Online Multi-modal Distance Learning for Scalable Multimedia Retrieval

Hao Xia, Pengcheng Wu, Steven C.H. Hoi
School of Computer Engineering
Nanyang Technological University
Singapore 639798
{xiah0002,wupe0003,chhoi}@ntu.edu.sg

ABSTRACT

In many real-word scenarios, e.g., multimedia applications, data often originates from multiple heterogeneous sources or are represented by diverse types of representation, which is often referred to as "multi-modal data". The definition of distance between any two objects/items on multi-modal data is a key challenge encountered by many real-world applications, including multimedia retrieval. In this paper, we present a novel online learning framework for learning distance functions on multi-modal data through the combination of multiple kernels. In order to attack large-scale multimedia applications, we propose Online Multi-modal Distance Learning (OMDL) algorithms, which are significantly more efficient and scalable than the state-of-the-art techniques. We conducted an extensive set of experiments on multi-modal image retrieval applications, in which encouraging results validate the efficacy of the proposed technique.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Retrieval models; H.2.8 [**Database Applications**]: Image databases

General Terms

Algorithms, Experimentation

Keywords

Online Learning; Graph Laplacian; Multi-modal Distance; Multimedia Retrieval

1. INTRODUCTION

Similarity search plays a fundamental role in a variety of multimedia retrieval tasks [29, 25, 11, 19, 7, 52], which has been extensively studied in multimedia and computer vision fields, especially for Content-Based Image Retrieval (CBIR) [38, 18, 34]. The crux of similarity search is to find some proximity function that can effectively measure distance/similarity between images [1, 30, 33]. 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 for visual similarity search. To address this issue, one group of active research is the Distance Metric Learning (DML) studies [13, 18, 36, 28, 46, 17, 47], which usually learn to optimize the Mahalanobis distance metric on some vector space to improve visual similarity search for various multimedia retrieval applications. Despite their successes for improving visual similarity search, most existing DML studies are limited in that they usually do not effectively handle the distance measure of multi-modal data that may originate from multiple resources and not necessarily be represented in a vector space.

Recently, the Multiple Kernel Partial Order Embedding (MKPOE) [32] has attempted to address this limitation by adopting multiple kernel learning techniques for integrating multiple sources of heterogeneous data into a single, unified distance space. Despite their pioneering study, MKPOE is limited in two key aspects: (i) it cannot preserve the intrinsic geometric structure of the underlying data, which has been shown to be able to improve the performance of many related applications; (ii) it has poor efficiency and scalability, which cannot be applied to large-scale applications.

To overcome the limitations of MKPOE, this paper proposes a novel Online Multi-modal Distance Learning (OMDL) scheme, which exploits the local dependency of underlying data distributions to enhance the learning efficacy of MKPOE, and further improves the efficiency and scalability by exploring online learning techniques.

The OMDL task is however very challenging because it must on one hand learn an optimal distance function for each kernel in each modality, and on the other hand determine an optimal combination of multiple kernels in building the final distance function with all modalities. To attack the challenges, we propose an online learning algorithm for OMDL, which learns both the optimal distance function with each individual kernel and the optimal combination of multiple kernels in an effective and scalable online learning framework. In particular, we apply the online passive aggressive learning technique [8] to learn the distance function for each individual kernel, and the Hedging learning technique to learn the optimal combination weights of multiple kernels, from a sequence of triplet training data.

WSDM'13, February 4–8, 2012, Rome, Italy.

As a summary, our main contributions include:

- We propose a novel OMDL framework, which exploits local dependency of underly data distribution for learning distance on multi-modal data using multiple kernels via an online learning scheme.
- We present effective OMDL algorithms, which learn both the optimal proximity function with an individual kernel and the optimal combination of multiple kernels in an efficient and scalable online learning approach.
- We conduct an extensive set of experiments to evaluate the performance of the proposed technique for multi-modal image retrieval on several image data sets, in which the encouraging results show clear advantages of OMDL over the state-of-the-art techniques.

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

2. RELATED WORK

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

2.1 Distance Metric Learning

Distance Metric Learning (DML) from side information has been actively studied in CBIR for 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 embedding of original data, where Euclidean distance can be adopted to measure proximity between the embedded objects.

In literature, various DML techniques have been proposed in both machine learning [4, 48, 43] and multimedia communities [18, 36, 10, 28, 45, 50, 45, 47]. Some well-known techniques include Relevant Component Analysis (RCA) [4], Discriminative Component Analysis (DCA) [18], Large Margin Nearest Neighbor (LMNN) [43], Information-Theoretic Metric Learning (ITML) [22], Regularized Metric Learning [36], and Laplacian Regularized Metric Learning (LRML) [17], and so on.

We also note that several kernel-based distance metric learning algorithms [6, 51, 52, 22] were proposed for learning distance functions in CBIR. Despite their successes, most existing DML studies are limited in that they usually do not effectively handle the distance measure of multi-modal data that may originate from multiple resources and not necessarily be represented in vector space.

2.2 Multiple Kernel Learning

Our work is also closely related to Multiple Kernel Learning (MKL) studies [27, 39], which aim to find the optimal combination of multiple kernels for learning classifiers towards a given classification task. Exemplar algorithms include the convex optimization [27], the Semi-Infinite Linear Program (SILP) approach [39], and the level method [49]. In addition, several recent studies [54, 23] address multiple kernel learning for multi-class and multi-labeled data, and some other works aim at improving its efficiency and generality [41, 42, 21].

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 distance functions for multimedia search 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.

Recently, the Multiple Kernel Partial Order Embedding (MKPOE) [32] adopts multiple kernel learning techniques for integrating multiple sources of heterogeneous data into a single, unified distance space. Our work is partially inspired to overcome the limitations of MKPOE by adopting graph Laplacian to capture the underlying data structure so as to improve the effectiveness of MKPOE, and tackling the efficiency and scalability issue by solving the learning problem via online learning.

We should note that this work was also inspired by our previous work on Online Multiple Kernel Learning [24, 16]. 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 distance function from triplets for multimedia retrieval [16]. Unlike the regular classification task, special care is needed in the design of algorithms to handle the triple constraints for multimedia retrieval.

3. PRELIMINARIES

To better motivate our work, in this section, we introduce an important previous work, i.e., Multiple Kernel Partial Order Embedding (MKPOE) [32], for learning distance with multiple kernels. For problem setting, assume we are given a set of n objects $\mathcal{X} = \{\mathbf{x}_i | i = 1, 2, \ldots, n\}$, and a collection of triplet constraints $\mathcal{C} = \{(i_t, j_t, k_t), t = 1, 2, \ldots, T\}$, where each triplet (i_t, j_t, k_t) indicates that object $\mathbf{x}_{i_t}$ is similar to object $\mathbf{x}_{j_t}$, but dissimilar to object $\mathbf{x}_{k_t}$. For simplicity, we will simply denote $\mathbf{x}_{i_t}$ as $\mathbf{x}_i$ for the rest of discussion.

