Learning a Maximum Margin Subspace for Image Retrieval

Xiaofei He, Deng Cai, *Student Member*, *IEEE*, and Jiawei Han, *Senior Member*, *IEEE*

Abstract—One of the fundamental problems in Content-Based Image Retrieval (CBIR) has been the gap between low-level visual features and high-level semantic concepts. To narrow down this gap, relevance feedback is introduced into image retrieval. With the user-provided information, a classifier can be learned to distinguish between positive and negative examples. However, in real-world applications, the number of user feedbacks is usually too small compared to the dimensionality of the image space. In order to cope with the high dimensionality, we propose a novel semisupervised method for dimensionality reduction called **Maximum Margin Projection** (MMP). MMP aims at maximizing the margin between positive and negative examples at each local neighborhood. Different from traditional dimensionality reduction algorithms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), which effectively see only the global euclidean structure, MMP is designed for discovering the local manifold structure. Therefore, MMP is likely to be more suitable for image retrieval, where nearest neighbor search is usually involved. After projecting the images into a lower dimensional subspace, the relevant images get closer to the query image; thus, the retrieval performance can be enhanced. The experimental results on Corel image database demonstrate the effectiveness of our proposed algorithm.

Index Terms—Multimedia information systems, image retrieval, relevance feedback, dimensionality reduction.

1 Introduction

Content-based Image Retrieval (CBIR) has attracted substantial interests in the last decade [4], [7], [12], [15], [24], [28], [31], [34], [35]. It is motivated by the fast growth of digital image databases, which, in turn, require efficient search schemes. Rather than describing an image by using text, in these systems, an image query is described using one or more example images. The low-level visual features (color, texture, shape, etc.,) are automatically extracted to represent the images. However, the low-level features may not accurately characterize the high-level semantic concepts. To narrow down this semantic gap, relevance feedback is introduced into CBIR [31]. With the user-provided negative and positive feedbacks, image retrieval can then be thought of as a classification problem [39], [40].

In real-world image retrieval systems, the relevance feedbacks provided by the user is often limited, typically less than 20, whereas the dimensionality of the image space can range from several hundreds to thousands. One of the crucial problems encountered in applying statistical techniques to image retrieval has been called the "*curse of dimensionality*." Procedures that are analytically or computationally manageable in low-dimensional spaces can become completely impractical in a space of several hundreds or thousands dimensions [14]. Thus, various techniques have been developed for reducing the dimensionality of the feature space, in the hope of obtaining a more manageable problem. The most popular dimensionality reduction algorithms include Principal Component Analysis (PCA) [14], [36] and Linear Discriminant Analysis (LDA) [14], [37]. PCA projects the data points into a lower dimensional subspace, in which the sample variance is maximized. It computes the eigenvectors of the sample covariance matrix and approximates the original data by a linear combination of the leading eigenvectors. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. Unlike PCA, which is unsupervised, LDA is a supervised dimensionality reduction algorithm. LDA encodes discriminatory information by finding directions that maximize the ratio of between-class scatter to within-class scatter. Both PCA and LDA have widely been applied to image retrieval, face recognition, information retrieval, and pattern recognition. However, they are designed for discovering only the *global* euclidean structure, whereas the local manifold structure is ignored. The global statistics such as variance is often difficult to estimate when there are no sufficient samples.

Recently, various researchers (see [30], [38], [1]) have considered the case when the data lives on or close to a submanifold of the ambient space. One then hopes to estimate the geometrical and discriminant properties of the submanifold from random points lying on this unknown submanifold. All of these approaches try to discover the intrinsic manifold structure. However, these methods are nonlinear and computationally expensive. In addition, they are defined only on the training data points, and it is unclear how the map can be evaluated for new test points. Therefore, they are not suitable for image retrieval. In [18], a ranking scheme on manifold is proposed for image retrieval. Its goal is to rank the images in the database with respect to the intrinsic global manifold structure. It has been shown to be superior to those approaches based on euclidean structure. He et al. [19] applied Locality Preserving Projections (LPP) [22] to find a linear approximation of

- *X. He is with Yahoo! Inc., 3333 West Empire Avenue, Burbank, CA 91504. E-mail: hex@yahoo-inc.com.*
- *D. Cai and J. Han are with the Department of Computer Science, University of Illinois at Urbana Champaign, 1334 Siebel Center, 201 N. Goodwin Ave., Urbana, IL 61801. E-mail: {dengcai2, hanj}@cs.uiuc.edu.*

Manuscript received 13 Nov. 2006; revised 26 Mar. 2007; accepted 17 Sept. 2007; published online 24 Sept. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0518-1106.
Digital Object Identifier no. 10.1109/TKDE.2007.190692.

the intrinsic data manifold. Image retrieval is then performed in the reduced subspace by using a euclidean metric. Most recently, a novel image retrieval method called *Augmented Relation Embedding* (ARE) [27] is proposed to learn a semantic manifold, which respects the user's preferences. Specifically, ARE constructs three relational graphs: one describes the similarity relations, and the other two encode relevant/irrelevant relations by using user-provided relevance feedbacks. With the relational graphs thus defined, learning a semantic manifold can be transformed into solving a constrained optimization problem.

In this paper, we propose a novel method, called **Maximum Margin Projection** (MMP), which focuses on local discriminant analysis for image retrieval. Its goal is to discover both geometrical and discriminant structures of the data manifold. In image retrieval, the relevance feedbacks provided by the user is often limited. Consequently, it is difficult to accurately estimate the *global* geometrical structure of the data manifold such as geodesics. Instead, one then hopes to estimate the local statistics such as the local covariance matrix [20], tangent space [44], etc. In our algorithm, we first construct a nearest neighbor graph to model the local geometrical structure of the underlying manifold. This graph is then split into a *within-class graph* and a *between-class graph* by using class information and neighborhood information. The within-class graph connects two data points if they share the same label or they are sufficiently close to each other, whereas the between-class graph connects data points having different labels. This way, both of the local geometrical and discriminant structures of the data manifold can accurately be characterized by these two graphs. Using the notion of graph Laplacian [10], we can then find a linear transformation matrix that maps the images to a subspace. At each local neighborhood, the margin between relevant and irrelevant images is maximized. Therefore, this linear transformation optimally preserves the local neighborhood information and the discriminant information.

This paper is structured as follows: In Section 2, we provide a brief review of manifold learning techniques. The MMP algorithm is introduced in Section 3. In Section 4, we describe how our MMP algorithm can be applied to relevance feedback image retrieval. The experimental results are presented in Section 5. Finally, we provide some concluding remarks and suggestions for future work in Section 6.

2 Manifold Learning Techniques

Since our algorithm is fundamentally based on manifold learning techniques, in this section, we provide a brief review of Locally Linear Embedding (LLE) [30], Isomap [38], and Laplacian Eigenmaps [1], which are three of the most popular manifold learning techniques. Let $\mathbf{x}_1, \cdots, \mathbf{x}_m$ be the data points sampled from an underlying submanifold $\mathcal{M}$ embedded in $\mathbb{R}^n$ and let y_i be the one-dimensional map of $\mathbf{x}_i$, $i = 1, \cdots, m$.

2.1 Locally Linear Embedding

The basic idea of LLE is that the data points might reside on a nonlinear submanifold, but it might be reasonable to assume that each local neighborhood is linear. Thus, we can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Specifically, we first construct a k nearest neighbor graph G with weight matrix W. Reconstructing errors are measured by the following cost function [30]:

$$\phi(W) = \sum_{i=1}^{m} \left\| \mathbf{x}_i - \sum_{j=1}^{m} W_{ij}\mathbf{x}_j \right\|^2,$$

which adds up the squared distances between all the data points and their reconstructions. Note that W_{ij} vanishes for distant data points. See [30] for finding a W that minimizes $\phi(W)$. Consider the problem of mapping the original data points to a line so that each data point on the line can be represented as a linear combination of its neighbors with the coefficients W_{ij}. Let $\mathbf{y} = (y_1, y_2, \cdots, y_m)^T$ be such a map. A reasonable criterion for choosing a "good" map is to minimize the following cost function [30]:

$$\Phi(\mathbf{y}) = \sum_{i=1}^{m} \Big(y_i - \sum_{j=1}^{m} W_{ij} y_j \Big)^2.$$

This cost function, like the previous one, is based on locally linear reconstruction errors, but here, we fix the weights W_{ij} while optimizing the coordinates y_i. It can be shown that the optimal embedding $\mathbf{y}$ is given by the minimum eigenvalue solution to the following eigenvalue problem:

$$(I - W)^T (I - W)\mathbf{y} = \lambda \mathbf{y},$$

where I is an $m \times m$ identity matrix.

