Spectral Regression: A Unified Subspace Learning Framework for Content-Based Image 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

Relevance feedback is a well established and effective framework for narrowing down the gap between low-level visual features and high-level semantic concepts in content-based image retrieval. In most of traditional implementations of relevance feedback, a distance metric or a classifier is usually learned from user's provided negative and positive examples. However, due to the limitation of the user's feedbacks and the high dimensionality of the feature space, one is often confront with the issue of the *curse of the dimensionality*. Recently, several researchers have considered manifold ways to address this issue, such as Locality Preserving Projections, Augmented Relation Embedding, and Semantic Subspace Projection. In this paper, by using techniques from spectral graph embedding and regression, we propose a unified framework, called *spectral regression*, for learning an image subspace. This framework facilitates the analysis of the differences and connections between the algorithms mentioned above. And more crucially, it provides much faster computation and therefore makes the retrieval system capable of responding to the user's query more efficiently.

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, Dimensionality Reduction, Manifold Learning, Subspace Learning, Spectral Regression

1. INTRODUCTION

Content-Based Image Retrieval (CBIR) has attracted substantial interests as the volumes of image data have grown rapidly during the last decade [9, 10, 15, 21, 22, 27, 28]. It is well known that one of the most challenging problems in CBIR is to bridge the semantic gap between low-level visual features and high-level semantic concepts. One feasible way to address this problem is through learning from the user's relevance feedback [21].

In real world image retrieval systems, the relevance feedbacks provided by the user is often limited, typically less than 20, while 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 [8]. 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 (or, subspace learning) algorithms includes Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA projects the data points into a lower dimensional subspace in which the sample variance is maximized while LDA finds projective directions by maximizing the ratio of between-class scatter to within-class scatter. Both PCA and LDA have been widely applied to image retrieval, face recognition, and pattern recognition. However, PCA is unsupervised thus cannot utilize the relevance feedback provided by the user. LDA is supervised, but it is hard to learn a function with good generalization capability with a small number of labeled examples (feedbacks) [8].

To this end, various researchers have considered the dimensionality reduction problem in semi-supervised situation. With both unlabeled and labeled images (relevance feedbacks), one hopes to find a better subspace for image representation. In this subspace, the semantic structure of the image data can be better revealed. The state-of-the-art semi-supervised subspace learning algorithms in CBIR are incremental Locality Preserving Projection (LPP) [10], Augmented Relation Embedding (ARE) [15] and Semantic Subspace Projection (SSP) [28]. All of these algorithms consider the case when the images live on or close to a submanifold of the ambient space. They estimate the geometrical and discriminant properties of the submanifold from random points lying on this unknown submanifold (both labeled and unlabeled). The effectiveness of these approaches have been verified in several experiments [10, 15, 28]. However, it is not clear what is the intrinsic relation between these algorithms although they have the same manifold assumption. Moreover, the computation of these methods involves eigen-decomposition of dense matrices which is expensive in both time and memory. It is difficult to apply these approaches to very high dimensional data of large size.

In this paper, we propose a novel subspace learning framework, called *Spectral Regression* (SR), which unifies many existing manifold-based subspace learning algorithms and provides an efficient way

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.
Copyright 2007 ACM 978-1-59593-701-8/07/0009 ...$5.00.

to solve the corresponding optimization problems. This framework provides with us a nice platform to analyze the difference and relationship between various kinds of algorithms. Moreover, it can also be used to design news algorithms. Based on this framework, we develop a novel semi-supervised subspace learning algorithm, *SR*, which is shown to be able to make efficient use of both labeled and unlabeled points to discover the intrinsic discriminant structure in the data. The experimental results validate that the new method achieves a significantly higher precision for image retrieval. The specific contributions of this paper include:

- It provides a unified graph embedding analysis of three state-of-the-art semi-supervised subspace learning algorithms: LPP, ARE, and SSP (Section 2).
- It proposes a novel spectral regression approach to solve the optimization problem of the linear graph embedding, which reduces the cubic-time complexity to linear-time complexity (Section 3).
- It develops a novel semi-supervised subspace learning algorithm SR in this framework, which is shown to be able to make efficient use of both labeled and unlabeled points to discover the intrinsic discriminant structure in the data (Section 3).
- We have performed extensive experimental comparisons of the four algorithms and provided the explanation of different behaviors of these algorithms based on the SR framework (Section 5).

We summarize our findings and discuss extensions to the current work in Section 6, which concludes the paper.

2. GRAPH EMBEDDING VIEW OF SUBSPACE LEARNING

In this Section, we provide a general framework of analysis for the existing subspace learning algorithms from the graph embedding viewpoint. Particularly, the computational complexities of these algorithms can be well studied within this framework.

2.1 Graph based Subspace Learning

Given m samples $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^n$, dimensionality reduction (or, subspace learning) aims at finding $\{\mathbf{z}_i\}_{i=1}^m \subset \mathbb{R}^d, d \ll n$, where $\mathbf{z}_i$ can "represent" $\mathbf{x}_i$. In the past decades, many algorithms, either supervised or unsupervised, have been proposed to solve this problem. Despite the different motivations of these algorithms, they can be nicely interpreted in a general *graph embedding* framework.

Given a graph G with m vertices, each representing a data point, let W be a symmetric $m \times m$ matrix with W_{ij} having the weight of the edge joining vertices i and j. The G and W can be defined to characterize certain statistical or geometric properties of the data set. The purpose of graph embedding is to represent each vertex of a graph as a low dimensional vector that preserves similarities between the vertex pairs, where similarity is measured by the edge weight.

Let $\mathbf{y} = [y_1, y_2, \cdots, y_m]^T$ be the map from the graph to the real line. The optimal $\mathbf{y}$ tries to minimize

$$\sum_{i,j}(y_i - y_j)^2 W_{ij}$$

under appropriate constraint. This objective function incurs a heavy penalty if neighboring vertices i and j are mapped far apart. Therefore, minimizing it is an attempt to ensure that if vertices i and j are "close" then y_i and y_j are close as well [7]. With some simple algebraic formulations, we have

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

where $L = D - W$ is the *graph Laplacian* [6] and D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. Finally, the minimization problem reduces to find

$$\mathbf{y}^* = \arg\min_{\mathbf{y}^T D\mathbf{y}=1} \mathbf{y}^T L\mathbf{y} = \arg\min \frac{\mathbf{y}^T L\mathbf{y}}{\mathbf{y}^T D\mathbf{y}} = \arg\max \frac{\mathbf{y}^T W\mathbf{y}}{\mathbf{y}^T D\mathbf{y}}, \quad (1)$$

where the constraint $\mathbf{y}^T D\mathbf{y} = 1$ removes an arbitrary scaling factor in the embedding. Many recently proposed manifold learning algorithms, like ISOAMP [26], Laplacian Eigenmap [2], Locally Linear Embedding [20], can be interpreted in this framework with different choices of W. The two matrices W and D play the essential role in this graph embedding approach. The choices of these two graph matrices can be very flexible. In later discussion, we use GE(W, D) to denote the graph embedding with maximization problem of $\max(\mathbf{y}^T W\mathbf{y})/(\mathbf{y}^T D\mathbf{y})$.

