Regularized Regression on Image Manifold for Retrieval*

Deng Cai
Dept. of Computer Science
UIUC
dengcai2@cs.uiuc.edu

Xiaofei He
Yahoo! Inc.
hex@yahoo-inc.com

Jiawei Han
Dept. of Computer Science
UIUC
hanj@cs.uiuc.edu

ABSTRACT

Recently, there have been considerable interests in geometric-based methods for image retrieval. These methods consider the image space as a smooth manifold and apply manifold learning techniques to find a Euclidean embedding. Thus, the Euclidean distances in the embedding space can be used as approximations to the geodesic distances on the manifold. A main advantage of these methods is that the relevance feedbacks during retrieval can be naturally incorporated into the system as prior information. In this paper, we consider the retrieval problem as a classification problem on manifold. Instead of learning a distance measure, we aim to learn a classification function on the image manifold. Considering efficiency is a key issue in image retrieval, especially on the Web scale, we propose a novel approach for image retrieval on manifold. This approach is based on a regularized linear regression framework. The local manifold structure and user-provided relevance feedbacks are incorporated into the image retrieval system through a *Locality Preserving Regularizer*. Extensive experiments are carried out on a large image database which demonstrates the efficiency and effectiveness of the proposed approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Relevance feedback*

General Terms

Algorithms, Performance, Theory

Keywords

Image Retrieval, Relevance Feedback, Regression

*The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

MIR'07, September 28–29, 2007, Augsburg, Bavaria, Germany.

1. INTRODUCTION

Content-Based Image retrieval (CBIR) has been a long standing problem in multimedia. Unlike text-based search which has achieved enormous success, the state-of-the-art CBIR systems are still far from satisfactory [24], [4], [21]. The difficulty of CBIR is essentially due to the difficulty of representing an image. The components of a text document, *i.e*., terms, can faithfully describe the topics of the document; whereas the components of an image, *i.e*., pixels, can hardly describe any semantics of the image. Let us compare the space of text documents with that of images. The document space is usually linear or close to linear. That is, if we add two document vectors together, the new document vector can still describe the topics of the two original documents to some extent. However, if we add two image vectors (pixel wise) together, the resulting image is not a naturally generated image and can no longer describe the semantics of the original two images. Mathematically, this is because of the *non-linearity* and *disconnectivity* of the pixel-based space of natural images.

Instead of pixels, visual features like color, texture, and shape have been proposed to represent images. This is referred to as *feature vector* thereafter, and the space of feature vectors is referred to as *feature space*. Since visual features cannot always describe the semantics of the images, relevance feedback is introduced as a powerful tool for soliciting information from the users [23]. During the last decade, relevance feedback has been at the core of the CBIR research. Most of the previous research on relevance feedback has fallen into the following three categories: (1) retrieval based on query point movement [22], (2) retrieval based on re-weighting of different feature dimensions [19], and (3) retrieval based on updating the probability distribution of images in the database [7]. However, all of these methods fail to take into account the geometrical structure of the feature space.

Recently there have been considerable interests in geometrically motivated approaches for image retrieval [11], [12], [13], [20], [31]. These methods consider the image feature space as a nonlinear space, particularly, manifold. He *et al*. proposed a method that finds a Euclidean embedding of the image manifold and performs image retrieval in the embedding space [13]. Lin *et al*. proposed an *Augmented Relation Embedding* method that maps the feature space into a semantic manifold which grasps the user's preferences. They construct two feedback relational graphs to incorporate the user-provided positive and negative examples [20]. Moreover, a manifold ranking-based method has been proposed to explore the relationships among all the data points in the feature space, and measure the relevances between the query and all the images in the database accordingly, which is different from traditional similarity metrics based on pair-wise distance [11]. These methods can successfully discover the intrinsic image manifold structure. However,

they aim to learn a distance measure on the image manifold, rather than a classification function on the image manifold. Therefore, they fail to make full use of the user-provided relevance feedbacks and thus may not be optimal in the sense of discriminating relevant images from irrelevant ones.

In this paper, we propose a novel approach for image retrieval on manifold. Our approach is fundamentally based on a regularized linear regression framework. There are many classification methods in literature, such as Support Vector Machine (SVM, [29]), boosting [26], Linear Discriminant Analysis (LDA, [8]), and regression [10]. Previous work has demonstrated that SVM can significantly improve retrieval performance [28][16]. However, one of the main disadvantages of SVM is its high computational complexity. It can hardly be scaled to the Web environment where the image search system needs to response to tens of thousands queries simultaneously. Therefore, we consider using the regression framework in this study. Besides its computational efficiency, another advantage of regression is that the prior information can be easily incorporated as a regularization term. Specifically, we first build an adjacency graph to model the local image manifold structure. User's relevance feedbacks are used to update the graph structure. In this way, the classifier obtained minimizes the least square error and simultaneously respects the graph structure.

The following highlights the major contributions of the paper:

1. The problem of relevance feedback image retrieval is a typical small-sample learning problem. Therefore, when regression framework is considered, a key factor is how to regularize it. The traditional solution to this is to apply an L_2 norm of the classification function which leads to minimum norm classifier, generally referred to as *ridge regression*. Such regularization term is data independent. The regularization term used in our approach explicitly takes into account the geometrical structure of the data space and makes use of the user's relevance feedbacks.

2. Most of previous manifold-based image retrieval methods consider image retrieval as a distance-based ranking problem. They suffer from the problem of how to combine the user-provided positive and negative examples. Different from them, we consider the image retrieval problem as a classification problem on manifolds which allows us to make efficient use of relevance feedbacks.

The rest of this paper is organized as follows. In Section 2, we provide a brief review of regression. We introduce our image retrieval approach on manifold in Section 3. The experimental results are presented in Section 4. Finally, we conclude the paper and provide suggestions for future work in Section 5.

2. A BRIEF REVIEW OF REGRESSION AND LOCALITY PRESERVING PROJECTION

In this section, we provide a brief review of Linear Regression and Locality Preserving Projection [14][12].