2.2 Isomap

Let $d_{\mathcal{M}}$ be the geodesic distance measure on $\mathcal{M}$ and let d be the standard euclidean distance measure in $\mathbb{R}^n$. Isomap aims at finding a euclidean embedding such that euclidean distances in $\mathbb{R}^n$ can provide a good approximation to the geodesic distances on $\mathcal{M}$. That is,

$$f^{opt} = \arg\min_{f} \sum_{i,j} \big(d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) - d\big(f(\mathbf{x}_i), f(\mathbf{x}_j)\big)\big)^2. \quad (1)$$

In real-life data sets, the underlying manifold $\mathcal{M}$ is often unknown and, hence, the geodesic distance measure is also unknown. In order to discover the intrinsic geometrical structure of $\mathcal{M}$, we first construct a k nearest neighbor graph G over all data points to model the local geometry. Once the graph is constructed, the geodesic distances $d_{\mathcal{M}}(i, j)$ between all pairs of points on the manifold $\mathcal{M}$ can be estimated by computing their shortest path distances $d_G(i, j)$ on the graph G. The procedure is given as follows: Initialize $d_G(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by an edge; otherwise, $d_G(\mathbf{x}_i, \mathbf{x}_j) = \infty$. Then, for each value of $l = 1, 2, \cdots, m$, in turn, replace all entries $d_G(\mathbf{x}_i, \mathbf{x}_j)$ by $\min\{d_G(\mathbf{x}_i, \mathbf{x}_j), d_G(\mathbf{x}_i, \mathbf{x}_l) + d_G(\mathbf{x}_l, \mathbf{x}_j)\}$. The matrix of final values $D_G = \{d_G(\mathbf{x}_i, \mathbf{x}_j)\}$ will contain the shortest path distances between all pairs of points in G. This procedure is named the Floyd-Warshall algorithm [11]. More efficient algorithms that exploit the sparse structure of the neighborhood graph can be found in [16]. Let D_Y denote the matrix of euclidean distances in the reduced subspace, that is,

$\{d_Y(i,j) = \|y_i - y_j\|\}$. Thus, Isomap aims at minimizing the following cost function:

$$\|\tau(D_G) - \tau(D_Y)\|_{L^2},$$

where the τ operator converts distances to inner products, which uniquely characterize the geometry of the data in a form that supports efficient optimization [38]. Specifically, $\tau(D) = -HSH/2$, where $S_{ij} = D_{ij}^2$, and $H = I - \frac{1}{m}\mathbf{e}\mathbf{e}^T$, $\mathbf{e} = (1, 1, \cdots, 1)^T$. It can be shown that the optimal embedding $\mathbf{y} = (y_1, \cdots, y_m)$ is given by the top eigenvector of the matrix $\tau(D_G)$.

2.3 Laplacian Eigenmap

The Laplacian Eigenmap is based on spectral graph theory [10]. Given a k nearest neighbor graph G with weight matrix W, the optimal maps can be obtained by solving the following minimization problem:

$$\min_{\mathbf{y}} \sum_{i,j=1}^{m} (y_i - y_j)^2 W_{ij} = \mathbf{y}^T L \mathbf{y},$$

where $L = D - W$ is the *graph Laplacian* [10], and $D_{ii} = \sum_j W_{ij}$. The objective function with our choice of weights W_{ij} incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close," then y_i and y_j are close as well. The weight matrix W can be defined as follows:

$$W_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ is among the } k\text{-nearest neighbors of } \mathbf{x}_j \\ & \text{or } \mathbf{x}_j \text{ is among the } k\text{-nearest neighbors of } \mathbf{x}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The optimal embedding $\mathbf{y}$ is given by the minimum eigenvalue solution of the following generalized eigenvalue problem:

$$L\mathbf{y} = \lambda D\mathbf{y}. \tag{3}$$

All the above-mentioned manifold learning algorithms are nonlinear and computationally expensive. In addition, they are defined only on the training data points and therefore cannot directly be applied to supervised learning problems. To overcome this limitation, some methods for out-of-sample extension have been proposed [3], [33]. Bengio et al. proposed a unified framework for extending LLE, Isomap, and Laplacian Eigenmap [3]. This framework is based on seeing these algorithms as learning eigenfunctions of a data-dependent kernel. The Nyström formula is used to obtain an embedding for a new data point. Sindhwani et al. proposed a semisupervised learning algorithm, which constructs a family of data-dependent norms on Reproducing Kernel Hilbert Spaces (RKHS) [33]. Explicit formulas are derived for the corresponding new kernels. The kernels thus support out-of-sample extension.

2.4 Linear Techniques

Recently, several linear manifold learning algorithms have been proposed and applied to image retrieval such as LPP [19] and ARE [27]. In particular, the ARE approach has been shown to be superior to LPP [27]. In addition, it would be interesting to note that both ARE and LPP are fundamentally based on Laplacian Eigenmaps and can be thought of as its linear variants.

Out of the state-of-the-art linear manifold learning algorithms, ARE is the most relevant to our algorithm. ARE performs the relevance feedback image retrieval by using three graphs, that is, the similarity relational graph on the whole image database and two feedback relational graphs, which incorporate the user-provided positive and negative examples. The objective function of ARE is given as follows:

$$\begin{aligned} &Maximize \quad J(V) = \sum_{i,j} \|V^T\mathbf{x}_i - V^T\mathbf{x}_j\|^2 \left(W_{ij}^N - \gamma W_{ij}^P\right) \\ &subject\ to \quad \sum_{i,j} \|V^T\mathbf{x}_i - V^T\mathbf{x}_j\|^2 W_{ij}^S = 1, \end{aligned} \tag{4}$$

where V is the transformation matrix. The matrix W^N describes the positively similar relations, and W^P describes the dissimilar relations. W^S is the weight matrix of the nearest neighbor graph constructed over all the data points. See [27] for the details.

In the following, we list the similarities and major differences between ARE and our algorithm:

1. Both ARE and our algorithm are graph-based approaches for learning a linear approximation to the intrinsic data manifold. Both of them make use of both labeled and unlabeled data. Moreover, both of them can be obtained by solving a generalized eigenvector problem.
2. In the ARE algorithm, the labeled and unlabeled images are considered equally important during the course of finding the optimal projection. In fact, even though we assign different weights to the labeled and unlabeled images, the eigenvector solution to ARE remains the same. However, it might be more reasonable for the algorithm to differentiate labeled and unlabeled images by assigning higher weights to the labeled images, especially when only limited labeled images are available, whereas the unlabeled images are abundant. Our algorithm overcomes this problem by formulating two different objective functions.

Some other linear manifold learning techniques can be found in [5], [8], [21].

3 Maximum Margin Projection

In this section, we introduce our *MMP* algorithm, which respects both discriminant and geometrical structures in the data. We begin with a description of the linear dimensionality reduction problem.

3.1 The Linear Dimensionality Reduction Problem

The generic problem of linear dimensionality reduction is explained as follows: Given a set $\mathbf{x}_1, \cdots, \mathbf{x}_m$ in $\mathbb{R}^n$, find a transformation matrix $A = (\mathbf{a}_1, \cdots, \mathbf{a}_d)$ that maps these m points to a set of points $\mathbf{y}_1, \cdots, \mathbf{y}_m$ in $\mathbb{R}^d$ $(d \ll n)$ such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$, where $\mathbf{y}_i = A^T\mathbf{x}_i$.

3.2 The Maximum-Margin Objective Function for Dimensionality Reduction

As we have previously described, naturally occurring data may be generated by structured systems with possibly much fewer degrees of freedom than what the ambient dimension would suggest. Thus, we consider the case when the data lives on or close to a submanifold of the ambient space. In this paper, we consider the particular question of maximizing a *local* margin between relevant and irrelevant images.

Given m data points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\} \subset \mathbb{R}^n$ sampled from the underlying submanifold $\mathcal{M}$, suppose that the first l points are labeled, and the rest $m-l$ points are unlabeled. In image retrieval, the labeled images include the original query image and the images with user's relevance feedback. The problem of image retrieval concerns ranking the unlabeled images according to their relevance to the original query image. In order to model the local geometrical structure of $\mathcal{M}$, we first construct a nearest neighbor graph G. For each data point $\mathbf{x}_i$, we find its k nearest neighbors and put an edge between $\mathbf{x}_i$ and its neighbors. Let $N(\mathbf{x}_i) = \{\mathbf{x}_i^1, \cdots, \mathbf{x}_i^k\}$ be the set of its k nearest neighbors. Thus, the weight matrix of G can be defined as follows:

$$W_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The nearest neighbor graph G with weight matrix W characterizes the local geometry of the data manifold. It has frequently been used in manifold-based learning techniques such as [1], [38], [30], [22]. However, this graph fails to discover the discriminant structure in the data.

In order to discover both geometrical and discriminant structures of the data manifold, we construct two graphs, that is, *within-class graph* G_w and *between-class graph* G_b. Let $l(\mathbf{x}_i)$ be the class label of $\mathbf{x}_i$, either relevant or not. For each data point $\mathbf{x}_i$, the set $N(\mathbf{x}_i)$ can naturally be split into two subsets: $N_b(\mathbf{x}_i)$ and $N_w(\mathbf{x}_i)$. $N_b(\mathbf{x}_i)$ contains the neighbors having different labels, and $N_w(\mathbf{x}_i)$ contains the rest of the neighbors. Note that some of the images in N_w may not have labels. However, there is reason to suspect that these images are likely to be related to $\mathbf{x}_i$ if they are sufficiently close to $\mathbf{x}_i$. Specifically,

$$\begin{aligned} N_b(\mathbf{x}_i) &= \{\mathbf{x}_i^j | l(\mathbf{x}_i^j) \neq l(\mathbf{x}_i), 1 \leq j \leq k\}, \\ N_w(\mathbf{x}_i) &= N(\mathbf{x}_i) - N_b(\mathbf{x}_i). \end{aligned}$$

Clearly, $N_b(\mathbf{x}_i) \cap N_w(\mathbf{x}_i) = \emptyset$, and $N_b(\mathbf{x}_i) \cup N_w(\mathbf{x}_i) = N(\mathbf{x}_i)$. Let W_w and W_b be the weight matrices of G_w and G_b, respectively. We define the following:

$$W_{b,ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_b(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_b(\mathbf{x}_i), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

$$W_{w,ij} = \begin{cases} \gamma, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same label,} \\ 1, & \text{if } \mathbf{x}_i \text{ or } \mathbf{x}_j \text{ is unlabeled} \\ & \text{but } \mathbf{x}_i \in N_w(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_w(\mathbf{x}_i), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

When two images share the same label, it is with high confidence that they share the same semantics. Therefore, the value of γ should relatively be large. In our experiments, γ is empirically set to be 50, and the number of nearest neighbors k is set to be 5.