The graph embedding approach described above only provides the mappings for the graph vertices in the training set. For some applications, a mapping for all samples, including new test samples, is required. If we choose a linear function, *i.e.*, $y_i = f(\mathbf{x}_i) = \mathbf{a}^T\mathbf{x}_i$, we have $\mathbf{y} = X^T\mathbf{a}$ where $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m] \in \mathbb{R}^{n\times m}$. Eqn. (1) can be rewritten as:

$$\mathbf{a}^* = \arg\max \frac{\mathbf{y}^T W\mathbf{y}}{\mathbf{y}^T D\mathbf{y}} = \arg\max \frac{\mathbf{a}^T XWX^T\mathbf{a}}{\mathbf{a}^T XDX^T\mathbf{a}}.$$

The optimal $\mathbf{a}$'s are the eigenvectors corresponding to the maximum eigenvalue of eigen-problem:

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

This approach is called linear extension of graph embedding. With different choices of affinity matrix W and constraint matrix D, this framework will lead to many popular linear dimensionality reduction algorithms, *e.g.*, Linear Discriminant Analysis [4] and Locality Preserving Projection [11].

In the following, we will analyze in detail the three state-of-the-art semi-supervised subspace learning algorithms in CBIR. They are incremental Locality Preserving Projection (LPP) [10], Augmented Relation Embedding (ARE) [15], and Semantic Subspace Projection (SSP) [28]. We will show that all of these three algorithms are linear extensions of graph embedding.

All the three algorithms use a k-nearest neighbors graph to model the local geometric structure of the data. Let the corresponding weight matrix be $W \in \mathbb{R}^{m\times m}$, defined by

$$W_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where $N_k(\mathbf{x}_i)$ denotes the set of k nearest neighbors of $\mathbf{x}_i$.

LPP

With the user-provided feedbacks (label information), the incremental LPP updates the k-nearest neighbors graph W as follows:

$$W_{ij}^{LPP} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same label,} \\ 0, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ have different labels,} \\ W_{ij}, & \text{otherwise.} \end{cases} \quad (3)$$

LPP then finds the optimal projection directions as:

$$\mathbf{a}^* = \arg\min \frac{\mathbf{a}^T XL^{LPP}X^T\mathbf{a}}{\mathbf{a}^T XD^{LPP}X^T\mathbf{a}}, \quad (4)$$

where D^{LPP} is a diagonal matrix whose entries are column sums (or row sums, since W^{LPP} is symmetric) of W^{LPP} and $L^{LPP} = D^{LPP} - W^{LPP}$ is the graph Laplacian. It is easy to verify that the objective function of LPP has the following equivalent variations:

$$\mathbf{a}^* = \arg\max \frac{\mathbf{a}^T X W^{LPP} X^T \mathbf{a}}{\mathbf{a}^T X D^{LPP} X^T \mathbf{a}} = \arg\max \frac{\mathbf{a}^T X W^{LPP} X^T \mathbf{a}}{\mathbf{a}^T X L^{LPP} X^T \mathbf{a}}.$$

LPP is the linear extension of graph embedding problem GE(W^{LPP}, D^{LPP}) or GE(W^{LPP}, L^{LPP}).

ARE

Different from LPP, ARE uses an additional graph[1] to encode the label information provided by user's relevance feedbacks. Let $\mathbf{F}^+$ denote the set of images in the user's feedback that are relevant to the query, and $\mathbf{F}^-$ denote the set of irrelevant images. ARE constructs the label graph as:

$$W_{ij}^{ARE} = \begin{cases} -\gamma, & \text{if } \mathbf{x}_i \in \mathbf{F}^+ \text{ and } \mathbf{x}_j \in \mathbf{F}^+, \\ 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ have different labels,} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where γ is a parameter used to take care of the possibility of unbalanced feedback. ARE then finds the optimal projection directions as:

$$\mathbf{a}^* = \arg\max \frac{\mathbf{a}^T X L^{ARE} X^T \mathbf{a}}{\mathbf{a}^T X L X^T \mathbf{a}}, \quad (6)$$

where L^{ARE} and L are the graph Laplacians of W^{ARE} and W in Eqn. (2) respectively. Clearly, ARE is the linear extension of graph embedding problem GE(L^{ARE}, L).

SSP

Similar to ARE, SSP also uses an additional graph to encode the label information:

$$W_{ij}^{SSP} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ have different labels,} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

SSP finds the optimal projection directions as:

$$\mathbf{a}^* = \arg\max \frac{\mathbf{a}^T S_{Diss} \mathbf{a}}{\mathbf{a}^T S_{GSSim} \mathbf{a}}, \quad (8)$$

where

$$\begin{aligned} S_{Diss} &= \sum_{i,j} (\mathbf{m}_i - \mathbf{m}_j)(\mathbf{m}_i - \mathbf{m}_j)^T W_{ij}^{SSP} \\ S_{GSSim} &= \sum_{i,j} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \overline{W}_{ij} \\ \overline{W} &= D^{-1} W \quad (W \text{ is defined in Eqn. (3)}) \\ \mathbf{m}_i &= \sum_j \mathbf{x}_j \overline{W}_{ij} \end{aligned}$$

Let $M = [\mathbf{m}_1, \cdots, \mathbf{m}_m]$. It is easy to check that $M = X\overline{W}^T$. Since W^{SSP} is symmetric, we have

$$\begin{aligned} S_{Diss} &= \sum_{i,j} (\mathbf{m}_i - \mathbf{m}_j)(\mathbf{m}_i - \mathbf{m}_j)^T W_{ij}^{SSP} \\ &= 2MD^{SSP}M^T - 2MW^{SSP}M^T \\ &= 2X\overline{W}^T L^{SSP} \overline{W} X^T \end{aligned}$$

$\overline{W}$ is non-symmetric. Let $\overline{D}$ and $\overline{D}'$ denote the diagonal matrices whose entries are row sums and column sums of $\overline{W}$ respectively. Define $\widetilde{W} = \overline{W} + \overline{W}^T$ which is symmetric and $\widetilde{D}$ be the diagonal matrices whose entries are row (or column) sums $\widetilde{W}$. It is easy to check that $\widetilde{D} = \overline{D} + \overline{D}'$. We have

$$\begin{aligned} S_{GSSim} &= \sum_{i,j} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \overline{W}_{ij} \\ &= X\overline{D}X^T - X\overline{W}X^T + X\overline{D}'X^T - X\overline{W}^T X^T \\ &= X(\widetilde{D} - \widetilde{W})X^T \\ &= X\widetilde{L}X^T \end{aligned}$$

where $\widetilde{L}$ is the graph Laplacian of $\widetilde{W}$.