2.1 Linear Regression

Suppose we have m labeled data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{1, -1\}$. Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$. Linear regression aims to fit a function

$$f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$$

such that the residual sum of square is minimized:

$$RSS(\mathbf{a}) = \sum_{i=1}^{m} \left(f(\mathbf{x}_i) - y_i\right)^2 \tag{1}$$

For the sake of simplicity, we append a new element "1" to each $\mathbf{x}_i$. Thus, the coefficient b can be absorbed into $\mathbf{a}$ and we have $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$. Let $\mathbf{y} = [y_1, \cdots, y_m]^T$. We have

$$RSS(\mathbf{a}) = \left(\mathbf{y} - X^T\mathbf{a}\right)^T\left(\mathbf{y} - X^T\mathbf{a}\right)$$

Requiring $\partial RSS(\mathbf{a})/\partial\mathbf{a} = 0$, we obtain:

$$\mathbf{a} = (XX^T)^{-1}X\mathbf{y} \tag{2}$$

It would be important to note that, in image retrieval, the number of labeled samples is often smaller than the number of features. Thus, the matrix XX^T is singular and the problem (1) is *ill posed*. A possible solution is to impose a penalty on the norm of $\mathbf{a}$:

$$RSS_{ridge}(\mathbf{a}) = \sum_{i=1}^{m} \left(y_i - \mathbf{a}^T\mathbf{x}_i\right)^2 + \lambda\|\mathbf{a}\|^2 \tag{3}$$

The solution to (3) is given below:

$$\mathbf{a} = \left(XX^T + \lambda I\right)^{-1}X\mathbf{y} \tag{4}$$

where I is a $n \times n$ identity matrix. It is clear to see that $XX^T + \lambda I$ is no longer singular. The term $\|\mathbf{a}\|^2$ in Eq. (3) is called Tikhonov regularizer [27]. In statistics, such regression is called *ridge regression* [10]. The Tikhonov regularizer $\|\mathbf{a}\|^2$ is data independent. It fails to discover the intrinsic geometrical structure of the feature space and the semantic relationship between images.

2.2 Locality Preserving Projection

Given m data points $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^n$, LPP uses a p-nearest neighbor graph to model the local geometrical structure in the data. Specifically, we put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close", *i.e.*, $\mathbf{x}_i$ and $\mathbf{x}_j$ are among p nearest neighbors of each other. Let W denote the corresponding weight matrix, the objective function of LPP is as follows [14]:

$$\begin{aligned}\mathbf{a}_{opt} &= \arg\min_{\mathbf{a}^T X D X^T \mathbf{a}=1} \sum_{ij} \left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W_{ij} \\ &= \arg\min_{\mathbf{a}^T X D X^T \mathbf{a}=1} \mathbf{a}^T X L X^T \mathbf{a}\end{aligned} \tag{5}$$

where $L = D - W$ is the *graph Laplacian* [5] and $D_{ii} = \sum_j W_{ij}$. The constraint $\mathbf{a}^T X D X^T \mathbf{a} = 1$ removes an arbitrary scaling factor in the embedding [1]. Such a objective function incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart.

The projection vector $\mathbf{a}$ that minimizes the objective function is given by the *minimum* eigenvalue solution to the generalized eigenvalue problem:

$$XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \tag{6}$$

Unlike ISOMAP which preserves global geometrical properties like geodesics, LPP preserves local geometrical structure. Specifically, LPP aims to preserve local distances and find a linear approximation to the manifold which can best preserve the local isometry. LPP has been successfully used in relevance feedback image retrieval.

3. REGULARIZED REGRESSION ON IMAGE MANIFOLD

In this section, we introduce our image retrieval approach on manifold. We begin with a formal description of the problem.

3.1 The Problem

The generic problem of image retrieval is the following. Given a query image $\mathbf{q}$ and an image database[1] $\{\mathbf{x}_i\}_{i=2}^m$, find a function f such that $f(\mathbf{x})$ reflects the semantic relationship between $\mathbf{x}$ and $\mathbf{q}$. The typical relevance feedback based retrieval process is outlined as follows.

1. The user provides his relevance feedback to the system by labeling images as "relevant" or "irrelevant".
2. The system modifies f using the feedbacks.
3. The system re-ranks the images and present the top ones to the user.

The most typical f is defined by using a distant measure [13],

$$f(\mathbf{x}) = dist(\mathbf{x}, \mathbf{q})$$

Another possible choice is to consider f as a classification function [28]. Specifically, $f(\mathbf{x}) > 0$ if $\mathbf{x}$ is relevant to $\mathbf{q}$ and $f(\mathbf{x}) < 0$ otherwise.

When we consider f as a classification function, the user provided positive and negative examples are used as training samples. Suppose we have a set of feedback samples, $\{(\mathbf{x}_i, y_i)\}_{i=2}^l$, where y_i is the label of $\mathbf{x}_i$ marked by the user:

$$y_i = \begin{cases} 1, & \mathbf{x}_i \text{ is relevant to the query image;} \\ -1, & \mathbf{x}_i \text{ is irrelevant to the query image.} \end{cases}$$

The label of the query image $\mathbf{q}$ can be regarded as 1. For simplicity, we denote the query image as $(\mathbf{x}_1, y_1)$. Thus, one tries to find a linear function $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ such that some predefined loss function $\mathcal{L}(\mathbf{a})$ is minimized. The most typical loss function is the least square error in Eq. (1). Some other popular loss functions includes the hinge loss used in SVM [29]:

$$\mathcal{L}(\mathbf{a}) = \sum_{i=1}^{l} \left(1 - y_i \mathbf{a}^T \mathbf{x}_i\right)_+$$

$$\left(1 - y_i \mathbf{a}^T \mathbf{x}_i\right)_+ = \begin{cases} 1 - y_i \mathbf{a}^T \mathbf{x}_i, & \text{if } y_i \mathbf{a}^T \mathbf{x}_i \leq 1; \\ 0, & \text{otherwise.} \end{cases}$$

and the logistic loss used in Logistic Regression [10]:

$$\mathcal{L}(\mathbf{a}) = \sum_{i=1}^{l} \log\left(1 + exp(-y_i \mathbf{a}^T \mathbf{x}_i)\right)$$

In this paper, we adopt the least square loss function for its simplicity and effectiveness.