Now, consider the problem of mapping the within-class graph and between-class graph to a line so that connected points of G_w stay as close together as possible, whereas connected points of G_b stay as distant as possible. Let $\mathbf{y} = (y_1, y_2, \cdots, y_m)^T$ be such a map. A reasonable criterion for choosing a "good" map is to optimize two objective functions

$$\min \sum_{ij} (y_i - y_j)^2 W_{w,ij}, \tag{8}$$

$$\max \sum_{ij} (y_i - y_j)^2 W_{b,ij}, \tag{9}$$

under appropriate constraints. The objective function (8) on the within-class graph incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart, whereas they are actually in the same class. Likewise, the objective function (9) on the between-class graph incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped close together, whereas they actually belong to different classes. Therefore, minimizing (8) is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close and share the same label, then y_i and y_j are close as well. In addition, maximizing (9) is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close but have different labels, then y_i and y_j are far apart. The learning procedure is illustrated in Fig. 1.

3.3 Optimal Linear Embedding

In this section, we describe our MMP algorithm, which solves the objective functions (8) and (9). Let $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m)$. Suppose $\mathbf{a}$ is a projection vector, that is, $\mathbf{y}^T = \mathbf{a}^T X$, where $X = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$ is a $n \times m$ matrix. Following some simple algebraic steps, the objective function (8) can be reduced to

$$\begin{aligned} & \frac{1}{2} \sum_{ij} (y_i - y_j)^2 W_{w,ij} \\ = \; & \frac{1}{2} \sum_{ij} \left(\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j\right)^2 W_{w,ij} \\ = \; & \sum_i \mathbf{a}^T \mathbf{x}_i D_{w,ii} \mathbf{x}_i^T \mathbf{a} - \sum_{ij} \mathbf{a}^T \mathbf{x}_i W_{w,ij} \mathbf{x}_j^T \mathbf{a} \\ = \; & \mathbf{a}^T X D_w X^T \mathbf{a} - \mathbf{a}^T X W_w X^T \mathbf{a}, \end{aligned}$$

where D_w is a diagonal matrix, and its entries are column (or row, since W_w is symmetric) sum of W_w, $D_{w,ii} = \sum_j W_{w,ij}$. Similarly, the objective function (9) can be reduced to

$$\begin{aligned} & \frac{1}{2} \sum_{ij} (y_i - y_j)^2 W_{b,ij} \\ = \; & \frac{1}{2} \sum_{ij} \left(\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j\right)^2 W_{b,ij} \\ = \; & \mathbf{a}^T X (D_b - W_b) X^T \mathbf{a} \\ = \; & \mathbf{a}^T X L_b X^T \mathbf{a}, \end{aligned}$$

where D_b is a diagonal matrix. Its entries are column (or row, since W_b is symmetric) sum of W_b, $D_{b,ii} = \sum_j W_{b,ij}$. $L_b = D_b - W_b$ is the Laplacian matrix of G_b.

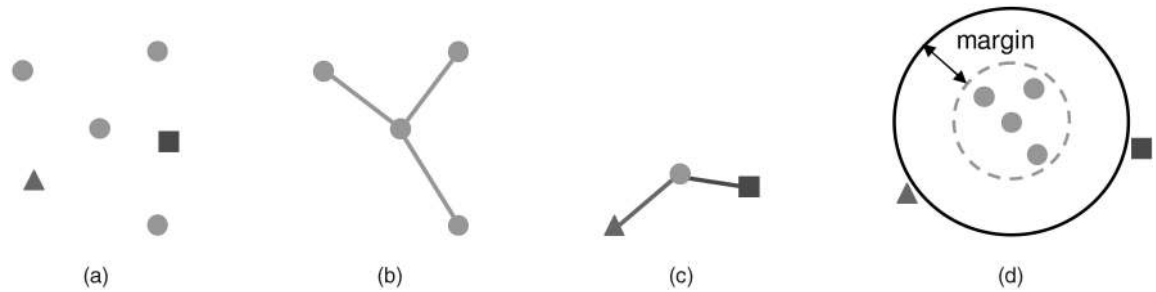

Fig. 1. (a) The center point has five neighbors. The points with the same shape belong to the same class. (b) The *within-class graph* connects nearby points with the same label. (c) The *between-class graph* connects nearby points with different labels. (d) After MMP, the margin between different classes is maximized.

Note that the matrix D_w provides a natural measure on the data points. If $D_{w,ii}$ is large, then it implies that the class containing $\mathbf{x}_i$ has a high density around $\mathbf{x}_i$. Therefore, the bigger the value of $D_{w,ii}$ is, the more "important" $\mathbf{x}_i$ becomes. Therefore, we impose a constraint as follows:

$$\mathbf{y}^T D_w \mathbf{y} = 1 \Rightarrow \mathbf{a}^T X D_w X^T \mathbf{a} = 1.$$

Thus, the objective function (8) becomes the following:

$$\min_{\mathbf{a}} 1 - \mathbf{a}^T X W_w X^T \mathbf{a}. \tag{10}$$

Equivalently,

$$\max_{\mathbf{a}} \mathbf{a}^T X W_w X^T \mathbf{a}. \tag{11}$$

In addition, the objective function (9) can be rewritten as follows:

$$\max_{\mathbf{a}} \mathbf{a}^T X L_b X^T \mathbf{a}. \tag{12}$$

Finally, the optimization problem reduces to finding

$$\arg \max_{\mathbf{a}^T X D_w X^T \mathbf{a}=1} \mathbf{a}^T X(\alpha L_b + (1-\alpha) W_w) X^T \mathbf{a}, \tag{13}$$

where α is a suitable constant, and $0 \leq \alpha \leq 1$. In our experiments, α is empirically set to be 0.5. The projection vector $\mathbf{a}$ that maximizes (13) is given by the maximum eigenvalue solution to the generalized eigenvalue problem:

$$X(\alpha L_b + (1-\alpha) W_w) X^T \mathbf{a} = \lambda X D_w X^T \mathbf{a}. \tag{14}$$

Let the column vector $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d$ be the solutions of (14) ordered according to their eigenvalues $\lambda_1 > \cdots > \lambda_d$. Thus, the embedding is given as follows:

$$\begin{aligned} \mathbf{x}_i \rightarrow \mathbf{y}_i &= A^T \mathbf{x}_i, \\ A &= (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d), \end{aligned}$$

where $\mathbf{y}_i$ is a d-dimensional vector, and A is an $n \times d$ matrix.

Note that if the number of samples (m) is less than the number of features (n), then $rank(X) \leq m$. Consequently,

$$\begin{aligned} &rank(X D_w X^T) \leq m \text{ and} \\ &rank(X(\alpha L_b + (1-\alpha) W_w) X^T) \leq m. \end{aligned}$$

The fact that $X D_w X^T$ and $X(\alpha L_b + (1-\alpha) W_w) X^T$ are $n \times n$ matrices implies that both of them are singular. In this case, one may first apply PCA to remove the components corresponding to zero eigenvalues.

Our algorithm also shares some common properties with some recent work on combing classification and metric learning such as *Distance-Function Alignment* (DA_{lign}) [42] and *Spectral Kernel Learning* (SKL) [25]. DA_{lign} first constructs a data similarity matrix by using some initial kernels (for example, Gaussian kernels or polynomial kernels). It then incorporates contextual information into the kernel and finds a distance metric, which is consistent to the new kernel. SKL is built on the principles of kernel target alignment and unsupervised kernel design [13]. It learns a new kernel by assessing the relationship between the new kernel and a target kernel induced by the given labels. All of these methods can be used to learn a similarity function. In addition, all of them can be thought of as semisupervised learning algorithms, since they make use of both labeled and unlabeled data points. In the following, we list the major differences between them:

1. Both DA_{lign} and SKL aim at learning a kernel and distance metric, whereas our algorithm aims at learning representations for the data points. With the new representations, one can apply the standard classification or clustering techniques in the new representation space. Both DA_{lign} and SKL fail to provide new representations of the data points.
2. Both DA_{lign} and SKL try to discover the *global* geometrical structure in the data. They first generate a gram matrix (similarity matrix) characterizing the similarity between any pairs of data points. Unlike them, our algorithm is motivated by discovering the *local* manifold structure. Therefore, we construct a sparse nearest neighbor graph to model the local geometry.
3. The objective functions are different. Both DA_{lign} and SKL are based on the kernel alignment principle [13] and optimize the alignment to the target kernel (or target distance function), whereas our algorithm is based on the maximum margin principle. Specifically, if two points are sufficiently close or share the same label, then their maps should also be sufficiently close. In addition, if two points have different labels, then their maps should be far apart.

3.4 Complexity Analysis

The complexity of MMP is dominated by three parts: k nearest neighbor search, matrix multiplication, and

solving a generalized eigenvector problem. Consider m data points in n-dimensional space. For the k nearest neighbor search, the complexity is $O((n+k)m^2)$. nm^2 stands for the complexity of computing the distances between any two data points. km^2 stands for the complexity of finding the k nearest neighbors for all the data points. The complexities for calculating the matrices $X(\alpha L_b + (1-\alpha)W_w)X^T$ and XD_wX^T are $O(m^2n + mn^2)$ and $O(mn^2)$, respectively. The third part is solving a generalized eigenvector problem $A\mathbf{a} = \lambda B\mathbf{a}$, where A and B are $n \times n$ matrices. To solve this generalized eigenvector problem, we need first to compute the Singular Value Decomposition (SVD) of the matrix B. The complexity of SVD is $O(n^3)$. Then, to project the data points into a d-dimensional subspace, we need to compute the first d smallest eigenvectors of an $n \times n$ matrix, whose complexity is $O(dn^2)$. Thus, the total complexity of the generalized eigenvector problem is $O((n+d)n^2)$. Therefore, the time complexity of the MMP algorithm is $O((n+k)m^2 + (m+n+d)n^2)$. Since $k \ll m$ and $d \ll n$, the complexity of MMP is determined by the number of data points and the number of features.

4 Content-Based Image Retrieval Using Maximum Margin Projection

In this section, we describe how MMP can be applied to CBIR. In particular, we consider relevance-feedback-driven image retrieval.