The objective function of SSP in Eqn. (8) can be rewritten as

$$\mathbf{a}^* = \arg\max \frac{\mathbf{a}^T S_{Diss} \mathbf{a}}{\mathbf{a}^T S_{GSSim} \mathbf{a}} = \arg\max \frac{\mathbf{a}^T X\overline{W}^T L^{SSP} \overline{W} X^T \mathbf{a}}{\mathbf{a}^T X \widetilde{L} X^T \mathbf{a}}$$

It is now clear that SSP is the linear extension of graph embedding problem GE($\overline{W}^T L^{SSP} \overline{W}$, $\widetilde{L}$).

The above analysis shows that all the three subspace learning algorithms are linear extensions of the graph embedding approach

$$\arg\max \frac{\mathbf{y}^T B \mathbf{y}}{\mathbf{y}^T C \mathbf{y}} \Rightarrow \arg\max \frac{\mathbf{a}^T X B X^T \mathbf{a}}{\mathbf{a}^T X C X^T \mathbf{a}} \quad (9)$$

with different choices of affinity graph B and constraint graph C. The optimal $\mathbf{a}$'s (projection functions) are the eigenvectors corresponding to the maximum eigenvalue of eigen-problem:

$$XBX^T\mathbf{a} = \lambda XCX^T\mathbf{a}. \quad (10)$$

2.2 Computational Analysis

To get a stable solution of the eigen-problem in Eqn. (10), the matrices XCX^T is required to be non-singular [24] which is not true when the number of features is larger than the number of samples. The Singular Value Decomposition (SVD) can be used to solve this problem. Suppose $rank(X) = r$, the SVD decomposition of X is

$$X = U\Sigma V^T \quad (11)$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_r)$ and $\sigma_1 \geq \cdots \geq \sigma_r > 0$ are the singular values of X, $U \in \mathbb{R}^{n\times r}$, $V = \in \mathbb{R}^{m\times r}$ and $U^TU = V^TV = I$. Let $\widetilde{X} = U^TX = \Sigma V^T$ and $\mathbf{b} = U^T\mathbf{a}$, we have

$$\mathbf{a}^T XBX^T \mathbf{a} = \mathbf{a}^T U\Sigma V^T BV\Sigma U^T \mathbf{a} = \mathbf{b}^T \widetilde{X} B \widetilde{X}^T \mathbf{b}$$

and

$$\mathbf{a}^T XCX^T \mathbf{a} = \mathbf{a}^T U\Sigma V^T CV\Sigma U^T \mathbf{a} = \mathbf{b}^T \widetilde{X} C \widetilde{X}^T \mathbf{b}$$

Now, the objective function in (9) can be rewritten as:

$$\mathbf{b}^* = \arg\max \frac{\mathbf{b}^T \widetilde{X} B \widetilde{X}^T \mathbf{b}}{\mathbf{b}^T \widetilde{X} C \widetilde{X}^T \mathbf{b}},$$

and the optimal $\mathbf{b}$'s are the eigenvectors corresponding to the maximum eigenvalues of eigen-problem:

$$\widetilde{X} B \widetilde{X}^T \mathbf{b} = \lambda \widetilde{X} C \widetilde{X}^T \mathbf{b}. \quad (12)$$

It is clear that $\widetilde{X} C \widetilde{X}^T$ is nonsingular and the above eigen-problem can be stably solved. After we get $\mathbf{b}^*$, the $\mathbf{a}^*$ can be obtained by

$$\mathbf{a}^* = U\mathbf{b}^*. \quad (13)$$

[1] The original ARE paper [15] uses two additional graphs. These two graphs can be equivalently unified into one as shown in this paper.

The above SVD approach has been widely used in many subspace learning algorithms (*e.g.*, LDA [4] and LPP [12]) to solve the singularity problem. For clarity, we name this approach as SVD+LGE (Linear Graph Embedding). The LPP, ARE and SSP can be treated as different instances of SVD+LGE.

Now let us analyze the computational complexity of SVD+LGE. We consider the case that the number of features (n) is larger than the number of samples (m) and use the term *flam* [23], a compound operation consisting of one addition and one multiplication, to present operation counts.

All these algorithms need to construct the k-nearest neighbor graph in Eqn. (2). The cost is around $\frac{1}{2}m^2n + 2mn + m^2 \log m$ flam. $\frac{1}{2}m^2n + 2mn$ is used to calculate the pairwise distances and $m^2 \log m$ is used for m times sorting[2]. The most efficient algorithm to calculate the SVD decomposition requires $\frac{3}{2}m^2n + \frac{9}{2}m^3$ flam [24]. When $m < n$, the rank of X is usually of m. Thus, $\widetilde{X}$ is square matrix of size $m \times m$. The calculation of matrices $\widetilde{X}B\widetilde{X}^T$ and $\widetilde{X}C\widetilde{X}^T$ requires $2m^3$ flam. The eigen-problem in Eqn. (12) requires $\frac{9}{2}m^3$ flam [24]. Overall, the time complexity of these subspace learning algorithms measured by flam is

$$m^2(2n + \log m) + 11m^3,$$

which is cubic-time complexity with respect to m. For large scale high dimensional data, these algorithms are unlikely to be applied.

3. SPECTRAL REGRESSION FRAMEWORK FOR SUBSPACE LEARNING

Although those semi-supervised subspace learning algorithms are effective in relevance feedback image retrieval, the high computational cost restricts them to be applied to large scale high dimensional data sets. In this section, we describe our approach which can overcome this difficulty.

3.1 Spectral Regression

In order to solve the this eigen-problem in Eqn. (10) efficiently, we use the following theorem:

THEOREM 1. *Let* **y** *be the eigenvector of eigen-problem*

$$B\mathbf{y} = \lambda C\mathbf{y} \tag{14}$$

with eigenvalue λ. *If* $X^T\mathbf{a} = \mathbf{y}$, *then* **a** *is the eigenvector of eigen-problem in Eqn. (10) with the same eigenvalue* λ.

PROOF. We have $B\mathbf{y} = \lambda C\mathbf{y}$. At the left side of Eqn. (10), replace $X^T\mathbf{a}$ by **y**, we have

$$XBX^T\mathbf{a} = XB\mathbf{y} = X\lambda C\mathbf{y} = \lambda XC\mathbf{y} = \lambda XCX^T\mathbf{a}$$

Thus, **a** is the eigenvector of eigen-problem Eqn. (10) with the same eigenvalue λ. □

Theorem (1) shows that instead of solving the eigen-problem Eqn. (10), the linear embedding functions can be acquired through two steps:

1. Solve the eigen-problem in Eqn. (14) to get **y**.
2. Find **a** which satisfies $X^T\mathbf{a} = \mathbf{y}$. In reality, such **a** might not exist. A possible way is to find **a** which can best fit the equation in the least squares sense:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \sum_{i=1}^{m} (\mathbf{a}^T\mathbf{x}_i - y_i)^2 \tag{15}$$

where y_i is the i-th element of **y**.