We first consider the case of learning distance with a single kernel. In particular, it first maps the data into a Reproducing Kernel Hilbert Space (RKHS) [2] $\mathcal{H}$ via a feature map ϕ with corresponding kernel function $\kappa(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}$; then the data is mapped to $\mathbb{R}^d$ by a linear projection $\mathbf{M} : \mathcal{H} \to \mathbb{R}^d$. The embedding function $g : \mathcal{X} \to \mathbb{R}^d$ is therefore the composition of the projection $\mathbf{M}$ with ϕ:

$$g(\mathbf{x}) = \mathbf{M}(\phi(\mathbf{x})) \tag{1}$$

It is further extended to the multiple kernel case by defining the embedding function as the concatenation:

$$g(\mathbf{x}) = (\mathbf{M}^p(\phi^p(\mathbf{x})))_{p=1}^m \tag{2}$$

where the index of kernel $p \in \{1, 2, \ldots, m\}$ and $(\cdot)_{p=1}^m$ denotes concatenation. By invoking the representer theorem [35] for each $\mathbf{M}^p$:

$$\mathbf{M}^p = \mathbf{N}^p (\mathbf{\Phi}^p)^\top \tag{3}$$

where $\mathbf{N}^p \in \mathbb{R}^{d \times n}$ is a real-valued matrix, $\mathbf{\Phi}^p$ is a matrix representation of $\mathcal{X}$ in $\mathcal{H}$ (i.e., $\mathbf{\Phi}_i^p = \phi^p(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathcal{X}$).

So the embedding function can now be written as:

$$g(\mathbf{x}) = (\mathbf{M}^p(\phi^p(\mathbf{x})))_{p=1}^m = (\mathbf{N}^p \mathbf{K}_{\mathbf{x}}^p)_{p=1}^m \tag{4}$$

where $\mathbf{K}_{\mathbf{x}}^p$ is the column vector formed by evaluating the kernel function κ^p at $\mathbf{x}$ against $\mathcal{X}$.

Further we use a Positive Semi-Definite (PSD) matrix $\mathbf{W}^p = (\mathbf{N}^p)^\top(\mathbf{N}^p)$. The optimization problem of MKPOE can be formally written as follows:

$$\min_{\mathbf{W}^p,\xi} \quad \sum_{p=1}^{m} tr(\mathbf{W}^p\mathbf{K}^p) + \frac{\beta}{|\mathcal{C}|}\sum_{\forall(i,j,k)\in\mathcal{C}}\xi_{ijk} \tag{5}$$

$$s.t. \quad d(\mathbf{x}_i,\mathbf{x}_j) \doteq \sum_{p=1}^{m}(\mathbf{K}_i^p - \mathbf{K}_j^p)^\top\mathbf{W}^p(\mathbf{K}_i^p - \mathbf{K}_j^p)$$

$$d(\mathbf{x}_i,\mathbf{x}_j) + 1 \le d(\mathbf{x}_i,\mathbf{x}_k) + \xi_{ijk},\ \xi_{ijk} \ge 0, \forall(i,j,k)\in\mathcal{C}$$

$$\mathbf{W}^p \succeq \mathbf{0} \quad p = 1,2,\ldots,m$$

where $\mathbf{K}^p$ is the kernel matrix corresponding to κ^p, $\mathbf{K}_i^p$ is the i-th column of it, and β is a trade-off parameter.

4. ONLINE MULTI-MODAL DISTANCE LEARNING (OMDL)

4.1 Overview

In this section, we present a framework of Online Multi-modal Distance Learning (OMDL) using multiple kernels. We first give our formulation of multi-modal distance learning and then propose the online algorithms.

4.2 Multi-modal Distance Learning

Let us denote by $f(\mathbf{x}_i,\mathbf{x}_j)$ a similarity function that measures the similarity between any two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{S} \in \mathbb{R}^{n\times n}$ is a similarity matrix where each element $S_{ij} = f(\mathbf{x}_i,\mathbf{x}_j) \ge 0$. Note that $f(\cdot,\cdot)$ does not have to be a kernel function that satisfies the Mercer's condition. Given n data instances, a kernel matrix $\mathbf{K}$ can be expressed as $\mathbf{K} = \mathbf{V}'\mathbf{V} \succeq \mathbf{0}$, where $\mathbf{V} = [\mathbf{v}_1,\ldots,\mathbf{v}_n]$ is the matrix of the embedding of the n data instances. The regularizer of the kernel matrix $\mathbf{K}$, which captures the local dependency [15, 14, 53] between the embeddings of $\mathbf{v}_i$ and $\mathbf{v}_j$ (i.e., the low dimensional embedding [37] of similar instances should be similar w.r.t. the similarity S_{ij}) can be defined as:

$$\begin{aligned}\mathbf{\Omega}(\mathbf{V},\mathbf{S}) &= \frac{1}{2}\sum_{i,j=1}^{n} S_{ij}\|\frac{\mathbf{v}_i}{\sqrt{D_i}} - \frac{\mathbf{v}_j}{\sqrt{D_j}}\|_2^2 \\ &= tr(\mathbf{V}\mathbf{L}\mathbf{V}') = tr(\mathbf{L}\mathbf{K})\end{aligned} \tag{6}$$

where $\mathbf{L}$ is the graph Laplacian matrix defined as:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{1/2} \tag{7}$$

where $\mathbf{D} = diag(D_1, D_2, \ldots, D_n)$ is a diagonal matrix with the diagonal elements defined as $D_i = \sum_{j=1}^{n} S_{ij}$. By incorporating graph Laplacian regularizer, we can formulate the optimization problem of the proposed multi-modal distance learning:

$$\min_{\mathbf{W}^p,\xi} \quad \sum_{p=1}^{m} tr(\mathbf{K}^p\mathbf{W}^p\mathbf{K}^p\mathbf{L}^p) + \frac{\beta}{|\mathcal{C}|}\sum_{\forall(i,j,k)\in\mathcal{C}}\xi_{ijk} \tag{8}$$

$$s.t. \quad d(\mathbf{x}_i,\mathbf{x}_j) \doteq \sum_{p=1}^{m}(\mathbf{K}_i^p - \mathbf{K}_j^p)^\top\mathbf{W}^p(\mathbf{K}_i^p - \mathbf{K}_j^p)$$

$$d(\mathbf{x}_i,\mathbf{x}_j) + 1 \le d(\mathbf{x}_i,\mathbf{x}_k) + \xi_{ijk},\ \xi_{ijk} \ge 0, \forall(i,j,k)\in\mathcal{C}$$

$$\mathbf{W}^p \succeq \mathbf{0} \quad p = 1,2,\ldots,m$$

In the above, the first regularizer term of the objective function enforces the smoothness of the distance on the manifold, which is the key difference as compared to the previous MKPOE approach. The above optimization can be solved by applying projected gradient descent by following a solution similar to MKPOE [32]. However, such an approach can be computationally intensive, which is hardly scalable for large-scale application. In the following, we will present an online learning algorithm to tackle this challenge.