4.1 Low-Level Image Representation

Low-level image representation is a crucial problem in CBIR. General visual features include color, texture, shape, etc. Color and texture features are the most extensively used visual features in CBIR. Compared with color and texture features, shape features are usually described after images have been segmented into regions or objects. Since robust and accurate image segmentation is difficult to achieve, the use of shape features for image retrieval has been limited to special applications where objects or regions are readily available.

In this work, We combine a 64-dimensional color histogram and a 64-dimensional Color Texture Moment (CTM) [43] to represent the images. The color histogram is calculated using $4 \times 4 \times 4$ bins in HSV space. CTM, which was proposed by Yu et al. [43], integrates the color and texture characteristics of the image in a compact form. CTM adopts local Fourier transform as a texture representation scheme and derives eight characteristic maps for describing different aspects of co-occurrence relations of image pixels in each channel of the (SVcosH, SVsinH, and V) color space. Then, CTM calculates the first and second moments of these maps as a representation of the natural color image pixel distribution. See [43] for details.

In fact, if the low-level visual features are accurate enough, that is, if the euclidean distances in the low-level feature space can accurately reflect the semantic relationship between images, then one can simply perform nearest neighbor search in the low-level feature space, and the retrieval performance can be guaranteed. Unfortunately, there is no strong connection between low-level visual features and high-level semantic concepts based on the state-of-the-art computer vision techniques. Thus, one has to resort to user interactions to discover the semantic structure in the data.

4.2 Relevance Feedback Image Retrieval

Relevance feedback is one of the most important techniques for narrowing down the gap between low-level visual features and high-level semantic concepts [31]. Traditionally, the user's relevance feedbacks are used to update the query vector or adjust the weighting of different dimensions. This process can be viewed as an online learning process in which the image retrieval system acts as a learner, whereas the user acts as a teacher. The typical retrieval process is outlined as follows:

1. The user submits a query image example to the system. The system ranks the images in the database according to some predefined distance metric and presents to the user the top ranked images.
2. The user provides his relevance feedbacks to the system by labeling images as "relevant" or "irrelevant."
3. The system uses the user-provided information to rerank the images in the database and returns the top images to the user. Repeat step 2 until the user is satisfied.

Here, we describe how the user's relevance feedbacks can be used to update the within-class and between-class graphs for discovering the semantic and geometrical structure of the image database. At the beginning of the retrieval, the user submits a query image $\mathbf{q}$. The images in the database are ranked according to their euclidean distances to $\mathbf{q}$, and the top images are presented to the user. The user is then required to mark the top returned images as "relevant" or "irrelevant." Naturally, we can divide the images into two classes. Let $l(\mathbf{x})$ denote the label of image $\mathbf{x}$. Thus, $l(\mathbf{x}_i) = 1$ if $\mathbf{x}_i$ is relevant to $\mathbf{q}$, and $l(\mathbf{x}_i) = -1$ if $\mathbf{x}_i$ is irrelevant to $\mathbf{q}$. Based on these relevance feedbacks, we can construct the within-class and between-class graphs as described in Section 3. Note that the nearest neighbor graph G can be constructed offline. At the beginning of the retrieval, there are no relevance feedbacks available. Thus, by our definition, the within-class graph is simply G, whereas the between-class graph is an empty graph. Consequently, $W_w = W$, $W_b = 0$, $D_w = D$, and the optimization problem (13) reduces to

$$\arg \max_{\mathbf{a}^T XDX^T\mathbf{a}=1} \mathbf{a}^T XWX^T\mathbf{a}. \tag{15}$$

In addition, the corresponding generalized eigenvector problem becomes

$$XWX^T\mathbf{a} = \lambda XDX^T\mathbf{a}. \tag{16}$$

Clearly, in this case, MMP reduces to LLP [22], [19]. During each iteration of relevance feedback, we only need to update the within-class and between-class graphs according to (6) and (7). By applying our MMP algorithm, we can then project the images into a lower dimensional subspace, in which semantically related images tend to be close to

each other. Let A be the transformation matrix; that is, $\mathbf{x}'_i = A^T\mathbf{x}_i$, and $\mathbf{q}' = A^T\mathbf{q}$. The distance between $\mathbf{x}'_i$ and $\mathbf{q}'$ can be computed as follows:

$$dist(\mathbf{x}'_i, \mathbf{q}') = \sqrt{(\mathbf{x}'_i - \mathbf{q}')^T(\mathbf{x}'_i - \mathbf{q}')} = \sqrt{(\mathbf{x}_i - \mathbf{q})^T AA^T(\mathbf{x}_i - \mathbf{q})}.$$

When MMP is applied, one needs to estimate the optimal dimensionality of the subspace. It would be important to note that our MMP algorithm is intrinsically a graph embedding algorithm. It is closely related to Laplacian Eigenmaps [1], LLP [22], spectral clustering [29], and Normalized Cut [32]. All of these algorithms are fundamentally based on spectral graph embedding and partitioning [10]. Previous studies have shown that when there are c classes, the optimal dimensionality should be close to c [23], [29]. Therefore, when there is no prior knowledge about the dimensionality and a brute-force search is infeasible, one can simply keep two dimensions, considering that there are two classes (relevant or not) for image retrieval. This theoretical result is strengthened by our experiments. See Fig. 5 for details.

In many situations, the number of images in the database can extremely be large, which makes the computation of our algorithm infeasible. In order to reduce the computational complexity, we do not take all the images in the database to construct the within-class and between-class graphs. Instead, we only take the top 300 images at the previous retrieval iteration, plus the labeled images, to find the optimal projection.

5 Experimental Results

In this section, we evaluate the performance of our proposed algorithm on a large image database. We begin with a description of the image database.

5.1 Experimental Design

The image database that we used consists of 7,900 images of 79 semantic categories from the COREL data set. It is a large and heterogeneous image set. Each image is represented as a 128-dimensional vector, as described in Section 4.1.

To exhibit the advantages of using our algorithm, we need a reliable way of evaluating the retrieval performance and the comparisons with other algorithms. We list different aspects of the experimental design in the following.

5.1.1 Evaluation Metrics

We use *precision-scope curve* and *precision rate* [26] to evaluate the effectiveness of the image retrieval algorithms. The scope is specified by the number N of top-ranked images presented to the user. The precision is the ratio of the number of relevant images presented to the user to the scope N. The precision-scope curve describes the precision with various scopes and thus gives the overall performance evaluation of the algorithms. On the other hand, the precision rate emphasizes the precision at a particular value of scope. In general, it is appropriate to present 20 images on a screen. Putting more images on a screen might affect the quality of the presented images. Therefore, the precision at the top 20 ($N = 20$) is especially important.

In a real image retrieval system, a query image is usually not in the image database. To simulate such an environment, we use *fivefold cross validation* to evaluate the algorithms. More precisely, we divide the whole image database into five subsets of equal size. Thus, there are 20 images per category in each subset. At each run of cross validation, one subset is selected as the query set, and the other four subsets are used as the database for retrieval. The precision-scope curve and precision rate are computed by averaging the results from the fivefold cross validation.

5.1.2 Automatic Relevance Feedback Scheme

We designed an automatic feedback scheme to model the retrieval process. For each submitted query, our system retrieves and ranks the images in the database. The top 10 ranked images were selected as the feedback images, and their label information (relevant or irrelevant) is used for reranking. Note that the images that have been selected at previous iterations are excluded from later selections. For each query, the automatic relevance feedback mechanism is performed for four iterations.

It is important to note that the automatic relevance feedback scheme used here is different from the ones described in [19], [27]. In [19], [27], the top four relevant and irrelevant images were selected as the feedback images. However, this may not be practical. In real-world image retrieval systems, it is possible that most of the top-ranked images are relevant (or irrelevant). Thus, it is difficult for the user to find both four relevant images and four irrelevant images. It is more reasonable for the users to provide feedback information only on the first screen shot (10 or 20 images).

5.1.3 Compared Algorithms

To demonstrate the effectiveness of our proposed image retrieval algorithm (MMP), we compare it with two state-of-the-art algorithms, that is, ARE [27] and Support Vector Machine (SVM), and a canonical algorithm, that is, LDA. Both MMP and ARE take the manifold structure into account and try to learn a subspace in which the euclidean distances can better reflect the semantic structure of the images. LDA is a canonical supervised dimensionality reduction algorithm. It projects the images into a one-dimensional space, in which the euclidean distances are used to rerank the images.

ARE performs the relevance feedback image retrieval by using three graphs, that is, the similarity-relational graph on the whole image database and two feedback-relational graphs that incorporate the user provided positive and negative examples. In the comparison experiments reported in [27], ARE is superior to the Incremental LLP approach [19]. A crucial problem in ARE is how we can determine the dimensionality of the subspace. In our experiments, we iterate all the dimensions and select the dimension with respect to the best performance. For both ARE and MMP, the euclidean distances in the reduced subspace are used for ranking the images in the database.