The advantages of this two-step approach are as follows:

1. Both B and C are sparse matrices and the top c eigenvectors of eigen-problem in Eqn. (14) can be efficiently calculated with Lanczos algorithms [24].
2. The technique to solve the least square problem is already matured [23] and there exist many efficient iterative algorithms (*e.g.*, LSQR [18]) that can handle very large scale least square problems.

In the situation that the number of samples is smaller than the number of features, the minimization problem (15) is *ill posed*. We may have infinitely many solutions for the linear equations system $X^T\mathbf{a} = \mathbf{y}$ (the system is underdetermined). The most popular way to solve this problem is to impose a penalty on the norm of **a**:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left(\sum_{i=1}^{m} \left(\mathbf{a}^T\mathbf{x}_i - y_i\right)^2 + \alpha\|\mathbf{a}\|^2 \right) \tag{16}$$

This is called regularization and is well studied in statistics. The regularized least square is also called ridge regression [8]. The $\alpha \geq 0$ is a parameter to control the amounts of shrinkage. Now we can see the third advantage of the two-step approach:

3 Since the regression was used as a building block, the regularization techniques can be easily incorporated and produce more stable and meaningful solutions, especially when there exist a large number of features [8].

The two-step approach essentially performs regression after the spectral analysis of the graph. Therefor, we name this new approach *Spectral Regression* (SR) [3].

3.2 Theoretical Analysis

The regularized least squares problem of SR in Eqn. (16) can be rewritten in the matrix form as:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left((X^T\mathbf{a} - \mathbf{y})^T(X^T\mathbf{a} - \mathbf{y}) + \alpha\mathbf{a}^T\mathbf{a} \right). \tag{17}$$

Requiring the derivative of right side with respect to **a** vanish, we get

$$\begin{aligned} (XX^T + \alpha I)\mathbf{a} &= X\mathbf{y} \\ \Rightarrow\ \mathbf{a} &= (XX^T + \alpha I)^{-1}X\mathbf{y} \end{aligned} \tag{18}$$

When $\alpha > 0$, this regularized solution will not satisfy the linear equations system $X^T\mathbf{a} = \mathbf{y}$ and **a** will not be the eigenvector of eigen-problem in Eqn. (10). It is interesting and important to see when SR gives the exact solutions of eigen-problem (10). Specifically, we have the following theorem:

THEOREM 2. *Suppose* **y** *is the eigenvector of eigen-problem in Eqn. (14), if* **y** *is in the space spanned by row vectors of* X, *the corresponding projective function* **a** *of SR calculated in Eqn. (18) will be the eigenvector of eigen-problem in Eqn. (10) as* α *deceases to zero.*

PROOF. See Appendix A. □

When the the number of features is larger than the number of samples, the sample vectors are usually linearly independent, *i.e.*, $rank(X) = m$. In this case, we will have a stronger conclusion for SR which is shown in the following Corollary.

[2] There exist more efficient algorithms to obtain the k-nearest neighbors in stead of sorting the m numbers. We will not discuss this since it is beyond the scope of this paper.

Table 1: Computational complexity of SVD+LGE and SR (operation counts, *flam* [23])

	C_W	C_{SVD}	C_{DEigen}	C_{All}
SVD+LGE	$m^2(\frac{1}{2}n + \log m)$	$\frac{3}{2}m^2n + \frac{9}{2}m^3$	$\frac{13}{2}m^3$	$m^2(2n + \log m) + 11m^3$
		C_{SEigen}	C_{RLS}	
SR		$dp_1m(k+8)$	$2dp_2mn$	$m^2(\frac{1}{2}n + \log m) + dm(p_1 + 2p_2n)$

C_W: Complexity of the graph construction.
C_{SVD}: Complexity of SVD decomposition.
C_{DEigen}: Complexity of dense eigen-problem.
C_{SEigen}: Complexity of sparse eigen-problem.
C_{RLS}: Complexity of regularized least squares.
C_{All}: Complexity of the whole algorithm.

m: the number of data samples.
n: the number of features. We consider the case that $n > m$
k: the number of nearest neighbors.
d: the number of dimensions calculated in SR.
p_1: the number of iterations in Lanczos.
p_2: the number of iterations in LSQR.

COROLLARY 3. *If the sample vectors are linearly independent,* i.e., $rank(X) = m$, *all the projective functions in SR are the eigenvectors of eigen-problem in Eqn. (10) as* α *deceases to zero. These solutions are identical to those of SVD+LGE in Eqn. (13).*

PROOF. See Appendix B. □

3.3 Computational Complexity Analysis

Besides constructing the k-nearest neighbor graph, SR needs to solve a sparse eigen-problem in Eqn. (14) and a regularized least squares problem in Eqn. (16).

The k-nearest neighbor matrix W in Eqn. (2) is sparse and there is around k non-zero elements in each row. Both matrices B and C are developed on W and they are also sparse (with k non-zero elements in each row). The Lanczos algorithm can be used to iteratively compute the first d eigenvectors within $dp_1m(k+8)$ flam, where p_1 is the number of iterations[3] in Lanczos [24].

The regularized least squares problem in Eqn. (16) can be efficiently solved by the iterative algorithm LSQR [18]. In each iteration, LSQR needs to compute two matrix-vector products in the form of $X\mathbf{p}$ and $X^T\mathbf{q}$. The remaining work load of LSQR in each iteration is $3m+5n$ flam [17]. Thus, the time cost of LSQR in each iteration is $2mn + 3m + 5n$. If LSQR stops after p_2 iterations[4], the time cost is $p_2(2mn + 3m + 5n)$. Finally, the total time cost for d projective functions is $dp_2(2mn + 3m + 5n)$.

We summarize our complexity analysis results in Table 1 and only show the dominant part of the time cost for simplicity. It is clear to see the computational advantage of SR over traditional SVD+LGE, especially for the large scale high dimensional data (with large m and n). Please refer our technical report [3][4] for more detailed analysis.

3.4 An Algorithm Instance

SR provides an efficient framework for graph embedding problems. With the different choices of affinity graph B and constraint graph C as discussed in Section (2), SR can efficiently calculate the solutions of LPP, ARE and SSP. Moreover, the spectral regression framework provides us a powerful platform to design new algorithms. In this subsection, we describe an algorithm instance developed under this framework, which will then be tested in the later experiments. For simplicity, we will name this algorithm as SR. In the remaining part of the paper, SR will be referred to this particular algorithm if there is no specific description.

SR is a semi-supervised subspace learning algorithm. Given a labeled set $\{\mathbf{x}_i\}_{i=1}^{l}$ and an unlabeled set $\{\mathbf{x}_i\}_{i=l+1}^{m}$. These samples belong to c classes and let l_r be the number of labeled samples in the r-th class ($\sum_{r=1}^{c} l_r = l$). Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$. Without loss of generality, we assume that the first l examples are labeled and these examples are ordered according to their labels. The algorithmic procedure of SR is stated below:

1. **Construct the adjacency graph**: Construct the k-nearest neighbors graph matrix with label information W as in Eqn. (3). Calculate the graph Laplacian $L = D - W$, where D is a diagonal matrix whose (i,i)-th element equals to the sum of the i-th column (or row, since W is symmetric) of W.