Most of previous classification based methods consider image retrieval as a supervised learning problem such that f is obtained by only using the training samples. However, image retrieval is intrinsically a semi-supervised learning problem in that the testing samples (images in the database) are available during the training process [6], [17], [25]. Naturally, an optimal classification function should take into account the distribution of the testing samples. In the next subsection, we introduce a novel image retrieval approach based on semi-supervised learning on image manifold. Our approach is fundamentally based on previous work on manifold learning [1], [14] and manifold regularization [2].

[1]For convenience, we will treat $\mathbf{q}$ as $\mathbf{x}_1$ in the later description.

3.2 Learning A Retrieval Function with Locality Preserving Regularizer

Given the image retrieval problem as described in the last subsection, we define:

$$\mathbf{y} = [y_1, y_2, \cdots, y_l]^T$$

$$X_1 = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l]$$

$$X_2 = [\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \cdots, \mathbf{x}_m]$$

$$X = [X_1, X_2]$$

where X_1 are the query image and feedback images, $\mathbf{y}$ is the label vector and X_2 are the remaining images in the database.

We aim to learn a function f such that $f(\mathbf{x}) > 0$ if $\mathbf{x}$ is relevant to $\mathbf{q}$ and $f(\mathbf{x}) < 0$ otherwise. In the situations when there is no sufficient training samples comparing to the number of features, such as image retrieval, one is often confronted with the overfitting problem. In order to overcome this problem and increase the generalization capability of the classifier, one hopes to make the function f as smooth as possible, based on the assumption that close points should have similar semantics. This suggests the following two general principles for learning an image retrieval function:

Principal 1 $f(\mathbf{x}_i) = y_i$, $i = 1, \cdots, l$.

Principal 2 If $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other, $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ are also close to each other.

Principal 1 can be formulated as a least square cost function:

$$\phi_1(f) = \sum_{i=1}^{l} \left(y_i - f(\mathbf{x}_i)\right)^2 \tag{7}$$

For principal 2, we use a p-nearest neighbor graph G to capture local geometrical structure in the data. Specifically, we put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close", *i.e.*, $\mathbf{x}_i$ and $\mathbf{x}_j$ are among p nearest neighbors of each other. Thus, principal 2 can be formulated as a locality preserving cost function:

$$\phi_2(f) = \sum_{i,j=1}^{m} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right)^2 W_{ij} \tag{8}$$

where W is the weight matrix of the p-nearest neighbor graph G. Such a cost function incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. The cost function (8) and its linearization are originally introduced in [1], [14]. The optimal f can be obtained by solving the following optimization problem [2]:

$$\begin{aligned} f^* &= \arg\min_f \phi(f) = \arg\min_f \left(\phi_1(f) + \lambda\phi_2(f)\right) \\ &= \arg\min_f \left(\sum_{i=1}^{l} \left(y_i - f(\mathbf{x}_i)\right)^2 + \right. \\ &\qquad \left. \lambda \sum_{i,j=1}^{m} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right)^2 W_{ij} \right) \end{aligned} \tag{9}$$

We will describe in details how to construct the weight matrix W in the next subsection. The locality preserving cost function $\phi_2(f)$ can also be regarded as a regularization term. Thus, regression with such a regularization term can be called Locality Preserving Regularized Regression (LPR Regression).

Consider a linear map, *i.e.*, $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$, we have:

$$\begin{aligned} & \phi_1(f) \\ = & \sum_{i=1}^{l} \left(y_i - \mathbf{a}^T\mathbf{x}_i\right)^2 \\ = & \left(\mathbf{y} - X_1^T\mathbf{a}\right)^T\left(\mathbf{y} - X_1^T\mathbf{a}\right) \\ = & \mathbf{y}^T\mathbf{y} - 2\mathbf{a}^T X_1\mathbf{y} + \mathbf{a}^T X_1 X_1^T\mathbf{a} \end{aligned}$$

Likewise, $\phi_2(f)$ can be reduced to:

$$\begin{aligned} & \phi_2(f) \\ = & \sum_{i,j=1}^{m} \left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W_{ij} \\ = & 2\mathbf{a}^T\Big(\sum_{ii} D_{ii}\mathbf{x}_i\mathbf{x}_i^T - \sum_{ij} W_{ii}\mathbf{x}_i\mathbf{x}_j^T\Big)\mathbf{a} \\ = & 2\mathbf{a}^T(XDX^T - XWX^T)\mathbf{a} \\ = & 2\mathbf{a}^T XLX^T\mathbf{a} \end{aligned}$$

where $L = D - W$ is the Laplacian matrix[2] and D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$.

Note that, $\mathbf{y}_1^T\mathbf{y}_1$ is a constant. Thus, the final cost function $\phi(f)$ can be written as follows:

$$\phi(\mathbf{a}) = -2\mathbf{a}^T X_1\mathbf{y} + \mathbf{a}^T X_1 X_1^T\mathbf{a} + 2\lambda\mathbf{a}^T XLX^T\mathbf{a} \qquad (10)$$

Requiring the derivative of $\phi(\mathbf{a})$ with respect to $\mathbf{a}$ vanish, we get:

$$\frac{\partial\phi(\mathbf{a})}{\partial\mathbf{a}^T} = 0 \qquad (11)$$

$$\Rightarrow \quad -X_1\mathbf{y} + X_1X_1^T\mathbf{a} + 2\lambda XLX^T\mathbf{a} = 0 \qquad (12)$$

$$\Rightarrow \quad \mathbf{a} = \left(X_1X_1^T + 2\lambda XLX^T\right)^{-1} X_1\mathbf{y} \qquad (13)$$

3.3 The Algorithm

Given a query image $\mathbf{q}$ and an image database $\{\mathbf{x}_2, \cdots, \mathbf{x}_m\} \subset \mathbb{R}^n$. Suppose the user provides the label information of the top $l-1$ images, we get $(\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)$, where y_i is the label of $\mathbf{x}_i$ marked by the user:

$$y_i = \begin{cases} 1, & \mathbf{x}_i \text{ is relevant to the query image;} \\ -1, & \mathbf{x}_i \text{ is irrelevant to the query image.} \end{cases}$$

The label of $\mathbf{q}$ can be regarded as 1. For simplicity, we denote the query image as $(\mathbf{x}_1, y_1)$. Let $X_1 = (\mathbf{x}_1, \cdots, \mathbf{x}_l) \in \mathbb{R}^{n\times l}$, $\mathbf{y} = (y_1, \cdots, y_l)^T \in \mathbb{R}^l$ and $X = (\mathbf{x}_1, \cdots, \mathbf{x}_m) \in \mathbb{R}^{n\times m}$.