4.3 Algorithms for OMDL

In this section, we present an online learning scheme which aims to sequentially updates the multi-modal distance function from a sequence of triplet constraints. In particular, given any received triplet constraint (i,j,k), it can be used to update two sets of target variables: (i) the parameters of the distance function defined on each individual kernel, i.e., $\mathbf{W}^p, p = 1,2,\ldots,m$ and (ii) the combination weights assigned to different kernels, denoted as $\mu_p, p = 1,2,\ldots,m$. Figure 1 shows the system flow of the proposed OMDL algorithm which learns distance functions from a sequence of triplet constraints.

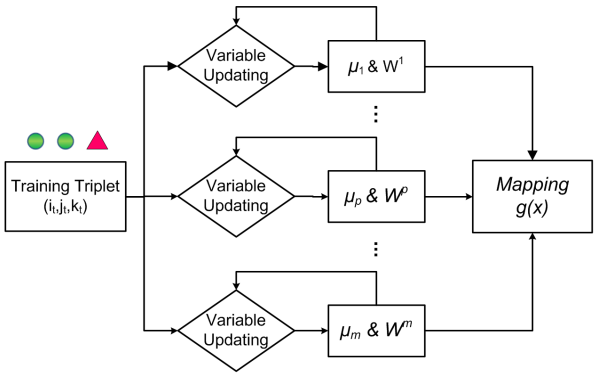

Figure 1: The system flow of the proposed OMDL algorithm from a sequence of training triplets.

Given these variables, the class of the combined kernel can be expressed as:

$$\mathbf{K}_\mu = \sum_{p=1}^{m}\mu_p\mathbf{K}^p\mathbf{W}^p\mathbf{K}^p \tag{9}$$

Correspondingly, the embedding function is:

$$g(\mathbf{x}) = (\sqrt{\mu_p}\mathbf{N}^p\mathbf{K}_\mathbf{x}^p)_{p=1}^m \tag{10}$$

The inner products between embedded points take the form:

$$\begin{aligned}\mathbf{A}_{ij} &= \langle g(\mathbf{x}_i), g(\mathbf{x}_j)\rangle \\ &= \sum_{p=1}^{m}(\sqrt{\mu_p}\mathbf{N}^p\mathbf{K}_i^p)^\top(\sqrt{\mu_p}\mathbf{N}^p\mathbf{K}_j^p) \\ &= \sum_{p=1}^{m}\mu_p(\mathbf{K}_i^p)^\top(\mathbf{N}^p)^\top(\mathbf{N}^p)(\mathbf{K}_j^p)\end{aligned} \tag{11}$$

Thus, the squared Euclidean distance can be expressed as:

$$\begin{aligned}d(\mathbf{x}_i,\mathbf{x}_j) &= \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|^2 \\ &= \sum_{p=1}^{m}\mu_p(\mathbf{K}_i^p - \mathbf{K}_j^p)^\top(\mathbf{N}^p)^\top(\mathbf{N}^p)(\mathbf{K}_i^p - \mathbf{K}_j^p) \\ &= \sum_{p=1}^{m}\mu_p d_p(\mathbf{x}_i,\mathbf{x}_j)\end{aligned} \tag{12}$$

Therefore, the key challenges of online multi-modal distance learning are to find effective and efficient solutions for solving both W^p and μ_p at each online learning round.

4.3.1 Algorithm on a Single Kernel

We first start by presenting the solution for online distance learning on a single kernel.

By following the principle of online passive aggressive learning [8], we can rewrite the formulation of the Online Distance Learning (ODL) as the following optimization for a single triplet (i, j, k) at some round t:

$$\begin{aligned} \min_{\mathbf{W},\xi} \quad & \frac{1}{2}\|\mathbf{W}-\mathbf{W}_{t-1}\|_F^2 + C_1 tr(\mathbf{KWKL}) + C_2\xi \quad (13)\\ s.t. \quad & d(\mathbf{x}_i,\mathbf{x}_j) \doteq (\mathbf{K}_i-\mathbf{K}_j)^\top \mathbf{W}(\mathbf{K}_i-\mathbf{K}_j)\\ & d(\mathbf{x}_i,\mathbf{x}_j)+1 \le d(\mathbf{x}_i,\mathbf{x}_k)+\xi,\ \xi\ge 0\\ & \mathbf{W}\succeq \mathbf{0} \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

In the following, we derive the solution to the ODL problem. At first, we solve the optimization problem without considering the PSD constraint, i.e., $\mathbf{W}\succeq\mathbf{0}$. Then the PSD constraint is applied by projecting onto the feasible set. The following proposition shows the analytic solution to the optimization without the PSD constraint.

PROPOSITION 1. *The closed-form solution to the optimization task in (13) can be expressed as follows:*

$$\mathbf{W} = \mathbf{W}_{t-1} - C_1\mathbf{KLK} - \tau\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K} \quad (14)$$

where τ is calculated by

$$\tau = \min\{C_2, \frac{h(l_{t-1})}{\|\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\|_F^2}\} \quad (15)$$

$h(x)=\max\{0,x\}$ is the hinge loss function, $\mathbf{E}_{ij}=(\mathbf{e}_i-\mathbf{e}_j)(\mathbf{e}_i-\mathbf{e}_j)^\top$ and

$$\begin{aligned} l_{t-1} \quad = \quad & -C_1 tr(\mathbf{KLKK}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K})\\ & +tr(\mathbf{W}_{t-1}\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K})+1 \end{aligned}$$