SVM implements the idea of mapping input data into a high-dimensional feature space, where a maximal margin

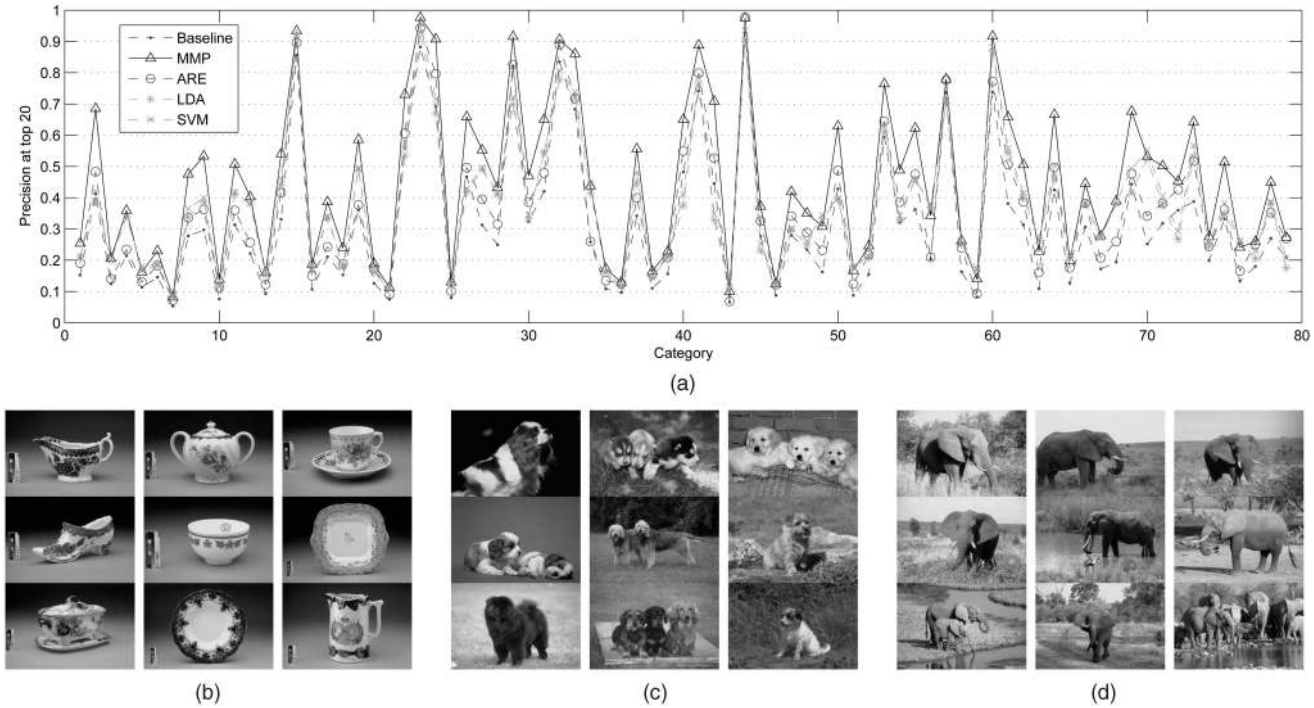

Fig. 2. (a) Precision at the top 20 returns of the four algorithms after the first feedback iteration. Our MMP algorithm is the best for almost all the categories. (b), (c), and (d) Sample images from categories 24, 25, and 30, respectively.

hyperplane is constructed [41]. Previous work has demonstrated that SVM can significantly improve retrieval performance [9], [24], [40]. The labeled images (query image and feedback images) $\{\mathbf{x}_i, y_i\}$ are used to learn a classification function by SVM. The LIBSVM software [6] was used in our system to solve the SVM optimization problem. Leave-one-out cross validation on the training images is applied to select the parameters in SVM. The RBF kernel is used in SVM.

5.2 Image Retrieval Performance

In the real world, it is not practical to require the user to provide many rounds of feedbacks. The retrieval performance after the first two rounds of feedbacks (especially the first round) is the most important. Fig. 2 shows the precision at the top 20 after the first round of feedbacks for all the 79 categories. The *baseline* curve describes the initial retrieval result without feedback information. Specifically, at the beginning of retrieval, the euclidean distances in the original 128-dimensional space are used to rank the images in the database. After the user provides relevance feedbacks, the ARE, SVM, LDA, and MMP algorithms are then applied to rerank the images in the database. The detailed results are also shown in Table 1. As can be seen, the retrieval performance of these algorithms varies with different categories. There are some *easy* categories, on which all the algorithms perform well, and some *hard* categories, on which all the algorithms perform poorly. Since the features that we used in our experiments are color and texture features, those categories containing images with similar colors and textures (for example, category 24 in Fig. 2b) get very good retrieval performance, whereas those categories containing images with different colors and textures (for example, category 25 in Fig. 2c) get poor retrieval performance. Among all the 79 categories, our MMP approach performs the best on 62 categories. For the remaining 17 categories, SVM performs the best on 14 of them, LDA performs the best on two of them, and ARE performs best on one of them.

Fig. 3 shows the average *precision-scope* curves of the different algorithms for the first two feedback iterations. Our MMP algorithm outperforms the other four algorithms on the entire scope. The performances of SVM and LDA are very close to each other. ARE performs better than SVM and LDA only when the scope is less than 15. Both SVM and LDA consistently outperform ARE at the second round of feedbacks. All of these four algorithms MMP, SVM, LDA, and ARE are significantly better than the baseline, which indicates that the user-provided relevance feedbacks are very helpful in improving the retrieval performance. By iteratively adding the user's feedbacks, the corresponding precisions (at the top 10, top 20, and top 50) of the algorithms are, respectively, shown in Fig. 4. As can be seen, our MMP algorithm performs the best, especially at the first round of relevance feedback. As the number of feedbacks increases, the performance difference between MMP and SVM gets smaller.

The actual computational time of different algorithms is given in Table 2. All of these four algorithms can respond to the user's query very fast, that is, within 0.1 s. Our MMP algorithm is as fast as ARE and slightly slower than SVM. LDA is the fastest. All of the experiments were performed on a Pentium IV 3.20-GHz Windows XP machine with a 2-Gbyte memory. In order to measure the significance of the improvement obtained by our algorithm, we did a t-test on the precision at the top 20 of the different algorithms, as shown in Table 3. For the comparison between our algorithm and SVM, ARE, LDA, and the baseline method,

TABLE 1
Precision at the Top 20 Returns of the Four Algorithms after the First Feedback Iteration (mean ± std-dev%)

	Baseline	MMP	ARE	LDA	SVM		Baseline	MMP	ARE	LDA	SVM
1	15.3±1.3	**25.4±3.1**	19.1±1.7	21.1±1.9	22.1±1.7	41	74.1±4.6	**88.8±2.7**	79.9±3.2	76.3±6.2	81.7±4.3
2	41.2±4.8	**68.6±7.6**	48.3±5.2	38.8±7.1	45.4±6.1	42	44.6±5.2	**70.9±7.7**	52.7±4.8	33.0±5.7	39.2±6.2
3	12.5±1.0	**20.6±2.9**	14.1±1.3	20.0±2.2	20.3±2.3	43	6.6 ±0.5	10.1±0.9	6.9 ±0.4	11.1±0.9	**11.6±0.8**
4	21.4±3.5	**35.9±7.4**	23.5±4.0	34.0±5.3	35.9±5.5	44	93.2±2.4	97.5±0.7	**97.7±0.3**	90.8±4.4	93.7±2.8
5	11.4±1.2	16.2±2.4	13.1±1.6	16.8±1.8	**18.6±2.0**	45	27.7±4.4	**37.3±6.3**	32.6±4.6	23.1±3.3	28.6±3.9
6	14.6±1.7	**23.0±3.5**	18.3±2.4	18.2±2.3	19.0±2.3	46	8.8 ±0.6	12.4±1.1	12.2±1.1	12.1±1.0	**12.8±1.0**
7	5.3 ±0.4	8.2 ±0.6	7.2 ±0.4	**8.5 ±0.5**	8.3 ±0.6	47	27.9±2.7	**41.9±5.3**	34.1±3.3	29.7±4.4	34.1±3.9
8	27.8±4.1	**47.6±7.2**	33.6±4.1	33.8±5.2	39.8±5.6	48	23.4±2.4	**35.1±4.0**	28.9±2.7	24.9±2.0	28.2±2.2
9	29.7±3.0	**53.3±7.0**	36.3±3.1	38.7±3.8	41.7±4.0	49	16.3±1.6	30.8±5.6	23.2±3.1	33.2±4.6	**34.9±5.0**
10	7.6 ±0.6	**13.9±1.2**	11.3±1.0	11.9±1.1	12.0±1.2	50	42.9±4.6	**62.9±5.7**	48.7±3.9	39.4±5.3	43.8±5.1
11	31.4±4.3	**50.8±5.6**	35.9±4.3	41.2±4.5	44.3±4.2	51	8.8 ±0.9	16.7±2.8	12.4±1.2	18.1±2.1	**18.4±2.0**
12	22.2±2.7	**40.3±7.1**	25.7±3.1	38.0±5.1	38.2±4.9	52	15.5±1.2	**24.6±2.8**	21.7±1.7	20.8±1.8	22.9±2.2
13	9.3 ±2.1	16.0±4.8	12.2±3.0	15.4±3.8	**17.0±4.4**	53	59.4±11.6	**76.4±10.2**	64.5±9.7	61.1±10.4	66.7±10.1
14	33.1±4.2	**54.0±9.3**	41.6±4.7	44.0±5.7	49.2±6.1	54	32.0±3.5	**48.9±6.5**	38.6±4.0	32.9±4.4	37.5±4.0
15	85.6±7.0	**93.4±2.1**	89.7±5.1	92.6±2.7	91.6±3.2	55	36.3±3.6	**62.3±5.0**	47.5±3.7	46.1±4.5	50.5±4.4
16	10.7±0.8	**18.7±2.1**	15.0±1.3	16.6±1.9	17.6±2.0	56	20.4±3.1	34.3±7.9	21.0±4.3	36.9±6.0	**38.1±6.2**
17	21.1±2.5	**38.7±7.4**	24.3±2.9	33.6±6.2	35.0±5.6	57	73.7±13.2	**78.0±12.5**	77.7±12.6	70.4±13.6	72.4±12.8
18	15.3±1.8	**23.9±5.0**	18.8±2.1	18.0±2.4	19.7±2.7	58	16.3±1.9	**26.2±4.8**	23.9±3.3	25.1±3.4	24.5±2.6
19	36.1±5.4	**58.6±7.7**	37.7±4.7	49.1±4.7	52.4±5.3	59	8.3 ±0.7	14.0±1.6	9.3 ±0.8	17.0±1.8	**17.3±1.9**
20	12.6±1.1	**18.9±2.2**	17.3±1.4	16.0±1.7	17.9±1.7	60	73.9±8.2	**91.7±3.4**	77.2±6.2	86.9±2.6	90.2±1.9
21	7.5 ±0.5	11.3±1.3	9.0 ±0.7	10.9±0.9	**12.0±0.9**	61	38.2±5.4	**65.8±6.1**	50.7±5.5	54.4±6.3	59.1±5.1
22	55.8±6.8	**73.0±6.1**	60.7±5.4	53.5±7.7	60.4±7.6	62	31.3±5.2	**50.6±9.9**	38.7±5.6	40.0±6.1	45.4±6.6
23	88.2±5.1	**97.7±0.8**	94.3±1.8	93.0±2.5	96.2±1.0	63	10.9±0.9	22.8±2.7	16.1±1.4	**27.0±3.0**	26.2±2.8
24	69.2±6.5	**90.8±3.3**	79.7±4.2	66.9±8.6	74.2±7.6	64	42.5±6.5	**66.6±8.2**	49.7±5.1	45.8±6.2	53.3±6.1
25	8.0 ±0.6	13.1±1.7	10.2±0.9	12.8±1.0	**13.6±1.3**	65	12.7±1.1	19.8±2.8	17.6±1.7	21.1±2.2	**21.8±2.3**
26	46.7±6.4	**65.9±6.7**	49.6±5.7	41.8±6.6	47.4±6.0	66	30.6±3.4	**44.5±6.2**	38.2±3.8	38.2±3.6	42.3±3.6
27	31.3±7.6	**55.3±13.9**	39.5±9.6	49.0±9.7	49.9±10.1	67	17.3±1.8	**27.8±3.6**	20.8±2.0	27.7±3.3	27.3±3.2
28	24.9±4.4	**43.2±9.1**	31.5±6.0	41.0±8.6	42.7±9.2	68	19.4±3.3	**38.9±10.3**	26.0±5.4	38.1±7.9	37.8±8.2
29	82.0±3.9	**91.5±0.8**	82.6±2.9	69.7±10.0	76.1±7.5	69	44.4±5.5	**67.7±6.3**	47.7±4.3	41.6±6.2	52.0±5.8
30	32.3±3.8	**47.1±7.6**	38.6±4.3	33.2±4.4	37.4±4.6	70	25.2±3.8	53.1±10.7	34.2±6.6	52.4±7.5	**56.0±8.2**
31	42.0±6.2	**65.1±8.2**	47.9±6.5	54.6±5.4	56.7±6.2	71	31.8±4.9	**50.2±9.5**	38.2±5.2	37.8±6.1	43.7±6.9
32	83.5±5.4	**90.6±3.1**	88.9±3.0	77.8±8.2	82.6±5.6	72	36.0±3.6	**45.2±4.4**	42.6±4.5	26.7±2.5	32.9±2.9
33	68.3±8.7	**86.1±4.0**	71.8±8.0	72.2±9.1	76.7±7.3	73	38.8±4.9	**64.2±8.6**	51.8±6.1	53.2±5.9	56.6±6.7
34	25.9±4.5	43.8±7.8	26.0±4.0	41.7±5.5	**46.9±6.3**	74	19.9±2.3	**27.5±5.0**	24.4±3.2	24.2±2.9	27.3±3.4
35	10.9±1.3	**17.3±3.0**	13.5±1.5	16.4±2.0	16.8±2.2	75	33.2±5.3	**51.4±9.1**	36.3±5.5	34.9±5.8	40.9±6.9
36	9.7 ±0.6	**13.0±1.3**	12.5±0.8	11.7±0.9	12.5±0.9	76	13.3±1.4	24.2±4.0	16.6±2.2	26.0±3.6	**26.1±3.7**
37	34.2±8.0	**55.7±11.5**	40.0±7.9	44.8±9.5	49.5±9.6	77	18.0±1.3	**26.0±3.1**	25.1±1.9	20.6±1.7	24.4±2.2
38	11.0±0.9	**16.5±2.1**	15.5±1.2	14.5±1.1	15.4±1.2	78	27.0±2.9	**45.0±4.6**	35.3±3.9	38.2±3.3	40.9±3.9
39	15.6±0.9	**22.9±2.2**	20.9±1.5	20.9±2.0	22.7±1.9	79	20.9±2.3	**27.3±3.9**	26.9±3.3	17.5±2.3	22.6±2.9
40	48.2±6.1	**65.1±7.8**	54.9±6.5	37.6±6.1	43.3±6.2						