The algorithmic procedure of Regression with Locality Preserving Regularizer (LPR Regression) is stated below:

1. **Constructing the adjacency graph**: Let G denote a graph with m nodes. The i-th node corresponds to the image $\mathbf{x}_i$.

[2]The Laplacian matrix $L(= D - W)$ for finite graph, or *graph Laplacian* [5], [9], is analogous to the Laplace Beltrami operator on compact Riemannian manifolds. While the Laplace Beltrami operator for a manifold is generated by the Riemannian metric, for a graph it comes from the adjacency relation. In manifold learning, the graph Laplacian is very important since the graph can be build to model the local structure of the data and the Laplacian (discrete approximation to the Laplace Beltrami operator) can capture the intrinsic geometric structure of the data. A lot of algorithms had been developed based on the graph Laplacian. In fact, both PCA and LDA can can be interpreted as spectral dimensionality reduction methods with different graph Laplacian (different graph structure). Please see [15] for more details.

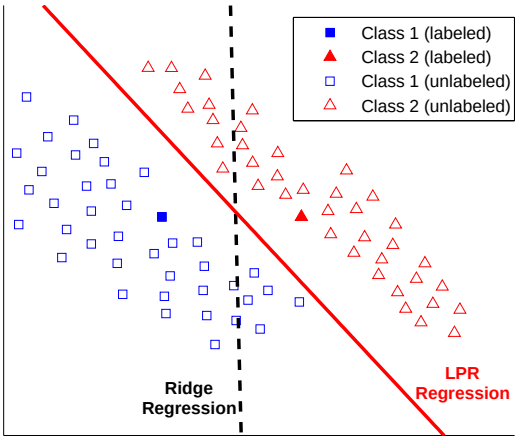


Figure 1: A comparison of Ridge Regression and LPR Regression for a two class problem where data are linearly separable. There is only one labeled example for each class. LPR Regression considers the geometrical structure of the whole dataset (labeled and unlabeled) and produces a high generalization capability function. Ridge Regression only considers the labeled information and fails to produce a satisfactory function.

We construct the graph G through the following three steps to model the local structure as well as the label information (user's feedback):

   (a) Put an edge between nodes i and j if $\mathbf{x}_i$ is among p nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among p nearest neighbors of $\mathbf{x}_i$.

   (b) Put an edge between nodes i and j if $\mathbf{x}_i$ shares the same label with $\mathbf{x}_j$.

   (c) Remove the edge between nodes i and j if the label of $\mathbf{x}_i$ is different with that of $\mathbf{x}_j$.

2. **Choosing the weights**: W is a sparse symmetric $m \times m$ matrix with W_{ij} having the weight of the edge joining vertices i and j.

   (a) If there is no edge between i and j, $W_{ij} = 0$.

   (b) Otherwise,

$$W_{ij} = \begin{cases} 1, & \mathbf{x}_i \text{ shares the same label with } \mathbf{x}_j. \\ \frac{\mathbf{x}_i^T\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_i\|}, & \text{otherwise.} \end{cases}$$

   The weight matrix W of graph G models the local structure of the image space as well as relevance information provided by the user.

3. Solve the linear equations:

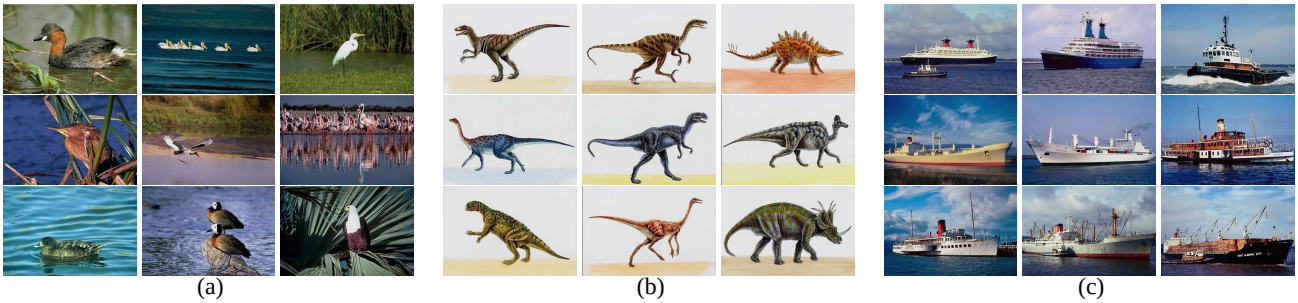
$$(X_1X_1^T + \lambda XLX^T)\mathbf{a} = X_1\mathbf{y} \qquad (14)$$

   which gives us the classification function:

$$\mathbf{a} = (X_1X_1^T + \lambda XLX^T)^{-1}X_1\mathbf{y} \qquad (15)$$

   where $L = D - W$ is the Laplacian matrix and D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$.

4. Re-rank the image database $\{\mathbf{x}_2, \cdots, \mathbf{x}_m\}$ by $y_i = \mathbf{a}^T\mathbf{x}_i$.



(a) (b) (c)

Figure 2: Sample images from category (7) *bird*, (23) *dinosaur* and (66) *ship*

4. EXPERIMENTS AND DISCUSSIONS

We performed several experiments to evaluate the effectiveness of the proposed algorithm on a large image database.

4.1 Image Dataset and Features

The image database we used consists of 7,900 images of 79 semantic categories, from COREL data set. It is a large and heterogeneous image set. Figure 2 shows some sample images.

We combined 64-dimensional color histogram and 64-dim Color Texture Moment (CTM) [30] to represent each image. The color histogram is calculated using $4\times4\times4$ bins in HSI space. The Color Texture Moment (CTM) is proposed by Yu *et al*. [30], which integrates the color and texture characteristics of an image in a compact form. CTM adopts local Fourier transform as a texture representation scheme and derive eight characteristic maps for describing different aspects of co-occurrence relations of image pixels in each channel of the ($SV\cos H$, $SV\sin H$, 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 [30] for details.