PROOF. We define the Lagrangian as:

$$\begin{aligned} f(\mathbf{W},\xi,\lambda,\tau) \quad = \quad & \frac{1}{2}\|\mathbf{W}-\mathbf{W}_{t-1}\|_F^2 + C_1 tr(\mathbf{KWKL}) + C_2\xi\\ & -\lambda\xi + \tau(1-\xi+\langle\mathbf{W},\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\rangle_F) \end{aligned}$$

where $\lambda\ge 0$ and $\tau\ge 0$ are Lagrangian multipliers. By setting $\frac{\partial f(\mathbf{W},\xi,\lambda,\tau)}{\partial\mathbf{W}}=0$, we have the following:

$$\mathbf{W}-\mathbf{W}_{t-1}+C_1\mathbf{KLK}+\tau\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}=0 \quad (16)$$

and therefore

$$\mathbf{W} = \mathbf{W}_{t-1} - C_1\mathbf{KLK} - \tau\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K} \quad (17)$$

Next by setting $\frac{\partial f(\mathbf{W},\xi,\lambda,\tau)}{\partial\xi}=0$, we have:

$$C_2-\lambda-\tau=0 \quad (18)$$

Since $\lambda\ge 0$, we have $\tau\le C_2$. Thus, we have the following:

$$\begin{aligned} & f(\tau)\\ = \quad & \frac{1}{2}\|\mathbf{W}-\mathbf{W}_{t-1}\|_F^2 + C_1 tr(\mathbf{KWKL}) + \tau\\ & +\tau\langle\mathbf{W},\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\rangle_F\\ = \quad & \frac{1}{2}\|\mathbf{W}-\mathbf{W}_{t-1}\|_F^2 - tr(\mathbf{W}(\mathbf{W}-\mathbf{W}_{t-1})) + \tau\\ = \quad & -\frac{1}{2}\|\mathbf{W}-\mathbf{W}_{t-1}\|_F^2 - tr(\mathbf{W}_{t-1}(\mathbf{W}-\mathbf{W}_{t-1})) + \tau\\ = \quad & -\frac{1}{2}\|C_1\mathbf{KLK}+\tau\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\|_F^2\\ & +C_1 tr(\mathbf{W}_{t-1}\mathbf{KLK}) + \tau tr(\mathbf{W}_{t-1}\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}) + \tau \end{aligned}$$

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

$$\begin{aligned} & \frac{\partial f(\tau)}{\partial\tau}\\ = \quad & -C_1 tr(\mathbf{KLKK}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}) - \tau\|\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\|_F^2\\ & +tr(\mathbf{W}_{t-1}\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}) + 1\\ = \quad & 0 \end{aligned}$$

Thus, we have

$$\tau = \frac{l_{t-1}}{\|\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\|_F^2} \quad (19)$$

where

$$\begin{aligned} l_{t-1} \quad = \quad & -C_1 tr(\mathbf{KLKK}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K})\\ & +tr(\mathbf{W}_{t-1}\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K})+1 \end{aligned}$$

Combining the fact that $\tau\le C_2$, we have

$$\tau = \min\{C_2, \frac{l_{t-1}}{\|\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\|_F^2}\} \quad (20)$$

Further by considering $\tau\ge 0$, we have:

$$\tau = \min\{C_2, \frac{h(l_{t-1})}{\|\mathbf{K}(\mathbf{E}_{ij}-\mathbf{E}_{ik})\mathbf{K}\|_F^2}\} \quad (21)$$

where $h(x)=\max\{0,x\}$ is the hinge loss function. □

4.3.2 Algorithm on Multiple Kernels

We now extend the above online distance learning on a single kernel to the setting of online multi-modal distance learning with multiple kernels. In particular, we tackle the challenges of the online multi-modal distance learning task by two steps: (i) we apply the previous online distance learning algorithm for learning distance function for each individual kernel; and (ii) we attempt to find the optimal combination of multiple kernels using the Hedging online learning algorithm by following the similar idea in [24]. Specifically, we denote by $\mu_p(t)$ the combination weight of the p-th kernel at t-th learning round. We iteratively update the following combination weight according to the online learning performance:

$$\mu_p(t) = \mu_p(t-1)\eta^{z_p(t)} \quad (22)$$

where $\eta\in(0,1)$ is a discounting parameter, and $z_p(t)$ is an indicator which outputs 1 when $d_p(\mathbf{x}_i,\mathbf{x}_j) > d_p(\mathbf{x}_i,\mathbf{x}_k)$, and 0 otherwise. The details of the proposed OMDL algorithm are shown in Algorithm 1.

Algorithm 1 Online Multi-modal Distance Learning (OMDL)

INPUT:

- n objects $\mathcal{X} = \{\mathbf{x}_i | i = 1, 2, \ldots, n\}$
- m kernel matrices $\mathbf{K}^p, p = 1, 2, \ldots, m$
- m graph Laplacian matrices $\mathbf{L}^p, p = 1, 2, \ldots, m$
- tradeoff parameter $C_1 > 0, C_2 > 0$
- discount weight $\eta \in (0, 1)$

Initialize: $\mathbf{W}_0^p = \mathbf{I}, \mu_p(0) = 1, p = 1, 2, \ldots, m$
for $t = 1, 2, \ldots, T$ **do**
  receive a triplet (i, j, k)
  for $p = 1, 2, \ldots, m$ **do**
    compute τ_t^p in (15).
    compute $\mathbf{W}_t^p$ in (14).
    PSD projection: $\lambda_i \mapsto \max(0, \lambda_i), i = 1, 2, \ldots, n$
    if $d_p(\mathbf{x}_i, \mathbf{x}_j) > d_p(\mathbf{x}_i, \mathbf{x}_k)$ **then**
      $z_p(t) = 1$
    else
      $z_p(t) = 0$
    end if
    update $\mu_p(t) = \mu_p(t-1)\eta^{z_p(t)}$
  end for
end for

OUTPUT: mapping $g : \mathcal{X} \to \mathbb{R}^{m \times n}$

4.4 Fast OMDL Algorithm

In this section, we propose a fast OMDL algorithm by low rank approximation (OMDL-LR). Specifically, instead of learning a high dimensional matrix $\mathbf{W}$ of $d \times d$, we attempt to learn a matrix $\mathbf{W}_{LR}$ of dimensionality $d_{LR} \times d_{LR}$ ($d_{LR} \ll d$). Following [5], we generate a $d \times d_{LR}$ random projection matrix $\mathbf{P}$ based on Gaussian distribution, and use $\mathbf{P}\mathbf{W}_{LR}\mathbf{P}^\top$ to approximate $\mathbf{W}$. Once the random projection matrix is chosen, it is straightforward for improving the rest of the algorithm by projecting the columns of kernel values accordingly, i.e., $\mathbf{K}_i \leftarrow \mathbf{P}^\top \mathbf{K}_i$. One may also consider other low-rank approximation methods, such as Nyström [44].

5. EXPERIMENTS

To evaluate our proposed scheme, we first test it on small-scale data sets. Then we apply it in image retrieval applications on some standard benchmark data sets. At last we report the results on a large-scale image retrieval data set.