2. **Construct the labeled graph**: Construct the weight matrix $W^{SR} \in \mathbb{R}^{m\times m}$ for labeled graph as

$$W_{ij}^{SR} = \begin{cases} 1/l_r, & \text{if both } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to} \\ & \text{the } r\text{-th class,} \\ 0, & \text{otherwise.} \end{cases} \qquad (19)$$

It is clear that W^{SR} has the structure as follows

$$W^{SR} = \begin{bmatrix} W_{l\times l} & 0 \\ 0 & 0 \end{bmatrix}$$

where $W_{l\times l} \in \mathbb{R}^{l\times l}$ has the following structure

$$W_{l\times l} = \begin{bmatrix} W^{(1)} & 0 & \cdots & 0 \\ 0 & W^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W^{(c)} \end{bmatrix} \qquad (20)$$

where $W^{(r)} \in \mathbb{R}^{l_r\times l_r}$ with all the elements equal to $1/l_r$ ($r = 1, \cdots, c$).

It is easy to check that W^{SR} is of rank c. Let D^{SR} be the diagonal matrix whose (i,i)-th element equals to the sum of the i-th column (or row, since W^{SR} is symmetric) of W^{SR}. The first l diagonal elements of D^{SR} are 1 and all the other elements of D^{SR} are zero.

3. **Responses generation**: Find the c eigenvectors of generalized eigen-problem with respect to non-zero eigenvalues:

$$W^{SR}\mathbf{y} = \lambda\Big(D^{SR} + L\Big)\mathbf{y}$$

Since W^{SR} is of rank c, we will have exactly c eigenvectors with respect to non-zero eigenvalues [24]. We denote them as $\mathbf{y}_1, \cdots, \mathbf{y}_c$.

4. **Regularized least squares**: Find c vectors $\mathbf{a}_1, \cdots, \mathbf{a}_c \in \mathbb{R}^n$. $\mathbf{a}_r$ ($r = 1, \cdots, c$) is the solution of regularized least square problem:

$$\mathbf{a}_r = \arg\min_{\mathbf{a}} \left(\sum_{i=1}^{m} (\mathbf{a}^T\mathbf{x}_i - y_i^r)^2 + \alpha\|\mathbf{a}\|^2 \right)$$

[3] Lanczos algorithm converges very fast, 20 iterations are usually enough to achieve a satisfactory precision [24].

[4] LSRQ converges very fast [18]. In our experiments, 30 iterations are enough.

where y_i^r is the i-th element of $\mathbf{y}_r$. Our theoretical analysis shows that the regularized least squares gives the eigenvector solution when α decreases to zero. In practical we can set $\alpha = 10^{-6}$.

5. **SR Embedding**: Let $A = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_c]$, A is a $n \times c$ transformation matrix. The samples can be embedded into c dimensional subspace by

$$\mathbf{x} \rightarrow \mathbf{z} = A^T\mathbf{x}$$

It is clear that our algorithm is a linear extension of graph embedding problem:

$$\mathbf{y}^* = \arg\max \frac{\mathbf{y}^T W^{SR} \mathbf{y}}{\mathbf{y}^T \Big(D^{SR} + L \Big) \mathbf{y}}.$$

To get a better understanding that why we choose this graph embedding, we need to examine our graph structure. Notice that the SR essentially computes the optimal projections with respect to the following objective function

$$\mathbf{a}^* = \arg\max \frac{\mathbf{a}^T X W^{SR} X^T \mathbf{a}}{\mathbf{a}^T X \Big(D^{SR} + L \Big) X^T \mathbf{a}} \qquad (21)$$

Let $X_l = [\mathbf{x}_1, \cdots, \mathbf{x}_l]$ be the labeled data matrix. Notice the special structure of W^{SR} and D^{SR}, we have

$$\begin{aligned} \mathbf{a}^* &= \arg\max \frac{\mathbf{a}^T X W^{SR} X^T \mathbf{a}}{\mathbf{a}^T X D^{SR} X^T \mathbf{a} + \mathbf{a}^T X L X^T \mathbf{a}} \\ &= \arg\max \frac{\mathbf{a}^T X_l W_{l \times l} X_l^T \mathbf{a}}{\mathbf{a}^T X_l X_l^T \mathbf{a} + \mathbf{a}^T X L X^T \mathbf{a}} \end{aligned}$$

The above objective function essentially includes two parts

$$\mathcal{O}_1 = \max \frac{\mathbf{a}^T X_l W_{l \times l} X_l^T \mathbf{a}}{\mathbf{a}^T X_l X_l^T \mathbf{a}} \quad \text{and} \quad \mathcal{O}_2 = \min \mathbf{a}^T X L X^T \mathbf{a},$$

where the first part focuses on the labeled set and the second part focuses on the whole data set.

It is easy to see that $\mathcal{O}_2$ is the objective function of Locality Preserving Projection (LPP) [11]. Minimizing $\mathcal{O}_2$ means that SR tries to preserve the local geometric structure of the whole data set. When the labeled data points X_l are centered, we have $X_l X_l^T = S_t$ and $X_l W_{l \times l} X_l^T = S_b$ [12], where S_t is the total scatter matrix and S_b is the between-class scatter matrix [4]. Since the within-class scatter matrix $S_w = S_t - S_b$, we have

$$\mathcal{O}_1 = \max \frac{\mathbf{a}^T X_l W_{l \times l} X_l^T \mathbf{a}}{\mathbf{a}^T X_l X_l^T \mathbf{a}} = \max \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_t \mathbf{a}} = \max \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}},$$

which is exactly the objective function of Linear Discriminant Analysis (LDA) [4]. Thus, maximizing $\mathcal{O}_1$ means that SR tries to calculate the projections with the best class separability on the labeled examples.

The above analysis links our approach to LDA and LPP. Specifically, SR searches for the project axes on which the data points with different labels can be best separated and meanwhile the local geometric structure on both labeled and unlabeled data is best preserved.

4. CONTENT-BASED IMAGE RETRIEVAL USING SPECTRAL REGRESSION

In this section, we describe how to apply Spectral Regression to CBIR. Particularly, we consider relevance feedback driven image retrieval.

Table 2: Image features used in the experiment

Feature Name	Dimension
Color Histogram [16]	166
Color Correlogram [13]	144
Color Moment [25]	9
Wavelet Texture [1]	18
Canny Edge [5]	72
All	409

4.1 Features for Image Retrieval

Low-level image representation is a crucial problem in CBIR. General visual features includes 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 sue of shape features for image retrieval has been limited to special applications where objects or regions are readily available. In this work, we use a 409-dimensional features as shown in Table (2) which combines color, texture and shape infomration.