4.2 Experimental Design

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 below.

4.2.1 Evaluation Metrics

To evaluate the effectiveness of an algorithm, we use both the *precision-scope curve* and the *precision rate* [18]. In our context, the scope specified by the number N, of top-ranking images returned in response to the user's query. The precision is the ratio of relevant images number to the scope N. A precision-scope curve records the precision over a range of scopes and can evaluate the overall performance of an algorithm. On the other hand, the precision rate emphasizes the precision for 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 presentation. Therefore, the retrieval performance at top 20 (the precision rate at scope 20) is especially important.

Besides precision, efficiency is also a key issue in image retrieval, especially when web scale is concerned. Thus we record and compare the running times of different algorithms.

4.2.2 Five-fold Cross Validation

In a real image retrieval system, a query image is usually not in the image database. To simulate such environment, we use five-fold cross validation to evaluate all the algorithms. More precisely, we divide the whole image database into five equal-size sets and there are 20 images per category in each set. In each run of cross validation, one set is picked as the query set, and the other four sets are left as database set. The precision-scope curve and precision rate are derived by averaging the results from the five runs of cross validation.

4.2.3 Relevance Feedback Scheme

For each submitted query, our system retrieves and ranks the images in the database set. The top 10 ranked images were selected as the feedback images, and their label information (relevant or irrelevant) are used for re-ranking. Note that the images have been selected in the previous iterations are excluded from later selections. And, with each query, the feedback mechanism is carried out for four iterations.

It is important to note that the relevance feedback scheme used here is different from the automatic feedback scheme described in [12], [20]. In [12], [20], the top four relevant and irrelevant images were selected as the feedback images. In reality, such scheme might be impossible. It is more reasonable for the users to provide feedback information on the first screen shot (10 or 20 images). However, the system can not guarantee there are at least four relevant images in these top ranked images.

4.2.4 Compared Algorithms

To demonstrate the effectiveness of our proposed algorithm, we compared the following four algorithms.

RidgeReg : The ridge regression algorithm described in Section 2. The classification function $\mathbf{w}$ is the solution of linear equations $(X_1X_1^T+\lambda I)\mathbf{w}=X_1\mathbf{y}$, where $X_1=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_l]$ are the query image vector $\mathbf{x}_1$ and $l-1$ feedback image vectors. $\mathbf{y}=[y_1,y_2,\cdots,y_l]^T$ are the feedback information, $y_i=1$ for relevant images and $y_i=0$ for irrelevant images. The *back slash* operator in Matlab is used in our system to solve the equation[3]. The parameter λ was set to 0.1.

LPRReg : Different from the ridge regularizer which is data independent, the locality preserving regularizer can capture the intrinsic geometrical structure of the feature space and the semantic relationship between images. Similarly, the classification function $\mathbf{a}$ is the solution of linear equations $(X_1X_1^T+\lambda XLX^T)\mathbf{a}=X_1\mathbf{y}$, where

$$X=(\mathbf{x}_1,\cdots,\mathbf{x}_l,\mathbf{x}_{l+1},\cdots,\mathbf{x}_m)$$

includes both labeled (feedback) images and unlabeled images in database. L is graph Laplacian defined on X as described in Section 3 and we set the parameter $p=5$. In

[3]The *back slash* operator in Matlab solve the equation by Gaussian elimination with partial pivoting, which is very efficient.

Table 1: Precision at top 20 returns of the four algorithms after the first feedback iteration

	P@20 after the first feedback iteration (%)						P@20 after the first feedback iteration (%)				
	Baseline	RidgeReg	LPRReg	SVM	ARE		Baseline	RidgeReg	LPRReg	SVM	ARE
1	15.25	21.00	**25.15**	21.05	19.10	41	74.10	76.25	**84.95**	78.30	79.85
2	41.20	38.70	**61.40**	41.95	48.30	42	44.55	32.90	**57.35**	34.80	52.65
3	12.45	20.00	**20.45**	20.00	14.05	43	6.60	11.10	10.15	**11.55**	6.85
4	21.35	33.90	**35.90**	34.15	23.45	44	93.20	90.70	94.70	93.55	**97.70**
5	11.35	16.80	17.85	**17.90**	13.05	45	27.70	23.15	**33.50**	25.25	32.55
6	14.55	18.10	**23.85**	18.85	18.25	46	8.75	12.10	**13.10**	12.65	12.15
7	5.30	8.50	**9.55**	8.20	7.15	47	27.90	29.80	**37.95**	30.10	34.10
8	27.75	33.75	**43.90**	36.50	33.55	48	23.40	24.80	**32.00**	26.20	28.85
9	29.65	38.60	**50.05**	39.95	36.25	49	16.25	33.10	**36.60**	34.40	23.15
10	7.60	11.75	**15.15**	11.70	11.30	50	42.85	39.35	**52.95**	39.75	48.70
11	31.35	41.15	**47.75**	41.90	35.90	51	8.75	18.10	**18.10**	17.85	12.40
12	22.15	38.00	**41.30**	38.05	25.70	52	15.45	20.60	**23.85**	21.45	21.70
13	9.30	15.45	**16.55**	16.50	12.20	53	59.40	61.05	**70.80**	63.25	64.45
14	33.05	43.70	**51.35**	46.15	41.60	54	32.00	32.85	**44.60**	34.30	38.55
15	85.55	92.50	**93.55**	91.80	89.65	55	36.25	46.00	**58.30**	46.00	47.45
16	10.70	16.55	**19.55**	17.25	15.00	56	20.35	36.85	35.20	**37.60**	20.95
17	21.10	33.45	**40.20**	34.00	24.30	57	73.65	70.05	73.85	71.05	**77.65**
18	15.30	18.00	**24.35**	18.20	18.80	58	16.30	25.00	**27.55**	25.25	23.85
19	36.05	48.90	**51.95**	49.50	37.70	59	8.25	16.90	16.25	**17.00**	9.25
20	12.60	15.95	**19.65**	16.95	17.30	60	73.85	86.80	**92.80**	88.15	77.20
21	7.50	10.75	**12.25**	11.35	9.00	61	38.15	54.30	**62.70**	55.85	50.65
22	55.80	53.40	**62.50**	55.80	60.65	62	31.30	39.95	**49.60**	41.45	38.65
23	88.20	92.85	**97.45**	95.75	94.30	63	10.90	**26.85**	26.05	26.30	16.05
24	69.20	66.85	**82.40**	71.60	79.70	64	42.50	45.80	**61.30**	50.20	49.70
25	7.95	12.70	**14.70**	13.05	10.15	65	12.65	21.00	**22.35**	21.15	17.60
26	46.70	41.55	**51.50**	42.95	49.60	66	30.60	38.05	**41.85**	38.50	38.20
27	31.25	48.85	**55.90**	49.70	39.50	67	17.25	**27.75**	27.05	27.10	20.75
28	24.90	40.90	**46.65**	41.45	31.50	68	19.42	38.12	**39.48**	38.18	25.97
29	82.04	69.37	80.99	72.11	**82.61**	69	44.35	41.50	**59.20**	49.30	47.70
30	32.30	33.00	**43.45**	34.55	38.60	70	25.20	52.30	54.75	**55.20**	34.15
31	42.00	54.55	**63.05**	54.00	47.90	71	31.80	37.80	**48.05**	40.60	38.15
32	83.45	77.50	88.65	80.65	**88.90**	72	36.00	26.75	34.10	29.35	**42.60**
33	68.30	72.15	**87.45**	73.90	71.75	73	38.75	52.70	**62.30**	56.65	51.75
34	25.85	41.60	44.35	**44.60**	25.95	74	19.90	24.30	**27.85**	25.60	24.40
35	10.85	16.35	**17.75**	16.20	13.50	75	33.20	34.75	**46.00**	38.20	36.30
36	9.65	11.60	**15.05**	11.95	12.50	76	13.30	25.90	**26.95**	26.15	16.55
37	34.20	44.75	**56.90**	48.00	39.95	77	18.00	20.55	**26.90**	22.95	25.10
38	11.00	14.40	**17.95**	14.70	15.50	78	27.00	38.10	**44.40**	38.70	35.25
39	15.60	20.85	**23.10**	21.65	20.85	79	20.90	17.40	25.25	20.70	**26.90**
40	48.20	37.25	52.60	41.05	**54.90**						