5.1 Experimental Testbed

We form our experimental testbed[1] by adopting three publicly available image data sets. These three data sets have been widely used for the benchmark of image retrieval, classification and recognition tasks.

The first testbed is the "Caltech256" database[2], which has been widely adopted for object recognition and image retrieval tasks [12, 26, 9, 7]. This database contains 256 object categories (excluding the background category) and a total of 30607 images. Following the similar experiments as the previous work [7], we pick 10, 20 or 50 out of the 256 classes to form three subsets (the same sets as used in [7]), which are named as "Caltech10", "Caltech20", and "Caltech50", respectively.

[1] `http://www.cais.ntu.edu.sg/~chhoi/OMDL/`
[2] `http://www.vision.caltech.edu/Image_Datasets/Caltech256/`

The second testbed is the "Corel5000" database [18]. 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 third testbed is the "ImageCLEF" database[3], which is a medical image data set for image retrieval benchmark evaluation. To examine the scalability, we also combine "ImageCLEF" with a collection of 1,000,000 social images crawled from Flickr, which is named "ImageCLEF+". For the Flickr images, we treat all of them as the background noise, which are mainly used to test the scalability of our algorithms.

5.2 Experimental Setup

For each data set, 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.

We need to generate side information in the forms of triplet and pairwise training instances. In our approach, we generate side information by sampling triplet constraints from the feedback of images in the training set. The feedback information is simply derived from class labels in this study, which can be collected from the relevance feedback mechanism in a practical CBIR system [20]. Specifically, we select 40% from the training set to form the training queries and the rest as the database pool. For each training query, we generate 5 triplets. A triplet (q, p^+, p^-) is generated by first randomly choosing an image p^+ from the database pool which belongs to the same class as the training query q, then choosing another image p^- from the database pool which belongs to another class. As a result, two pairwise constraints, $(q, p^+, +1)$ and $(q, p^-, -1)$, can be derived from (q, p^+, p^-). For the small-scale experiment, the triplets for validation (test) are generated in the similar manner as that of the training set, but restrict q from validation (query) set, p^+ and p^- from the test set. The final results are averaged over 5 random sets of training data. We measure both mean and standard deviation of the results, and highlight the best case by performing student t-tests with the significance level $\alpha = 0.05$.

The performance of the small-scale experiment is evaluated by the prediction accuracy on triplets from the test set. For the image retrieval experiments, we evaluated the performance by standard performance metrics for multimedia retrieval. Specifically, for each query image, all the images in the database are ranked according to their similarities to the query. We can measure the precision at top n ranked 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) [31] 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

[3] http://imageclef.org/

Table 1: Experimental results on three small-scale data sets

Dataset	Metric	Kernel-Best	Kernel-U	Kernel-W	MKPOE	OMDL	OMDL-LR
Caltech(S)	accuracy	0.4700	0.6200	0.5900	0.6967	**0.7933**	**0.7900**
	std	±0.0000	±0.0000	±0.0000	±0.0681	**±0.0058**	**±0.0100**
	time(s)	-	-	-	150.5669	5.6428	**0.4364**
	std				±1.1985	±0.0793	**±0.0088**
Corel(S)	accuracy	0.6900	0.6200	0.6700	0.7167	**0.8067**	**0.7967**
	std	±0.0000	±0.0000	±0.0000	±0.0153	**±0.0351**	**±0.0306**
	time(s)	-	-	-	149.4588	5.6499	**0.4446**
	std				±1.3813	±0.3046	**±0.0262**
ImageCLEF(S)	accuracy	0.7700	0.7400	0.7500	0.8600	**0.9333**	**0.9333**
	std	±0.0000	±0.0000	±0.0000	±0.0300	**±0.0058**	**±0.0289**
	time(s)	-	-	-	152.1280	5.6533	**0.4364**
	std				±0.5545	±0.0050	**±0.0071**

examples over the total retrieved examples, while recall is the ratio of the relevant examples retrieved over the total relevant examples in the database.

Two parameters for the proposed OMDL algorithm are set as follows: (i) k — the number of nearest neighbors for graph Laplacian $\mathbf{L}$ is set to 5 for the small-scale experiment in Table 1 and 50 for the image retrieval applications in Table 2; (ii) d_{LR} — the dimensionality of the low-rank matrix $\mathbf{W}_{LR}$ for OMDL-LR is set to 20 for the small-scale experiment and 100 for the image retrieval application. Finally, all the experiments were running in MATLAB on a Linux machine with 3GHz Intel CPU and 16GB RAM.

5.3 Feature Descriptors and Kernel Functions

Here we describe feature descriptors for image representation and the set of kernel functions used.

5.3.1 Feature Descriptors for Image Representation

We adopt both global and local feature descriptors to extract features for representing images.

For global features, we extract five kinds of features from images, 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 adopted in previous CBIR studies.

For local features, we extract the bag-of-words features using two types of descriptors: (i) the SIFT descriptor, in which we adopt the Hessian-Affine interest region detector with threshold 500; and (ii) the SURF descriptor, in which we use 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-words for the local features. By choosing different descriptors (SIFT/SURF) and vocabulary sizes (200/1000), we totally extracted four kinds of local features: SIFT200, SIFT1000, SURF200 and SURF1000.

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.

5.3.2 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. Specifically, we adopt RBF kernel functions to build kernels on each kind of feature, which thus results in a total of 9 different kernels. RBF kernel is computed as $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-\frac{d(\mathbf{x},\mathbf{x}')}{\gamma})$, where $d(\cdot,\cdot)$ is the Euclidean distance, the kernel parameter γ is selected as the mean of the pairwise distance.

5.4 Experiment I: Learning Distance on Small-Scale Data Sets

In this section, we follow the same setting as [32]. We build 3 small-scale data sets from the above benchmark data sets by first choosing 10 object categories, and then randomly sampling 20 examples from each category. We name them "Caltech(S)", "Corel(S)" and "ImageCLEF(S)" respectively. We adopt 5 kernels built from global features described in section 5.3 augmented by a set of kernels with five "noise" kernels, each of which was generated by sampling random points from the unit sphere in $\mathbb{R}^3$ and applying the RBF kernel.

5.4.1 Comparison Algorithms

In this experiment, we implemented the following algorithms for comparison:

- Kernel-Best: we test the performance of all the kernels on the validation set by adopting kernel value as similarity, and then select the best kernel with the highest performance;
- Kernel-U: we build a new kernel as the uniform combination of all kinds of kernels;
- Kernel-W: we test the performance of all kinds of kernels on the validation set by adopting kernel value as similarity, and then build a kernel as the weighted combination of all kinds of kernels where the weight is computed as e^{mAP_p}, $p = 1, 2, \ldots, m$;
- MKPOE: the existing MKPOE algorithm in [32];
- OMDL: the proposed algorithm in Algorithm 1;
- OMDL-LR: the proposed fast OMDL algorithm in Section 4.4 which learns a low rank matrix by random projection to speed up OMDL.

