Roweis, "Metric learning by collapsing classes," in *Advances in Neural Information Processing Systems*, 2005.

[24] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *ICCV*, 2007, pp. 1–8.

[25] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *ICML*, 2007, pp. 209–216.

[26] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale

online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, 2010.

[27] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*, Ottawa, Canada, 2001, pp. 107–118.

[28] H. Chang and D.-Y. Yeung, "Kernel-based distance metric learning for content-based image retrieval," *Image Vision Comput.*, vol. 25, no. 5, pp. 695–703, 2007.

[29] D.-Y. Yeung and H. Chang, "A kernel approach for semisupervised metric learning," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 141–149, 2007.

[30] T. Hertz, A. Bar-Hillel, and D. Weinshall, "Learning a kernel function for classification with small training samples," in *ICML*, 2006, pp. 401–408.

[31] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Learning nonparametric kernel matrices from pairwise constraints," in *Proceedings of International Conference on Machine Learning*, 2007, pp. 361–368.

[32] J. Zhuang, I. W. Tsang, and S. C. H. Hoi, "A family of simple non-parametric kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 1313–1347, Jul. 2011.

[33] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[34] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[35] Z. Xu, R. Jin, I. King, and M. R. Lyu, "An extended level method for efficient multiple kernel learning," in *Advances in Neural Information Processing Systems*, 2008.

[36] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proceedings of International Conference on Machine Learning*, Corvalis, Oregon, 2007, pp. 1191–1198.

[37] S. Ji, L. Sun, R. Jin, and J. Ye, "Multi-label multiple kernel learning," in *Advances in Neural Information Processing Systems*, 2008.

[38] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proceedings of International Conference on Machine Learning*, 2009, pp. 1065–1072.

[39] S. V. N. Vishwanathan, Z. sun, N. Ampornpunt, and M. Varma, "Multiple kernel learning and the smo algorithm," in *NIPS*, 2010, pp. 2361–2369.

[40] A. Jain, S. V. N. Vishwanathan, and M. Varma, "Spg-gmkl: Generalized multiple kernel learning with a million kernels," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, August 2012.

[41] R. Jin, S. C. H. Hoi, and T. Yang, "Online multiple kernel learning: Algorithms and mistake bounds," in *ALT*, 2010, pp. 390–404.

[42] S. C. Hoi, R. Jin, P. Zhao, and T. Yang, "Online multiple kernel classification," *Machine Learning*, vol. 90, no. 2, pp. 289–316, 2013.

[43] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[44] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009, pp. 606–613.

[45] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420.

[46] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007.

[47] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[48] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proceedings of International Conference on Machine Learning*, 2003, pp. 11–18.

[49] I. W. Tsang, P. ming Cheung, and J. T. Kwok, "Kernel relevant component analysis for distance metric learning," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2005, pp. 954–959.

[50] S. C. Hoi and M. R. Lyu, "A multimodal and multilevel ranking scheme for large-scale video retrieval," *Multimedia, IEEE Transactions on*, vol. 10, no. 4, pp. 607–619, 2008.

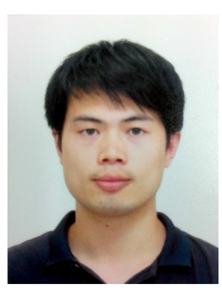
[51] P. Zhao, S. C. Hoi, and R. Jin, "Double updating online learning." *Journal of Machine Learning Research*, vol. 12, pp. 1587–1615, 2011.

[52] P. Zhao, J. Wang, P. Wu, R. Jin, and S. C. Hoi, "Fast bounded online gradient descent algorithms for scalable kernel-based online learning," 2012.

Hao Xia is currently a PhD candidate in the School of Computer Engineering at the Nanyang Technological University, Singapore. He received his bachelor degree from Tsinghua University, Beijing, P.R. China, in 2008. His research interests are statistical machine learning, data mining, and multimedia information retrieval.

Steven C. H. Hoi is an Associate Professor of the School of Computer Engineering at Nanyang Technological University, Singapore. He received his Bachelor degree from Tsinghua University, P.R. China, in 2002, and his Ph.D degree in computer science and engineering from The Chinese University of Hong Kong, in 2006. His research interests are machine learning and data mining and their applications to multimedia information retrieval (image and video retrieval), social media and web mining, and computational finance, etc. He has published over 100 referred papers in top conferences and journals in related areas. He has served as general cochair for ACM SIGMM Workshops on Social Media (WSM'09, WSM'10, WSM'11), program co-chair for the fourth Asian Conference on Machine Learning (ACML'12), book editor for "Social Media Modeling and Computing", guest editor for ACM Transactions on Intelligent Systems and Technology (ACM TIST), technical PC member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including NSF in US and RGC in Hong Kong. He is a member of IEEE and ACM.

Rong Jin is currently an Associate Professor in the department of Computer Science and Engineering at Michigan State University. He received his Ph.D. degree in Computer Science from Carnegie Mellon University, 2003. His research interests are statistical learning and its application to large-scale information management, including web text retrieval, content-based image retrieval, gene regulatory network reconstruction, neuron data analysis, and visual object recognition.



Peilin Zhao is currently a PhD candidate in the School of Computer Engineering at the Nanyang Technological University, Singapore. He received his bachelor degree from Zhejiang University, Hangzhou, P.R. China, in 2008. His research interests are statistical machine learning, and data mining.