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 to narrow down the gap between low level visual features and high level semantic concepts [21]. 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 and the user acts as a teacher. The typical retrieval process is outlined as follows:

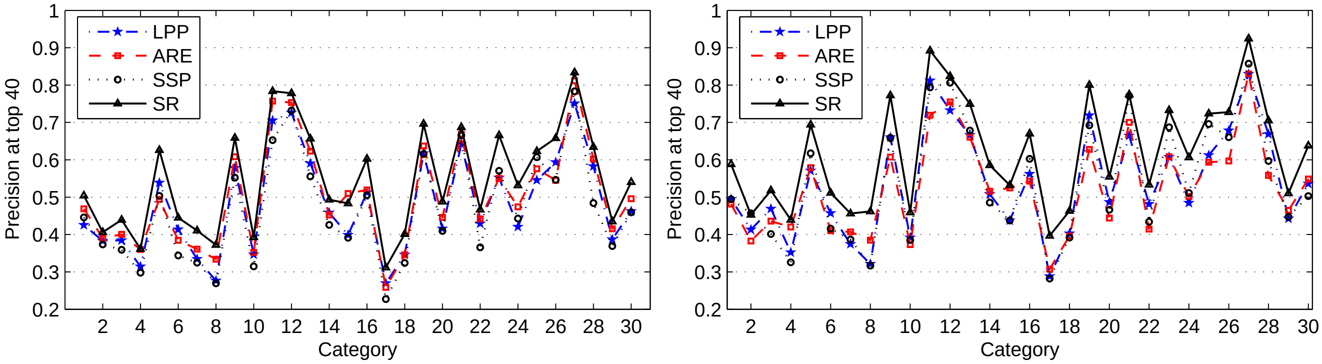
1. The user submits a query image example to the system. The system ranks the images in database according to some predefined distance metric and presents to the user the top ranked images.

2. The user provides his relevance feedbacks to the system by labeling images as "relevant" or "irrelevant".

3. The system uses the user's provided information to re-rank the images in database and returns to the user the top images. Go to step 2 until the user is satisfied.

All the four subspace learning algorithms (LPP, ARE, SSP and SR) can use the user's relevance feedbacks to update their graphs, which leads to better subspace for semantic concepts. Let $\mathbf{q}$ denote the query image and A be the transformation matrix of one subspace learning algorithm, i.e. $\mathbf{x}'_i = A^T\mathbf{x}_i$ and $\mathbf{q}' = A^T\mathbf{q}$. The distance between $\mathbf{x}'_i$ and $\mathbf{q}'$ can be computed as follows:

$$\begin{aligned} dist(\mathbf{x}'_i, \mathbf{q}') &= \sqrt{(\mathbf{x}'_i - \mathbf{q}')^T (\mathbf{x}'_i - \mathbf{q}')} \\ &= \sqrt{(\mathbf{x}_i - \mathbf{q})^T A A^T (\mathbf{x}_i - \mathbf{q})} \end{aligned}$$



(a) Feedback Iteration 1

(b) Feedback Iteration 2

Figure 1: Precision at top 40 returns of the four algorithms after the 1st feedback iteration (a) and the 2nd feedback iteration (b). Our SR algorithm is the best for almost all the categories.

For a general subspace learning algorithm, one needs to estimate the optimal dimensionality of the subspace which could be very hard in practical. Our analysis shows that there will be only c dimensions for SR subspace, where c is the number of classes. For image retrieval, $c = 2$ since there are two classes (relevant or not). Since all the other three suffer the problem of dimensionality estimation, this is one of the advantages of applying SR instead of LPP/ARE/SSP.

In many situations, the number of images in the database can be extremely large, which makes the computation of all the algorithms infeasible. In order to reduce the computational complexity, we do not take all the images in the database to construct the k nearest neighbors graphs. Instead, we only take the top 400 images at the previous retrieval iteration, plus the labeled images, to find the optimal projection.

5. EXPERIMENTS AND DISCUSSIONS

In this section, we present several experimental results and comparisons to show the effectiveness and efficiency of the proposed algorithm. All of our experiments have been performed on a P4 3.20GHz Windows XP machine with 2GB memory.

5.1 Evaluation and Implementation Settings

The COREL data set is widely used in many CBIR systems, such as [10, 15, 27, 28]. For the sake of evaluations, we also choose this data set for testing. 30 categories of color images were selected, where each consists of 100 images. Each image is represented as a 409-dimensional vector as described in Section 4.1.

To exhibit the advantages of using our approach, we need a reliable way of evaluating the retrieval performance and the comparisons with other systems. Different aspects of the experimental design are described below.

Evaluation Metrics

Due to the relatively low recall in CBIR system, we do not use the *precision-recall* curve [14]. Instead, we use *precision-scope curve* and *precision rate* as the performance evaluation metrics [15]. 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 an overall performance evaluation of the algorithms. On the other hand, the precision rate emphasizes the precision at a particular value of scope.

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 the algorithms which is also adopted in the paper [15]. More precisely, we divide the whole image database into five subsets with 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 five-fold cross validation.

Automatic Relevance Feedback Scheme

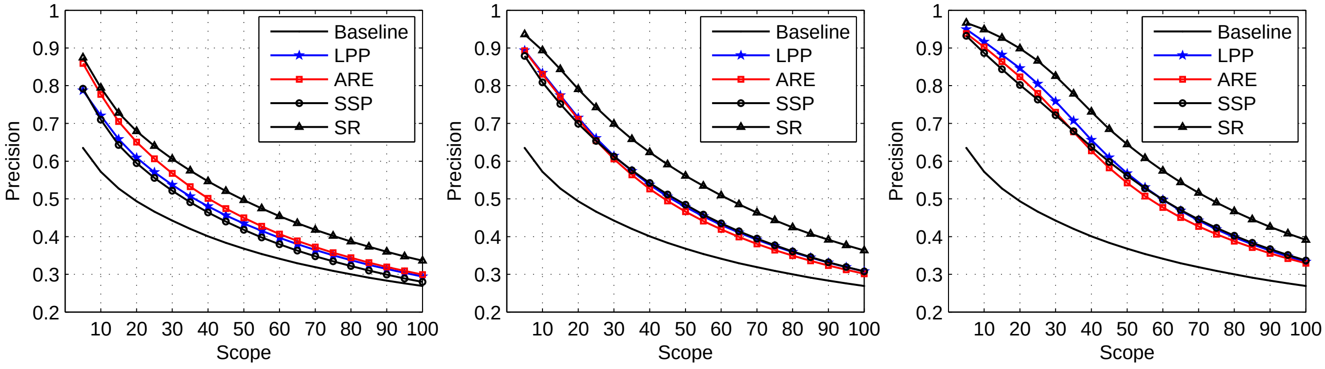
We designed an automatic feedback scheme to model the retrieval process. For each submitted query, our system retrieves and ranks the images in the database. The top 10 ranked images were selected as the feedback images, and their label information (relevant or irrelevant) is used for re-ranking. Note that, the images which have been selected at previous iterations are excluded from later selections. For each query, the automatic relevance feedback mechanism is performed for four iterations. The similar scheme was used in [10], [15], [28].