5.4.2 Experimental Results

Table 1 summarizes the experimental results of the compared algorithms on three small-scale data sets. Some observations can be drawn from the results as follows.

First, we can see that both Kernel-U and Kernel-W beat Kernel-Best on "Caltech(S)", but fail on the other two data sets. This implies that kernel combination is possible to improve the performance, but if the combination weight is not assigned appropriately, it could even decrease the performance.

Second, we found that MKPOE significantly outperforms Kernel-Best, Kernel-U, and Kernel-W, which shows the importance of learning the multi-modal distance.

Third, we found that OMDL not only significantly improves the accuracy performance of MKPOE, but also spends much less time. On average, OMDL is about 30 times faster than MKPOE.

Finally, OMDL-LR can further improve the efficiency while maintain comparable performance with OMDL. On overage, OMDL-LR is about 450 times faster than MKPOE.

5.5 Experiment II: Learning Distance for Medium-scale Image Retrieval Applications

In this section, we examine the proposed algorithm on some medium-scale image retrieval data sets, which have been widely used for benchmarking.

5.5.1 Comparison Algorithms

To examine the efficacy of the proposed algorithms extensively, we have compared our technique with a variety of state-of-the-art algorithms as follows:

- Kernel-Best: we test the performance of all the kernels on the validation set by adopting kernel value as similarity, and then select the best kernel with the highest performance;
- RCA-Best: we train RCA [3] model on the training set for all the features and test each model on the validation set, and then select the best feature with the highest mAP for RCA;
- KRCA-Best: we train KRCA [40] model on the training set for all the kernels and test each model on the validation set, and then select the best kernel of the highest mAP for KRCA;
- LMNN-Best: we train LMNN [43] model on the training set for all the features and test each model on the validation set, and then select the best feature of the highest mAP for LMNN;
- OASIS-Best: we train OASIS [7] model on the training set for all the features and test each model on the validation set, and then select the best feature of the highest mAP for OASIS;
- Kernel-Con: we build a kernel on the concatenation of all kinds of features together;
- RCA-Con: we concatenate all kinds of features together first, and then run RCA;
- KRCA-Con: we concatenate all kinds of features together first, and the run KRCA;
- LMNN-Con: we concatenate all kinds of features together first, and then run LMNN;
- OASIS-Con: we concatenate all kinds of features together first, and then run OASIS;
- Kernel-U: we build a new kernel as the uniform combination of all kinds of kernels;
- Kernel-W: we test the performance of all kinds of kernels on the validation set by adopting kernel value as similarity, and then build a kernel as the weighted combination of all kinds of kernels where the weight is computed as e^{mAP_p}, $p = 1, 2, \ldots, m$;
- OMDL-LR: the proposed fast OMDL algorithm in Section 4.4 which learns a low rank matrix by random projection to speed up OMDL.

Remark. We note that we will only apply OMDL-LR to the rest experiments and exclude MKPOE and OMDL because of two key reasons: (i) MKPOE is not practical for large data sets due to its extremely low efficiency, which can be seen from the previous experiment; (ii) OMDL-LR is able to improve the efficiency of OMDL while maintaining comparable learning efficacy.

5.5.2 Experimental Results

Table 2 summarizes the experimental results of mAP performance of the compared algorithms on some standard benchmark data sets, and Figure 2 illustrates the details of the top-n precision results on one of the data sets. We can draw some observations from the results as follows.

First, by comparing Kernel-Best with Kernel-U and Kernel-W, it is obvious that kernel combination can lead to better performance, and the weighted combination outperforms uniform combination.

Second, we found that those methods based on concatenated feature outperform those based on best feature in some cases, such as "Corel5000" and "ImageCLEF" , but fail on the other data sets. This implies that feature concatenation is not optimal to combine different kinds of features.

Finally, by examining the results of OMDL-LR, we found that it outperforms the other algorithms on all the data sets. This promising result shows that the proposed OMDL algorithm is able to learn an effective distance function on multi-modal data.

5.6 Experiment III: Learning Distance for Large-scale Image Retrieval Applications

In this section, we apply the proposed algorithm on a large-scale image retrieval application to test its scalability. The data set we use is "ImageCLEF+", which is a combination of "ImageCLEF" and 1 million social photos crawled from Flickr. The compared algorithms are the same as those in the previous section. Experimental results of mAP performance are shown in the last column of Table 2. Figure 3 also illustrate the details of the top-n precision results.

Similar observations can be drawn from these results. Kernel combination (Kernel-U and Kernel-W) can lead to better performance than Kernel-Best, and the weighted combination outperforms uniform combination. Those methods based on concatenated feature outperform those based on best feature on this data set. And it is obvious that OMDL-LR outperforms all the other algorithms. The promising results show that the proposed algorithm can learn an effective distance function with multi-modal data for large-scale image retrieval applications.

Table 2: Experimental results of distance learning for image retrieval applications.

Algorithm	Metric	Caltech10	Caltech20	Caltech50	Corel5000	ImageCLEF	ImageCLEF+
Kernel-Best	mAP	0.2315	0.1657	0.1010	0.1789	0.4090	0.0959
	std	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000
RCA-Best	mAP	0.2343	0.1714	0.1012	0.1801	0.4709	0.1098
	std	±0.0003	±0.0001	±0.0001	±0.0001	±0.0004	±0.0002
KRCA-Best	mAP	0.2489	0.1867	0.1087	0.2113	0.5497	0.1078
	std	±0.0004	±0.0003	±0.0001	±0.0002	±0.0012	±0.0008
LMNN-Best	mAP	0.2365	0.1696	0.1049	0.1909	0.4939	0.1069
	std	±0.0013	±0.0014	±0.0002	±0.0002	±0.0013	±0.0019
OASIS-Best	mAP	0.2472	0.1852	0.1010	0.1797	0.4325	0.1062
	std	±0.0036	±0.0052	±0.0000	±0.0054	±0.0087	±0.0115
Kernel-Con	mAP	0.2115	0.1609	0.0998	0.2518	0.3959	0.1598
	std	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000
RCA-Con	mAP	0.2173	0.1699	0.1040	0.2591	0.5044	0.2176
	std	±0.0001	±0.0002	±0.0000	±0.0002	±0.0006	±0.0009
KRCA-Con	mAP	0.2404	0.1955	0.1114	0.3240	0.5797	0.1936
	std	±0.0008	±0.0001	±0.0001	±0.0002	±0.0012	±0.0005
LMNN-Con	mAP	0.2419	0.1781	0.1097	0.2768	0.5373	0.2272
	std	±0.0009	±0.0013	±0.0002	±0.0005	±0.0021	±0.0030
OASIS-Con	mAP	0.2350	0.1633	0.1001	0.2518	0.4518	0.1560
	std	±0.0044	±0.0043	±0.0004	±0.0077	±0.0119	±0.0047
Kernel-U	mAP	0.2536	0.2133	0.1247	0.3310	0.4415	0.2554
	std	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000
Kernel-W	mAP	0.2557	0.2148	0.1253	0.3321	0.4602	0.2682
	std	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000	±0.0000
OMDL-LR	mAP	**0.3377**	**0.2498**	**0.1356**	**0.3639**	**0.6136**	**0.3147**
	std	**±0.0047**	**±0.0025**	**±0.0004**	**±0.0068**	**±0.0098**	**±0.0220**