our implementation, the value of m is set to 300, *i.e.*, the top ranked 300 images (labeled and unlabeled) plus the query image are used to estimate the image manifold for a particular query.

Also, the *back slash* operator in Matlab is used in our system to solve the equation. The parameter λ is set to 0.1, and the effect of parameter selection will be discussed later.

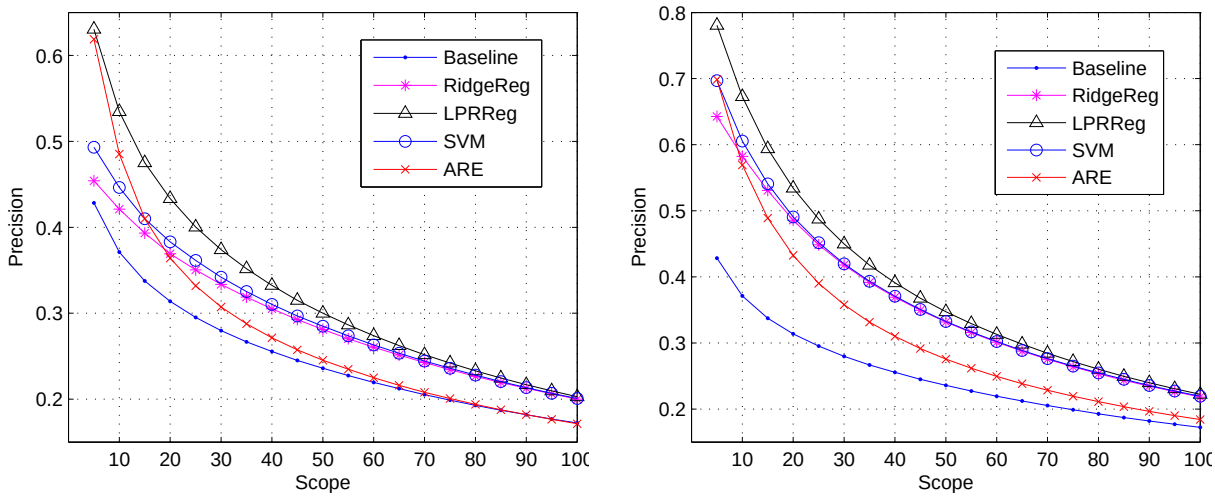
SVM : The Support Vector Machines approach. The labeled images $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ are used to learn the classification function by SVM. The LIBSVM system [3] was used in our system to solve the SVM optimization problem. Cross-validation on the labeled images is used to select the parameters in SVM.

ARE : *Augmented Relation Embedding*, which was proposed by Lin *et. al.* [20]. ARE tries to map the feature space into a semantic manifold that grasps the user's preferences. ARE constructs two feedback relational graphs to incorporate the user provided positive and negative examples. In the comparison experiment reported in [20], ARE is superior than Locality Preserving Projection in the incremental semi-supervised mode [12]. Based on the consideration of efficiency, the top 300 ranked images plus the query image are used to estimate the image manifold for a particular query.

Different from the previous three methods which learn a classification function, ARE tries to learn a subspace in which the Euclidean distance can better reflect the semantic structure of images. A crucial problem of ARE is how to determine the dimensionality of the subspace. In our experiments, we iterate all the dimensions and select the dimension with respect to the best performance. However, in real world image retrieval applications, one has to estimate the dimensionality.