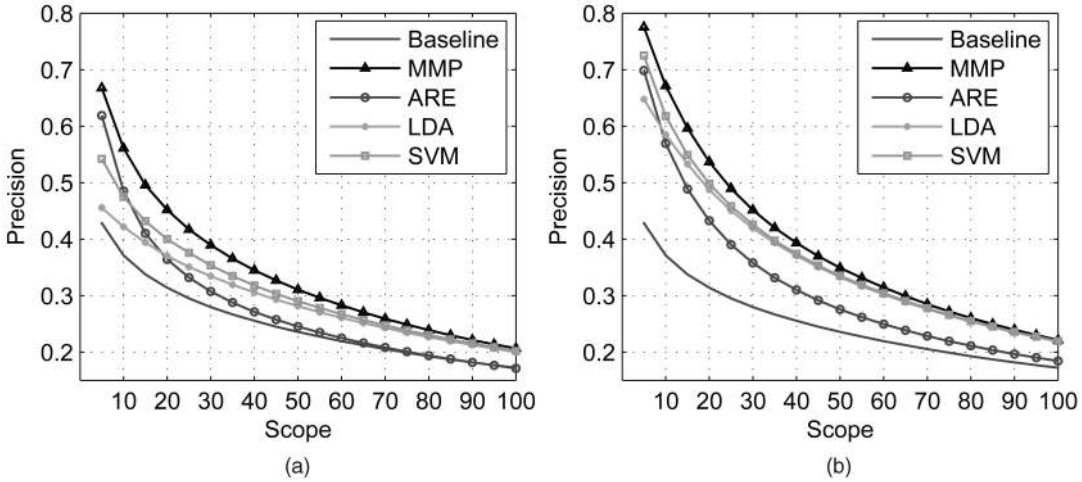

Fig. 3. The average *precision-scope* curves of different algorithms for the first two feedback iterations. The MMP algorithm performs the best on the entire scope. (a) Feedback iteration 1. (b) Feedback iteration 2.

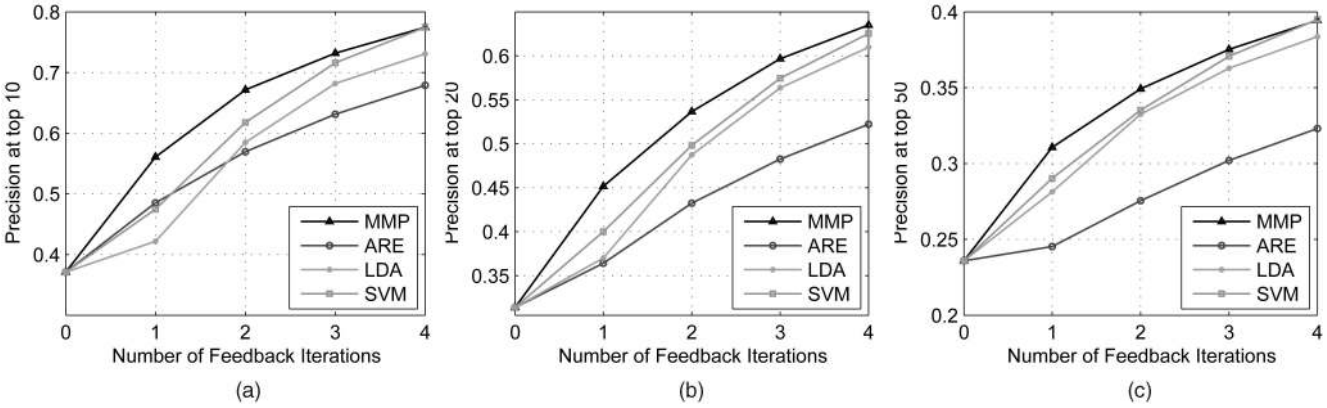

Fig. 4. Precisions at the (a) top 10 (P@10), (b) top 20 (P@20), and (c) top 50 (P@50) of the three algorithms. As can be seen, our MMP algorithm performs the best, especially at the first round of relevance feedback.

the P values are $1e{-}9$, $1e{-}21$, $1e{-}12$, and $1e{-}28$, respectively. All of these tests show that the improvement of our proposed method over the other methods are significant.

5.3 Embedding Dimensions

Unlike SVM, the ARE and MMP algorithms are subspace learning algorithms. For ARE and MMP, there is a problem on how the dimensionality of the subspace can be determined. Fig. 5 shows the retrieval performance of ARE with different numbers of dimensions. Each curve corresponds to an iteration. It is interesting to see that our MMP algorithm always gets the best performance with two dimensions. For the ARE algorithm, the best performance is obtained with $20 \sim 40$ dimensions. In practice, it is more difficult for ARE to estimate the optimal dimensionality.

5.4 Model Selection

Model selection is a crucial problem in most of the learning problems. In some situations, the learning performance may drastically vary with different choices of the parameters, and we have to apply some model selection methods (such as Cross Validation and Bootstrapping [17]) for estimating the generalization error. In this section, we evaluate the performance of our algorithm with different values of the parameters.

In our MMP algorithm, there are three parameters: k, α, and γ. k is the number of nearest neighbors. The parameter α controls the weight between a within-class graph and a between-class graph. The parameter γ controls the weight between labeled and unlabeled images. In our previous experiments, we empirically set them as $k = 5$, $\alpha = 0.5$, and $\gamma = 50$. Fig. 6 shows the performance of our algorithm (P@10, P@20, and P@30) after the first round of feedbacks with respect to different values of these three parameters. As can be seen, our algorithm is not sensitive to α and γ. For the value of k, since our algorithm tries to discover the *local* geometrical and discriminant structures of the data space, it is usually set to a small number, which is typically less than 10.