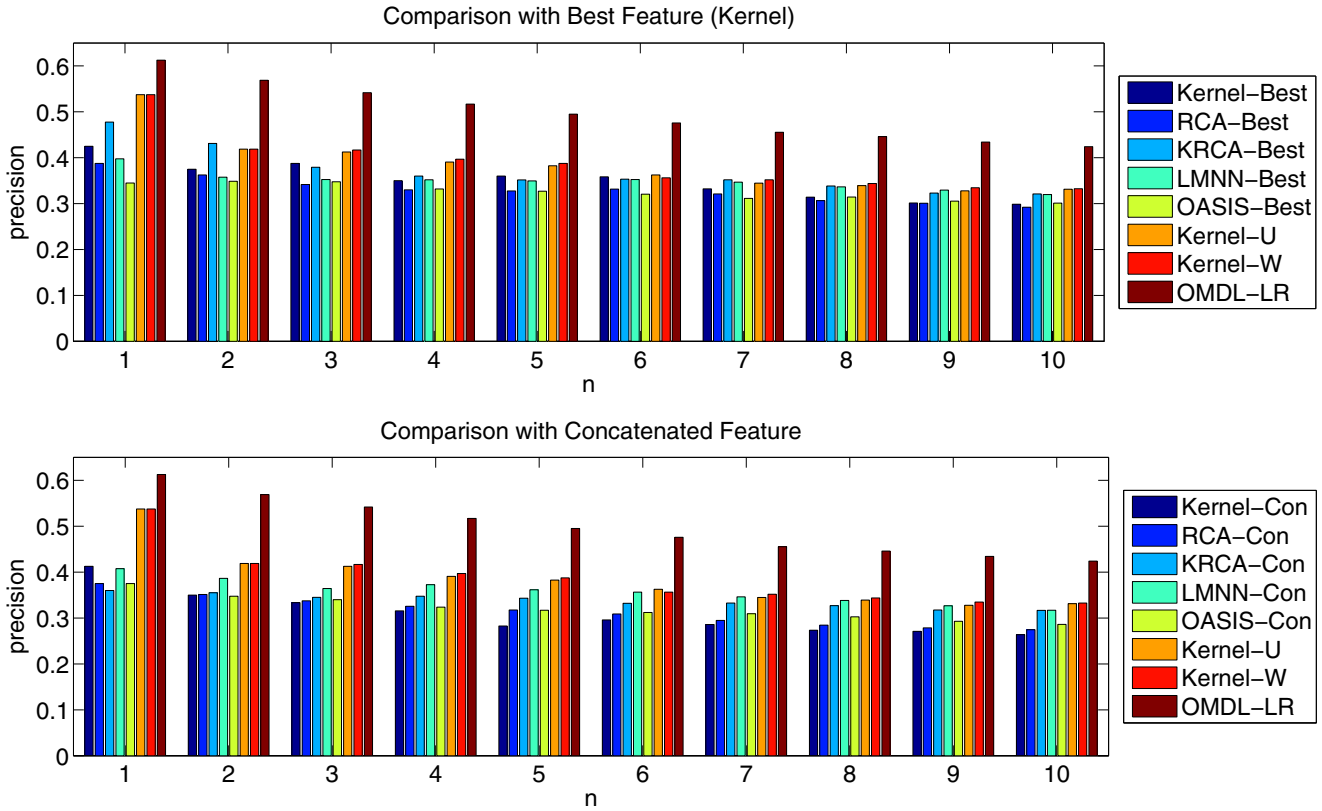

Figure 2: Top-n precision of retrieval results on "Caltech10" data set.