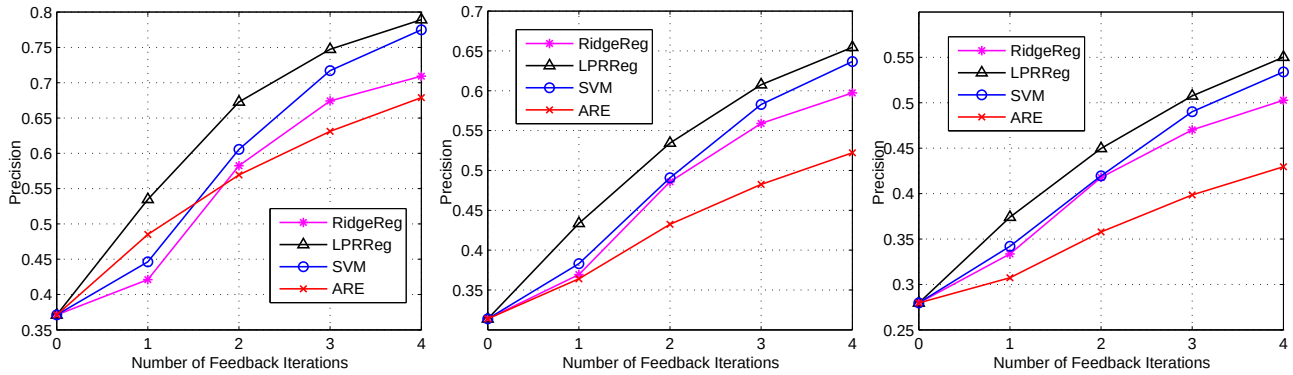
There are two interesting points in these four compared algorithms.

1. The first three (RidgeReg, LPRReg and SVM) are classification based methods. These algorithms try to learn a classification function f and re-rank the image database by this function. The last one (ARE) is a subspace learning method. The images in the database are re-ranked by their distances with the query image in the subspace. If we want to re-rank



(a) Feedback Iteration 1 (b) Feedback Iteration 2

Figure 3: The average *precision-scope* curves of different algorithms for the first two feedback iterations. The LPRReg is the best algorithm on the entire scope.



(a) Precision at Top 10 (b) Precision at Top 20 (c) Precision at Top 30

Figure 4: Performance evaluation of the four learning algorithms in learning the semantic concepts from the feedbacks. (a) Precision at top 10, (b) Precision at top 20 and (c) Precision at top 30.

the whole image database, apparently, the latter approach will be much more time consuming and can hardly scale to a very large image database.

2. Two of the four algorithms (RidgeReg and SVM) are supervised and only consider the labeled images. In image retrieval, the number of labeled images is usually very small, especially in the first round of feedback. The other two algorithms (LPRReg and ARE) are semi-supervised. They consider both labeled images and unlabeled images. Those large amount of unlabeled images can help to reveal the intrinsic geometrical structure of the feature space and the semantic relationship between images. Thus, these two semi-supervised algorithms are expected to achieve reasonable performance even with a small amount of labeled images.

4.3 Image Retrieval Performance

Table (1) shows the precision at top 20 of the first feedback iteration for all the 79 categories. The *Baseline* indicates the initial result without feedback information. The retrieval performances of all the algorithms vary with the 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 we used in our experiments are color and texture features, those categories with similar color and texture (*e.g.*, category 23 in Figure 2(b)) will have good retrieval performance. While those categories with different color and texture (*e.g.* category 7 in Figure 2(a)) tend to have poor retrieval performance. Among all the 79 categories, our LPRReg is the best for 64 categories and ranked No.2 for the remaining 15 categories.

Figure 3 shows the average *precision-scope* curves of different algorithms for the first two feedback iterations. By iteratively adding the user's feedback, the corresponding precision results (at top 10, top 20 and top 30) of the four algorithms are respectively shown in Figure 4. The running time of different algorithms for each query are also shown in Table 2. We would like to highlight several points on these results.

- Our algorithm LPRReg achieves the highest precision on all the feedback iterations and on the whole scope. The reason is that LPRReg learns a classification function which is powerful in the sense of discriminating between relevant images

Table 2: Average running time of different algorithms on processing one query

	Time on different feedback iteration (s)				
	1	2	3	4	5
RidgeReg	0.005	0.005	0.005	0.006	0.006
LPRReg	0.052	0.056	0.059	0.059	0.062
SVM	0.058	0.068	0.077	0.084	0.091
ARE	0.094	0.095	0.096	0.097	0.098

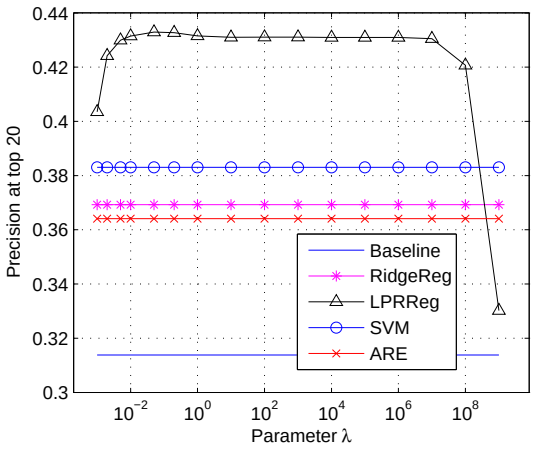


Figure 5: The performance of LPR regression vs. the parameter λ. The LPR regression is very stable with respect to the parameter λ. The performance is almost the same with the λ from 10^{-2} to 10^{7}.

and irrelevant images. Meanwhile, LPRReg explicitly considers the intrinsic geometrical structure of image manifold, which guarantee the best performance even with only a very small number of labeled images.

- The discriminating power between relevant images and irrelevant images is the key of a classification function for image retrieval. A classifier with good discriminating power on unlabeled images is expected. However, with a small number of labeled images, the learned classifier might not be good for the test (unlabeled) images, which is known as the generalization capability of a classifier. In such situation, regularization is needed. SVM can be thought of as a classifier with a large margin regularizer [10]. Although such regularizer is quite powerful with a small number of training data, it fails to consider the intrinsic image manifold structure which can be revealed by the unlabeled images.

- At the first feedback iteration, the result of ARE is quite good, especially the precision on top 5 and top 10. The reason is that the number of labeled images is pretty small in the first feedback round. ARE considers both labeled and unlabeled image while the supervised algorithms (RidgeReg and SVM) only consider the labeled images. In the later feedback iterations, the feedback information is accumulated and the classifier approaches give better results. ARE aims at learning a distance measure on the image manifold, and thus may not be optimal in the sense of discriminating between relevant images and irrelevant images.