TABLE 2
Average Runtime of Different Algorithms for Processing One Query

	Time at different feedback iterations (s)			
	1	2	3	4
MMP	0.066	0.071	0.075	0.080
LDA	0.012	0.016	0.019	0.023
SVM	0.048	0.058	0.067	0.074
ARE	0.067	0.073	0.077	0.082

5.5 Visualization of Semantics

In the previous sections, we have presented some quantitative results of our algorithm. In this section, we give a visual example to demonstrate how our algorithm works during the retrieval process. It would be important to note that the original idea of this experimental scheme for visualizing semantics comes from [27].

Since our MMP algorithm is essentially a dimensionality reduction algorithm, we can project the images into a 2D plane for visualization. Fig. 7 shows the embedding results for two queries, that is, *firework* and *horse*. The two query images are presented at the top of the figure. Each query has three embedding plots, and each plot corresponds to an iteration. The two plots on the first row demonstrate the initial embedding results. Since initially, there are no feedbacks, we apply PCA for 2D embedding. After the relevance feedbacks are provided, our MMP algorithm is applied for 2D embedding. The embedding plots on the second and third rows correspond to the first and second rounds of relevance feedback. On the left-hand side of each plot, we present the labeled relevant images (denoted by F^+) and irrelevant images (denoted by F^-). Due to space limitations, we show at most four images for the relevant (irrelevant) set. In each plot, the star point stands for the query image. The dark gray points stand for the images relevant to the query image, and the light gray points stand for the images irrelevant to the query image. The large dark gray and large light gray points, respectively, denote the relevant and irrelevant feedbacks that will be returned to the system (or, in practice, the user) for labeling. The region centered at the query image is zoomed-in to give a better

TABLE 3
P Values of the T-Tests on P@20 of Different Algorithms

	Baseline	ARE	LDA	SVM
MMP	1e-28	1e-21	1e-12	1e-9

Note that, P-value indicates that the difference is significant.

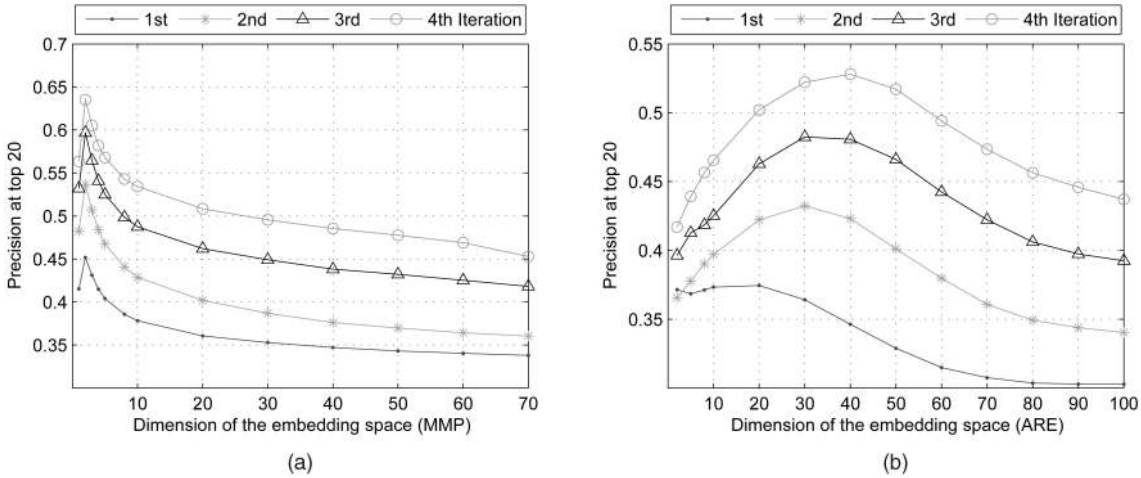

Fig. 5. (a) The performance of MMP versus the dimensionality. MMP always achieves the best performance at dimensionality 2. This property shows that MMP does not suffer from the problem of dimensionality estimation. (b) The performance of ARE versus the dimensionality. The best performances at different feedback iterations appear at different dimensions, which makes it hard to estimate the optimal dimensionality in practice.

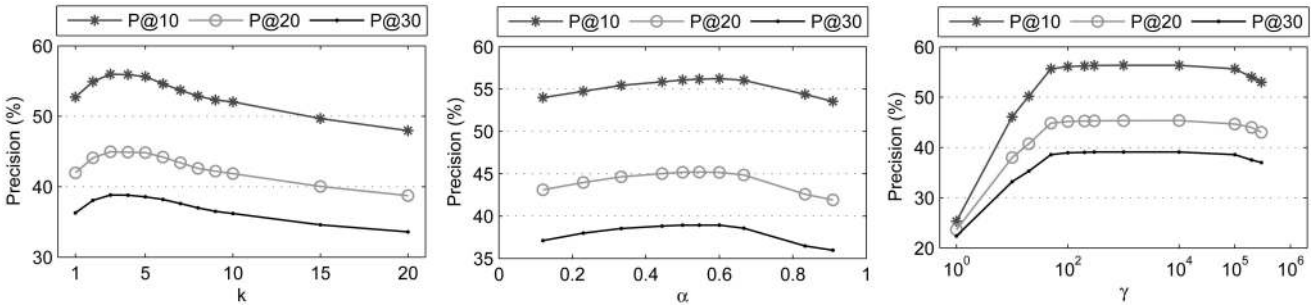

Fig. 6. Model selection for MMP: retrieval precision versus different values of the parameters k, α, and γ.

view of the 100 nearest neighbors of the query image. As can be seen, as the relevance feedbacks are provided, the relevant images (dark gray points) progressively gather together around the query image, whereas the irrelevant images (light gray points) go far from the query image. For the visualization of the ARE algorithm, the reader is referred to [27].

6 Conclusion and Future Work

This paper presents a novel manifold learning algorithm, called MMP, for image retrieval. In the first step, we construct a between-class nearest neighbor graph and a within-class nearest neighbor graph to model both geometrical and discriminant structures in the data. The standard spectral technique is then used to find an optimal projection, which respects the graph structure. This way, the euclidean distances in the reduced subspace can reflect the semantic structure in the data to some extent. In comparison with two state-of-the-art methods, that is, ARE and SVM, the experimental results validate that the new method achieves a significantly higher precision for image retrieval. Our proposed MMP algorithm performs especially good at the first round of relevance feedback (10 feedbacks). As more feedbacks are provided, the performance difference between MMP and SVM gets smaller. Both MMP, SVM, and ARE significantly outperform the baseline, which indicates that relevance feedback is important for image retrieval.

Several questions remain to be investigated in our future work:

1. There are currently two directions for relevance feedback image retrieval. One is classification based (for example, SVM), and the other is metric learning based (for example, ARE and MMP). It remains unclear which direction is more promising. In general, metric-learning-based approaches are more flexible, since they can be thought of as data preprocessing, and the other learning techniques may be applied in the new metric space. On the other hand, with a sufficient number of training samples, the classification-based approaches may be able to find an optimal boundary between relevant and irrelevant images and thus may outperform metric-learning-based approaches. In the machine learning community, recently, there has been a lot of interest in geometrically motivated approaches to data classification in high-dimensional spaces [2]. These approaches can be thought of as a combination of metric learning and classification. A more systematic investigation of these approaches for image retrieval needs to be carried out.
2. In this paper, we consider the image retrieval problem on a small, static, and closed-domain image data. A much more challenging domain is the World Wide Web (WWW). When it comes to searching WWW images, it is possible to collect a large amount of user click information. This information can then be used as training data to perform *pseudo*relevance feedback by applying our techniques.