Compared Algorithms

To demonstrate the effectiveness and efficiency of our proposed image retrieval algorithm (SR), specifically the instance we described in Section 3.4, we compare it with three state-of-the-art semi-supervised subspace learning algorithms, *i.e.* incremental Locality Preserving Projection (LPP) [10], Augmented Relation Embedding (ARE) [15] and Semantic Subspace Projection (SSP) [28].

A crucial problem of LPP (or, ARE and SSP) 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. For SR, we simply use the 2-dimensional subspace. For all these algorithms, the Euclidean distances in the reduced subspace are used for ranking the images in the database. All these algorithms need to construct a k-nearest neighbors graph, we empirically set $k = 5$.

It is important to note that all the three algorithms (LPP, ARE and SSP) can be fit into the spectral regression framework to be efficiently computed. However, to show the advantages of SR, we implemented all the three algorithms in their ordinary ways (SVD+LGE approach described in Section 2).



(a) Feedback Iteration 1 (b) Feedback Iteration 2 (c) Feedback Iteration 4

Figure 2: Compare the retrieval performance of different algorithms. (a)-(c) Via illustrating with the precision-scope curves, we plot the results in the 1st, 2nd, and 4th feedback iteration, respectively. The SR algorithm performs the best on the entire scope for all the three feedback iterations.

5.2 Image Retrieval Performance

In 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 is the most important. Figure (1) shows the precision at top 40 after the first and second round of feedback for all the 30 categories. As can be seen, the retrieval performance of these algorithms varies with different categories. Our SR approach performs the best for almost all the 30 categories.

Figure 2 shows the average *precision-scope* curves of the different algorithms for the 1st, 2nd and 4th feedback iterations. The *baseline* curve describes the initial retrieval result without feedback information. Specifically, at the beginning of retrieval, the Euclidean distances in the original 409-dimensional space are used to rank the images in the database. After the user provides relevance feedbacks, the LPP, ARE, SSP, and SR algorithms are then applied to re-rank the images in the database. Our SR algorithm significantly outperforms the other three algorithms on the entire scope. The overall performances of LPP, ARE and SSP are very close to each other. ARE performs better than the other two at the first round, especially with a small scope. All these four algorithms are significantly better than the baseline, which indicates that the user provided relevance feedbacks are very helpful for improving the retrieval performance. By iteratively adding the user's feedbacks, the corresponding precisions (at top 20, top 40 and top 60) of the algorithms are respectively shown in Figure 3. As can be seen, our SR algorithm performs the best for all rounds of relevance feedback.

Table 3 gives the processing time for each query of the four algorithms. All the three algorithms LPP, ARE and SSP are computed by SVD+LGE approach as we described in Section 2. It is clear to see the SR has a significant computational advantage over the SVD+LGE approach. This results verified our theoretical analysis on computational complexity in Table 1.

Table 3: Time on processing one query for each method (s)

	t_W	t_{SVD}	t_{GEigen}	t_{All}
LPP	0.062	0.453	0.494	1.009
ARE			0.489	1.004
SSP			0.487	1.002
		t_{SEigen}	t_{RLS}	
SR		0.024	0.041	0.127

t_W: time on the graph construction.
t_{SVD}: time on SVD decomposition.
t_{GEigen}: time on generalized eigen-problem.
t_{SEigen}: time on sparse eigen-problem
t_{RLS}: time on regularized least squares

Table 4: Graph embedding for different algorithms

	numerator	denominator
LPP	W	L
ARE	L^{ARE}	L
SSP	$\overline{W}^T L^{SSP} \overline{W}$	$\widetilde{L}$
SR	W^{SR}	$D^{SR} + L$

W is defined in Eqn. (3), L is the graph Laplacian.

5.3 Model Selection on k

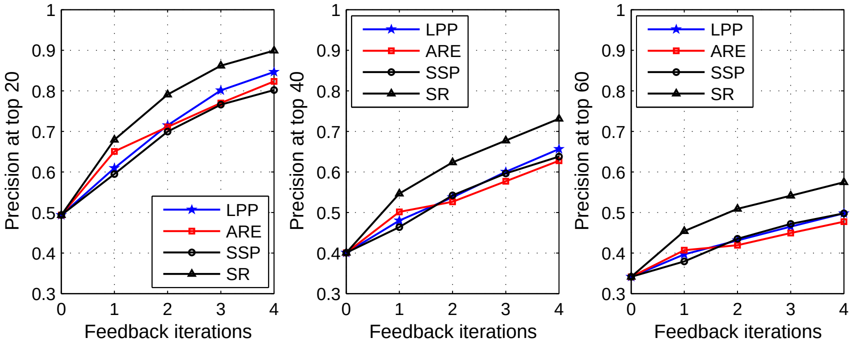
All the four algorithms are semi-supervised subspace learning algorithms. They use a k-nearest neighbor graph to model the local geometric structure of both labeled and unlabeled data. Thus, a interesting and important question could be how these algorithm sensitive to the parameter k. This is so called model selection, which is a crucial problem in most of the learning problems. In some situations, the learning performance may vary drastically with different choices of the parameters and we have to apply some model selection methods (such as Cross Validation and Bootstrapping, [8]) for estimating the generalization error. In this subsection, we evaluate the performance of the four algorithms with different values of k.

Figure (4) shows the precision at top 40 returns of the four algorithms after the first round of feedback with respect to different values of k. As can be seen, SR and ARE are more stable with different values of k. We will try to explain this result in the discussion subsection. Overall, since all the algorithms try to discover the *local* geometrical structure of the data space, it is usually set to a small number, typically less than 10.

5.4 Discussion

The spectral regression framework provides us a nice platform to analyze different algorithms. All the four algorithms we compared in the experiment are linear extensions of graph embedding. Their essential differences should lie on the different choices of graphs (affinity graph and constraint graph). For convenience, we list the different graphs used by four algorithms in Table 4.

The numerator indicates part of the objective function that the



(a) P@20

(b) P@40

(c) P@60

Figure 3: Precisions at top 10 (P@10), top 20 (P@20), and 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.

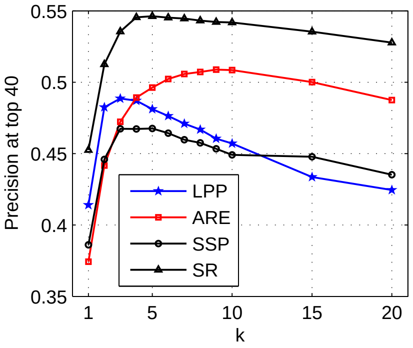


Figure 4: Model selection for LPP, ARE, SSP and SR: P@40 vs. different values of the parameter k.

algorithm tries to *maximize*, while the denominator indicates the part which will be *minimized*. It can be see that the denominators of all the four algorithms are essentially same, which is the Laplacian of k-nearest neighbor graph[5]. All these algorithms try to minimize $\mathbf{a}^T XLX^T\mathbf{a}$ which is essentially the objective function of LPP [11]. The differences between these algorithms are in the numerator part. Both L^{ARE} and W^{SR} are only dependent on the labeled data, while W and $\overline{W}^T L^{SSP}\overline{W}$ are dependent on the whole k-nearest neighbor graph. This explains why LPP and SSP are more sensitive to the parameter k as shown in Figure 4. As we analyzed in the previous section, W^{SR} is essentially the graph of between-class scatter matrix. Thus, SR can obtain the projective axes on which the data points with different labels are best separated. This is the reason why SR can achieve significant better results than other three algorithms.