- Considering efficiency is a key issue in image retrieval, especially when web scale is concerned, classification framework rather than subspace learning framework might be a better choice. Also, regression framework could be more competitive than SVM.

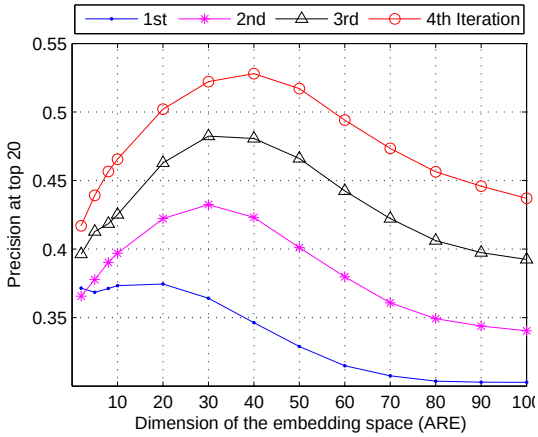


Figure 6: The performance of ARE vs. the dimensionality. The best performance in difference feedback iterations appears at different dimensions, which makes hard to estimate the intrinsic dimensionality in reality.

4.4 Parameter Selection

In the LPR regression, There is a parameter λ to adjust the weight of the regularization part. In our previous experimental results, we empirically set the λ as 0.1. In this subsection, we try to examine the stability of LPR regression with respect to the parameter λ. Figure 5 shows the performance changing of LPR regression at the first feedback iteration with the parameter λ. We can see that the LPR regression is very stable. The performance is almost the same with the λ from 10^{-2} to 10^{7}.

Different from regression or SVM, the ARE algorithm is a subspace learning algorithm. For ARE, there is a problem of how to determine the dimensionality or the subspace. Figure 6 shows the performance changing of ARE at different feedback iterations with the reduced dimensionality. We see that the best performances at different feedback iterations appear at different dimensional subspaces, which makes hard to estimate the dimensionality of the reduced space in reality.

5. CONCLUSIONS AND FUTURE WORK

A novel image retrieval approach on manifold is proposed in this paper. Our retrieval method is based on learning a classification function on the image manifold. Our method has two major advantages: as a classification based method, our method is more efficient than subspace learning based methods; second, our method explicitly takes into consideration the geometrical structure of the image manifold by making use of both labeled and unlabeled images. Several experiments demonstrate the efficiency and effectiveness of our method.

Due to the efficiency consideration, the algorithm proposed in our paper is linear. Thus it may fail to capture those complex image manifolds which are highly non-linear. However, it is easy to extend our algorithm to Reproducing Kernel Hilbert Space (RKHS) which leads to Locality Preserving Regularized Kernel Regression.

We have noticed that there is a trend in many areas, including

multimedia information retrieval, computer vision, pattern recognition, and data mining, that the data points are considered as drawn from sampling a probability distribution that has support on or near a submanifold of Euclidean space. Manifold based techniques have shown its superiority to Euclidean based techniques for CBIR by several researchers [13], [20], [11]. All of these algorithms (including the one presented in this paper) have a implicit assumption that the data points are *uniformly* sampled from the image manifold. If the data points are not uniformly sampled from the manifold, the graph Laplacian may not converge to the true Laplace Beltrami operator on the manifold. However, the real world image database might be much more complicated and far from uniform. We are currently exploring this problem in theory and practice.

6. REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, 2001.

[2] M. Belkin, P. Niyogi, and V. Sindwani. On manifold regularization. In *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[4] E. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):26–38, January 2003.

[5] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997.

[6] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1553–1567, 2004.

[7] 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 Transactions on Image Processing*, 9:20–37, 2000.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.

[9] S. Guattery and G. L. Miller. Graph embeddings and laplacian eigenvalues. *SIAM Journal on Matrix Analysis and Applications*, 21(3):703–723, 2000.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[11] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *Proceedings of the ACM Conference on Multimedia*, New York, October 2004.

[12] X. He. Incremental semi-supervised subspace learning for image retrieval. In *Proceedings of the ACM Conference on Multimedia*, New York, October 2004.

[13] X. He, W.-Y. Ma, and H.-J. Zhang. Learning an image manifold for retrieval. In *Proceedings of the ACM Conference on Multimedia*, New York, October 2004.

[14] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2003.

[15] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[16] C.-H. Hoi and M. R. Lyu. A novel log-based relevance feedback technique in content-based image retrieval. In *Proceedings of the ACM Conference on Multimedia*, New York, October 2004.

[17] S. Hoi and M. Lyu. A semi-supervised active learning framework for image retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'05)*, 2005.

[18] D. P. Huijsmans and N. Sebe. How to complete performance graphs in content-based image retrieval: Add generality and normalize scope. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):245–251, 2005.

[19] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Query databases through multiples examples. In *Proc. 24th International Conference on Very Large Databases*, pages 218–227, New York, 1998.

[20] Y.-Y. Lin, T.-L. Liu, and H.-T. Chen. Semantic manifold learning for image retrieval. In *Proceedings of the ACM Conference on Multimedia*, Singapore, November 2005.

[21] W.-Y. Ma and B. S. Manjunath. Netra: a toolbox for navigating large image databases. *Multimedia Systems*, 7(3), May 1999.

[22] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in mars. In *IEEE Conference on Image Processing*, pages 815–818, Santa Barbara, CA, Oct. 1997.

[23] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[24] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[25] Q. Tian, J. Yu, Q. Xue, and N. Sebe. A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval. In *IEEE Int. Conf. on Multimedia and Expo (ICME'04)*, 2004.

[26] K. Tieu and P. Viola. Boosting image retrieval. In *Proceedings of the ACM Conference on Multimedia*, Hilton Head Island, SC, June 2000.

[27] A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math.*, (4), 1963 (English Translation).

[28] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, 2001.

[29] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[30] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moments for content-based image retrieval. In *International Conference on Image Processing*, pages 24–28, 2002.

[31] J. Yu and Q. Tian. Learning image manifolds by semantic subspace projection. In *Proceedings of the ACM Conference on Multimedia*, Santa Barbara, October 2006.