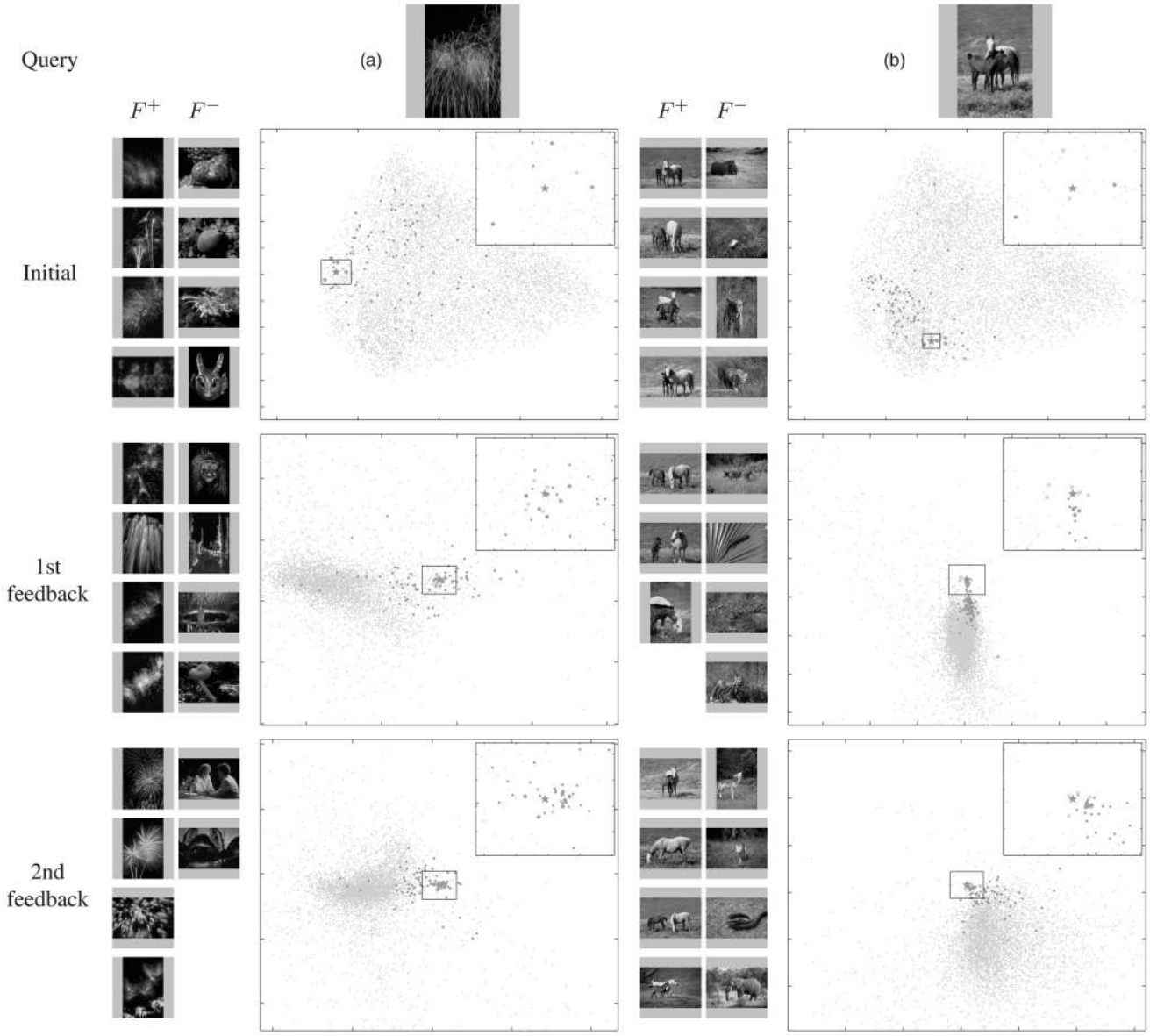

Fig. 7. Semantic visualization of (a) *firework* and (b) *horse*. Each (zoomed-in) rectangle encloses the 100 nearest neighbors of the query image. Through iterative feedbacks, the relevant images (dark gray points) progressively gather together around the query, whereas the irrelevant ones (light gray points) go far from the query. Note that the original idea of this experimental scheme for semantic visualization comes from [27].

3. It would be very interesting to explore different ways of constructing the image graph to model the semantic structure in the data. There is no reason to believe that the nearest neighbor graph is the only or the most natural choice. For example, for Web image search, it may be more natural to use the hyperlink information to construct the graph.

REFERENCES

[1] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems 14,* pp. 585-591, MIT Press, 2001.

[2] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Examples," *J. Machine Learning Research,* 2006.

[3] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N.L. Roux, and M. Ouimet, "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," *Advances in Neural Information Processing Systems 16,* 2003.

[4] J. Bi, Y. Chen, and J.Z. Wang, "A Sparse Support Vector Machine Approach to Region-Based Image Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '04),* 2004.

[5] D. Cai, X. He, K. Zhou, J. Han, and H. Bao, "Locality Sensitive Discriminant Analysis," *Proc. 20th Int'l Joint Conf. Artificial Intelligence (IJCAI '07),* Jan. 2007.

[6] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines,* http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[7] E. Chang, K. Goh, G. Sychay, and G. Wu, "Cbsa: Content-Based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machines," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 13, no. 1, pp. 26-38, Jan. 2003.

[8] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local Discriminant Embedding and Its Variants," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '05),* 2005.

[9] Y. Chen, X.S. Zhou, and T.S. Huang, "One-Class SVM for Learning in Image Retrieval," *Proc. IEEE Int'l Conf. Image Processing (ICIP '01),* pp. 34-37, 2001.

[10] F.R.K. Chung, "Spectral Graph Theory," *Proc. CBMS Regional Conf. Series in Math.,* vol. 92, 1997.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms,* second ed. MIT Press, 2001.

[12] I.J. Cox, T.P. Minka, T.V. Papathomas, and P.N. Yianilos, "The Bayesian Image Retrieval System, Pichunter: Theory, Implementation, and Psychophysical Experiments," *IEEE Trans. Image Processing,* vol. 9, pp. 20-37, 2000.
[13] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On Kernel-Target Alignment," *Advances in Neural Information Processing Systems 14,* 2001.
[14] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. Wiley-Interscience, 2000.
[15] K.-S. Goh, E.Y. Chang, and W.-C. Lai, "Multimodal Concept-Dependent Active Learning for Image Retrieval," *Proc. 12th Ann. ACM Int'l Conf. Multimedia (Multimedia '04),* Oct. 2004.
[16] A. Grama, G. Karypis, V. Kumar, and A. Gupta, *An Introduction to Parallel Computing: Design and Analysis of Algorithms,* second ed. Addison Wesley, 2003.
[17] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2001.
[18] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Manifold-Ranking-Based Image Retrieval," *Proc. 12th Ann. ACM Int'l Conf. Multimedia (Multimedia '04),* Oct. 2004.
[19] X. He, "Incremental Semi-Supervised Subspace Learning for Image Retrieval," *Proc. 12th Ann. ACM Int'l Conf. Multimedia (Multimedia '04),* Oct. 2004.
[20] X. He, D. Cai, and W. Min, "Statistical and Computational Analysis of LPP," *Proc. 22nd Int'l Conf. Machine Learning (ICML '05),* 2005.
[21] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood Preserving Embedding," *Proc. 11th Int'l Conf. Computer Vision (ICCV '05),* 2005.
[22] X. He and P. Niyogi, "Locality Preserving Projections," *Advances in Neural Information Processing Systems 16,* MIT Press, 2003.
[23] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face Recognition Using Laplacianfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 3, pp. 328-340, Mar. 2005.
[24] S.C. Hoi, M.R. Lyu, and R. Jin, "A Unified Log-Based Relevance Feedback Scheme for Image Retrieval," *IEEE Trans. Knowledge and Data Eng.,* vol. 18, no. 4, pp. 509-524, Apr. 2006.
[25] S.C.H. Hoi, M.R. Lyu, and E.Y. Chang, "Learning the Unified Kernel Machines for Classification," *Proc. 12th ACM SIGKDD Int'l Conf. KNowledge Discovery and Data Mining,* pp. 187-196, 2006.
[26] D.P. Huijsmans and N. Sebe, "How to Complete Performance Graphs in Content-Based Image Retrieval: Add Generality and Normalize Scope," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 2, pp. 245-251, Feb. 2005.
[27] Y.-Y. Lin, T.-L. Liu, and H.-T. Chen, "Semantic Manifold Learning for Image Retrieval," *Proc. 13th Ann. ACM Int'l Conf. Multimedia (Multimedia '05),* Nov. 2005.
[28] W.-Y. Ma and B.S. Manjunath, "Netra: A Toolbox for Navigating Large Image Databases," *Multimedia Systems,* vol. 7, no. 3, May 1999.
[29] A.Y. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems 14,* pp. 849-856, MIT Press, 2001.
[30] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 500, pp. 2323-2326, 2000.
[31] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 8, no. 5, 1998.
[32] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[33] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the Point Cloud: From Transductive to Semi-Supervised Learning," *Proc. 22nd Int'l Conf. Machine Learning (ICML '05),* 2005.
[34] A.W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 12, pp. 1349-1380, Dec. 2000.
[35] J. Smith and S.-F. Chang, "Visualseek: A Fully Automated Content-Based Image Query System," *Proc. Fourth Ann. ACM Int'l Conf. Multimedia (Multimedia '96),* 1996.
[36] Z. Su, S. Li, and H.-J. Zhang, "Extraction of Feature Subspace for Content-Based Retrieval Using Relevance Feedback," *Proc. Ninth Ann. ACM Int'l Conf. Multimedia (Multimedia '01),* 2001.
[37] D.L. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 831-836, Aug. 1996.
[38] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 500, pp. 2319-2323, 2000.
[39] K. Tieu and P. Viola, "Boosting Image Retrieval," *Proc. Eighth Ann. ACM Int'l Conf. Multimedia (Multimedia '00),* June 2000.
[40] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," *Proc. Ninth Ann. ACM Int'l Conf. Multimedia (Multimedia '01),* pp. 107-118, 2001.
[41] V.N. Vapnik, *Statistical Learning Theory.* John Wiley & Sons, 1998.
[42] G. Wu, E.Y. Chang, and N. Panda, "Formulating Context-Dependent Similarity Functions," *Proc. 13th Ann. ACM Int'l Conf. Multimedia (Multimedia '05),* Nov. 2005.
[43] H. Yu, M. Li, H.-J. Zhang, and J. Feng, "Color Texture Moments for Content-Based Image Retrieval," *Proc. IEEE Int'l Conf. Image Processing (ICIP '02),* pp. 24-28, 2002.
[44] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," *SIAM J. Scientific Computing,* vol. 26, no. 1, 2004.

Xiaofei He received the BS degree in computer science from Zhejiang University, China, in 2000 and the PhD degree in computer science from the University of Chicago, in 2005. He is currently a research scientist at Yahoo! Research. His research interests include machine learning, information retrieval, computer vision, and multimedia.

Deng Cai received the BEng and MEng degrees in automation from Tsinghua University, China, in 2000 and 2003, respectively. He is currently working toward the PhD degree in the Department of Computer Science, University of Illinois, Urbana Champaign. His research interests include machine learning, data mining, and information retrieval. He is a student member of the IEEE.

Jiawei Han is a professor in the Department of Computer Science, University of Illinois, Urbana-Champaign. He has been working on research in data mining, data warehousing, stream data mining, spatiotemporal and multimedia data mining, biological data mining, social network analysis, text and Web mining, and software bug mining, for which he has published more than 300 conference proceedings and journal articles. He has chaired or served in many program committees of international conferences and workshops. He also served or is on the editorial boards of *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Knowledge and Data Engineering*, the *Journal of Computer Science and Technology*, and the *Journal of Intelligent Information Systems*. He is the founding Editor in Chief of the *ACM Transactions on Knowledge Discovery from Data* and is on the board of directors for the executive committee of the ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). He is a fellow of the ACM and a senior member of the IEEE. He has received many awards and recognition, including the ACM SIGKDD Innovation Award in 2004 and the IEEE Computer Society Technical Achievement Award in 2005.