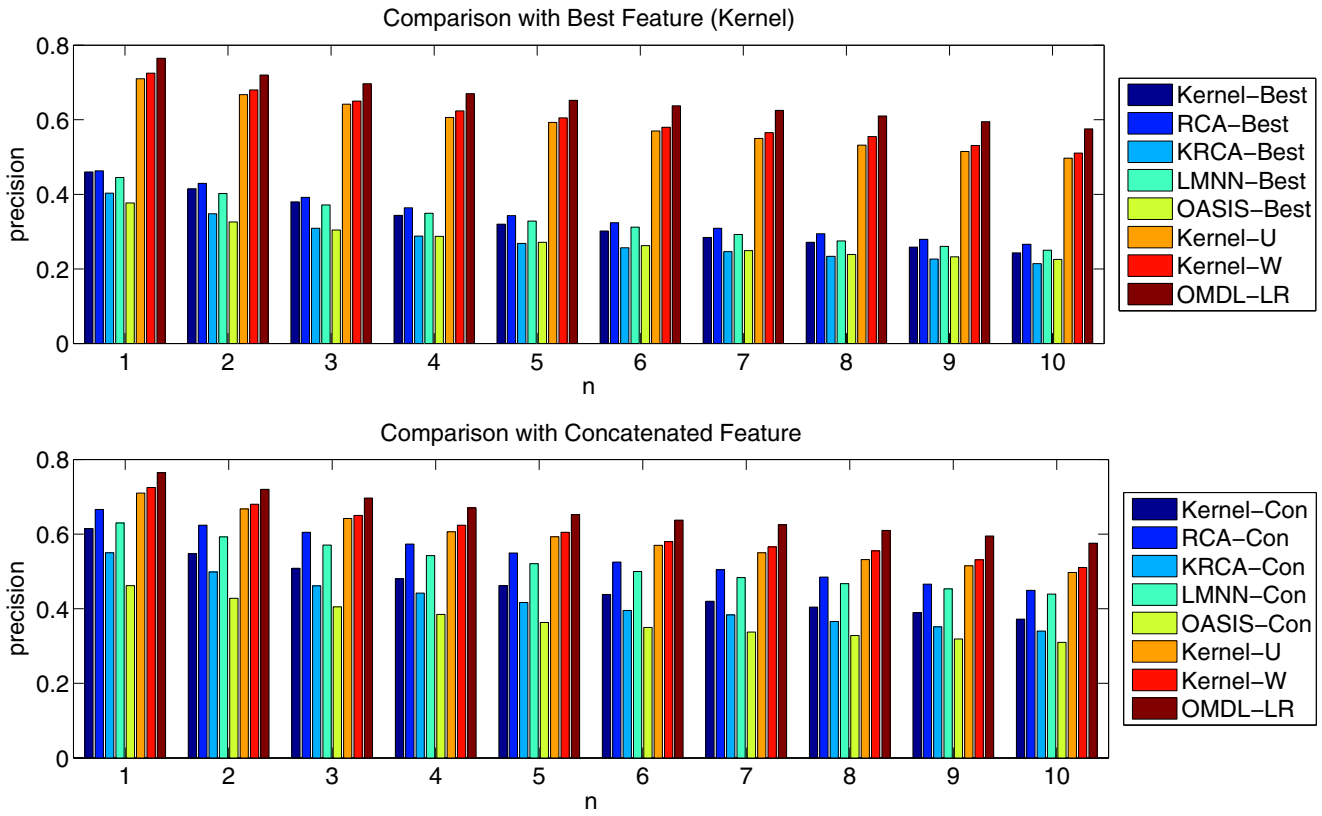

Figure 3: Top-n precision of retrieval results on "ImageCLEF+" data set.

6. CONCLUSIONS

This paper addressed an important challenge of learning proximity functions on multi-modal data for multimedia retrieval. Unlike the previous Multiple Kernel Partial Order Embedding (MKPOE) which suffers from poor efficiency and scalability, this paper proposed a novel framework of Online Multi-modal Distance Learning (OMDL), which on one hand enhances the learning efficacy of MKPOE by exploiting underlying data distribution via the graph Laplacian, and on the other hand significantly improves the efficiency and scalability via an online learning scheme. In this paper, we developed efficient online learning algorithms and extensively evaluated the proposed OMDL algorithms on several public image data sets. The empirical results showed that the proposed OMDL algorithms are not only more effective than the MKPOE technique, but also significantly more efficient and scalable for large-scale applications. Future work will explore theoretical analysis of our technique.

Acknowledgments

This work was in part supported by Singapore MOE Academic tier-1 grant (RG33/11) and Microsoft Research grant.

7. REFERENCES

[1] 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*, pages 2357–2362, 2000.

[2] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[3] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning*, pages 11–18, 2003.

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

[5] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *KDD*, pages 245–250, 2001.

[6] H. Chang and D.-Y. Yeung. Kernel-based distance metric learning for content-based image retrieval. *Image Vision Comput.*, 25(5):695–703, 2007.

[7] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.

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

[9] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009.

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

[11] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1371–1384, 2008.

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[13] G. Guo, A. K. Jain, W.-Y. Ma, and H.-J. Zhang. Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Transactions on Neural Networks*, 13(4):811–820, 2002.

[14] S. C. H. Hoi and R. Jin. Active kernel learning. In *Proceedings of International Conference on Machine Learning*, pages 400–407, 2008.

[15] 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*, pages 361–368, 2007.

[16] S. C. H. Hoi, R. Jin, P. Zhao, and T. Yang. Online multiple kernel classification. *Machine Learning*, 2012. to appear.

[17] S. C. H. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Trans. Multimedia Comput. Commun. Appl.*, 6(3):18:1–18:26, Aug. 2010.

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

[19] S. C. H. Hoi and M. R. Lyu. A multimodal and multilevel ranking scheme for large-scale video retrieval. *IEEE Transactions on Multimedia*, 10(4):607–619, June 2008.

[20] S. C. H. Hoi, M. R. Lyu, and R. Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Trans. KDE*, 18(4):509–204, 2006.

[21] 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.

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

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

[24] R. Jin, S. C. H. Hoi, and T. Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *ALT*, pages 390–404, 2010.

[25] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1877–1890, 2008.

[26] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In *ICCV*, pages 1–8, 2007.

[27] 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*, 5:27–72, 2004.

[28] 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.

[29] 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.*, 2(1):1–19, 2006.

[30] B. Li, E. Y. Chang, and Y.-L. Wu. Discovery of a perceptual distance function for measuring image similarity. *Multimedia Syst.*, 8(6):512–522, 2003.

[31] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[32] B. McFee and G. R. G. Lanckriet. Learning multi-modal similarity. *Journal of Machine Learning Research*, 12:491–523, 2011.

[33] A. Qamra, Y. Meng, and E. Y. Chang. Enhanced perceptual distance functions and indexing for image replica recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):379–391, 2005.

[34] 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.*, 30(11):1902–1912, 2008.

[35] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *COLT/EuroCOLT*, pages 416–426, 2001.

[36] L. Si, R. Jin, S. C. H. Hoi, and M. R. Lyu. Collaborative image retrieval via regularized metric learning. *ACM Multimedia Systems Journal*, 12(1):34–44, 2006.

[37] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831, 2005.

[38] 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*, 22(12):1349–1380, 2000.

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

[40] 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)*, pages 954–959, 2005.

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

[42] S. V. N. Vishwanathan, Z. sun, N. Ampornpunt, and M. Varma. Multiple kernel learning and the smo algorithm. In *NIPS*, pages 2361–2369, 2010.

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

[44] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[45] 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*, pages 135–144, 2009.

[46] L. Wu, S. C. H. Hoi, and N. Yu. Semantics-preserving bag-of-words models and applications. *Trans. Img. Proc.*, 19(7):1908–1920, July 2010.

[47] P. Wu, S. C. H. Hoi, P. Zhao, and Y. He. Mining social images with distance metric learning for automated image tagging. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 197–206, Hong Kong, China, 2011. ACM.

[48] 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.

[49] 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.

[50] 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.*, 32(1):30–44, 2010.

[51] D.-Y. Yeung and H. Chang. A kernel approach for semisupervised metric learning. *IEEE Transactions on Neural Networks*, 18(1):141–149, 2007.

[52] J. Zhuang and S. C. H. Hoi. A two-view learning approach for image tag ranking. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM'11, pages 625–634, Hong Kong, China, 2011.

[53] J. Zhuang, I. W. Tsang, and S. C. H. Hoi. A family of simple non-parametric kernel learning algorithms. *Journal of Machine Learning Research*, 12:1313–1347, 2011.

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