6. CONCLUSION AND FUTURE WORK

This paper presents a novel subspace learning framework, called *Spectral Regression*, for relevance feedback image retrieval. This framework can interpret many stat-of-the-art graph based subspace learning algorithms, such as LPP [10], ARE [15] and SSP [28], which provides us better understanding of these algorithms. The spectral regression can naturally be used by all these algorithms for a much more efficient computation. Moreover, our framework can be used as a general platform to develop new algorithms for subspace learning. As shown in this paper, we have proposed a novel semi-supervised subspace learning algorithm called SR by designing a between-class scatter graph for labeled examples and a local neighbor graph for both labeled and unlabeled examples. This new algorithm is shown to be able to make efficient use of both labeled and unlabeled points to discover the intrinsic discriminant structure in the data. The experimental results validate that the new method achieves a significantly higher precision for image retrieval.

Several questions remain to be investigated in our future work:

1. We only discuss the linear extension approach of graph embedding in this paper. However, it is easy to extend our framework to reproducing kernel Hilbert space.

2. 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.

Acknowledgments

We would like to thank Xinjing Wang at MSRA for providing the data. The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813.

7. REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.

[2] 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.

[3] D. Cai, X. He, and J. Han. Spectral regression for dimensionality reduction. Technical report, UIUC, UIUCDCS-R-2007-2856, May 2007.

[4] D. Cai, X. He, and J. Han. SRDA: An efficient algorithm for large scale discriminant analysis. Technical report, UIUC, UIUCDCS-R-2007-2857, May 2007.

[5] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

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

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

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

[9] 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.

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

[5] With some additional label information as in Eqn. (3).

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

[12] 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.

[13] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. pages 762–768, 1997.

[14] 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.

[15] 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.

[16] C. L. Novak and S. A. Shafer. Anatomy of a color histogram. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'92)*, pages 599–605, 1992.

[17] C. C. Paige and M. A. Saunders. Algorithm 583 LSQR: Sparse linear equations and least squares problems. *ACM Transactions on Mathematical Software*, 8(2):195–209, June 1982.

[18] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.

[19] R. Penrose. A generalized inverse for matrices. In *Proceedings of the Cambridge Philosophical Society*, volume 51, pages 406–413, 1955.

[20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[21] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interative content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), 1998.

[22] A. W. 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.

[23] G. W. Stewart. *Matrix Algorithms Volume I: Basic Decompositions*. SIAM, 1998.

[24] G. W. Stewart. *Matrix Algorithms Volume II: Eigensystems*. SIAM, 2001.

[25] M. A. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[26] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[27] 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.

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

APPENDIX

A. PROOF OF THEOREM 2

PROOF. Suppose $rank(X) = r$, the SVD decomposition of X is

$$X = U\Sigma V^T \tag{22}$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_r)$, $U \in \mathbb{R}^{n\times r}$, $V \in \mathbb{R}^{m\times r}$ and we have $U^TU = V^TV = I$. The $\mathbf{y}$ is in the space spanned by row vectors of X, therefor, $\mathbf{y}$ is in the space spanned by column vectors of V. Thus, $\mathbf{y}$ can be represented as the linear combination of the column vectors of V. Moreover, the combination is unique because the column vectors of V are linear independent. Suppose the combination coefficients are $b_1, \cdots, b_r$. Let $\mathbf{b} = [b_1, \cdots, b_r]^T$, we have:

$$V\mathbf{b} = \mathbf{y} \;\Rightarrow\; V^TV\mathbf{b} = V^T\mathbf{y} \;\Rightarrow\; \mathbf{b} = V^T\mathbf{y} \;\Rightarrow VV^T\mathbf{y} = \mathbf{y} \tag{23}$$

To continue our proof, we need introduce the concept of pseudo inverse of a matrix [19], which we denote as $(\cdot)^+$. Specifically, pseudo inverse of the matrix X can be computed by the following two ways:

$$X^+ = V\Sigma^{-1}U^T$$

and

$$X^+ = \lim_{\lambda\to 0}(X^TX + \lambda I)^{-1}X^T$$

The above limit exists even if X^TX is singular and $(X^TX)^{-1}$ does not exist [19]. Thus, the regularized least squares solution in SR

$$\mathbf{a} = \left(XX^T + \alpha I\right)^{-1} X\mathbf{y} \overset{\alpha\to 0}{=} (X^T)^+\mathbf{y} = U\Sigma^{-1}V^T\bar{\mathbf{y}}$$

Combine with the equation in Eqn. (23), we have

$$X^T\mathbf{a} = V\Sigma U^T\mathbf{a} = V\Sigma U^TU\Sigma^{-1}V^T\mathbf{y} = VV^T\mathbf{y} = \mathbf{y}$$

By Theorem (1), $\mathbf{a}$ is the eigenvector of eigen-problem in Eqn. (10). □

B. PROOF OF COROLLARY 3

PROOF. The matrices B and C are of size $m \times m$ and there are m eigenvectors $\{\mathbf{y}_j\}_{j=1}^m$ of eigen-problem (14). Since $rank(X) = m$, all the m eigenvectors $\mathbf{y}_j$ are in the space spanned by row vectors of X. By Theorem (2), all m corresponding $\mathbf{a}_j$ of SR are eigenvectors of eigen-problem in Eqn. (10) as α decreases to zero. They are

$$\mathbf{a}_j^{SR} = U\Sigma^{-1}V^T\mathbf{y}_j.$$

Consider the eigen-problem in Eqn. (12), since the m eigenvectors $\mathbf{y}_j$ are also in the space spanned by row vectors of $\widetilde{X} = U^TX = \Sigma V^T$, eigenvector $\mathbf{b}_j$ will be the solution of linear equations system $\widetilde{X}^T\mathbf{b}_j = \mathbf{y}_j$. The row vectors of $\widetilde{X} = \Sigma V^T$ are linearly independent, thus $\mathbf{b}_j$ is unique and

$$\mathbf{b}_j = \Sigma^{-1}V^T\mathbf{y}_j.$$

Thus, the projective functions of SVD+LGE

$$\mathbf{a}_j^{SVD+LGE} = U\mathbf{b}_j = U\Sigma^{-1}V^T\mathbf{y}_k = \mathbf{a}_j^{SR}$$

□