

# Supervised Locality Preserving Indexing for Text Categorization

Han Liu\*

Department of Computer Science  
University of Illinois at Urbana-Champaign  
Email: hanliu@ncsa.uiuc.edu

May 8, 2005

## ABSTRACT

A major characteristic of text categorization problems is the prohibitive high dimensionality of the feature space. Most discrimination methods can not work in such a condition, *Latent Semantic Indexing* (LSI) has been adopted to solve this problem. However, LSI is not an optimal representation for text categorization task mainly because of two reasons: first, the discriminative categorical information is ignored under this completely unsupervised mode; second, LSI only concentrating on reconstruction error, the neighborhood relationship can not be preserved.

In this paper, we propose an alternative method named “*Supervised Locality Preserving Indexing* (SLPI)”, By explicitly exploiting the categorical and neighborhood information, the documents can be projected into a lower dimensional semantic space in which the documents related to the same semantics and categories are close to each other. Different from Latent Semantic Indexing , our approach tries to discover both the geometric and discriminating structures of the document space. Some theoretical analysis of this method is provided. Extensive experimental evaluations are performed on Reuters-21578 and TDT2 data sets.

**Keywords:** Text Categorization, Supervised Locality Preserving Indexing, Latent Semantic Indexing, Dimensionality Reduction, Semantics

---

\*Technical report submitted according to the regulations of the advanced course CS 598 “Advanced Topics in Information Retrieval”, taught by prof. Chengxiang zhai at the University of Illinois at Urbana-Champaign

# 1 Introduction

Text categorization concerns the automatic assignment of documents to predefined categories, which is one of the most crucial techniques to organize the documents in a supervised manner. The typical discrimination methods are directly performed in the data space. However, a statistical challenge revolves around issues of text categorization task is that the document space is always of very high dimensionality, ranging from several hundreds to thousands. Due to the consideration of the *curse of dimensionality*, it is desirable to first project the documents into a lower dimensional subspace in which the semantic structure of the document space becomes clear. In the low dimensional semantic space, the traditional discrimination methods such as Nearest Neighbors classifier, Fisher Linear Discriminant Analysis can then be applied well.

*Latent Semantic Indexing* (LSI) [5] is a widely used document indexing method which produces low dimensional representations. Latent Semantic Indexing aims to address the problems deriving from the use of *synonymous*, *near-synonymous*, and *polysemous* words as dimensions of document and query representations. LSI is optimal in the sense of preserving inner-product. In other words, the inner product of two documents in the low dimensional semantic space is close to that in the original document space. Although inner product based similarity measures have proved to be effective in information retrieval, there is no evidence that it is optimal as to discovering the discriminating structure of the document space, especially when the document space is highly non-linear.

Some recently work on spectral clustering shows its capability to handle highly non-linear data. Also, its strong connections to differential geometry make it capable of discovering the manifold structure of the document space. As a result, it has been widely used in image segmentation [13], image clustering [11], video event recognition [16], etc. However, spectral clustering can not be easily generalized to novel data points. Some recent work on out-of-sample extensions can be referred to [2].

In this paper, we propose to use a novel dimensionality reduction algorithm named *Supervised Locality Preserving Indexing* (SLPI). Different from Latent Semantic Indexing, SLPI can have more discriminating power. Thus, the documents related to the same semantics or belong to the same category are close to each other in the low dimensional representation space. Also, SLPI is derived by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the document manifold. Therefore, it can discover the non-linear manifold structure to some extent. Some theoretical justifications can be traced back to [10][9]. The original LPI is not optimal in the sense of text categorization, since it's still under an unsupervised mode. A simple modification of LPI is proposed to incorporate discriminative categorical information into the affinity matrix. In this low dimensional space, we then apply traditional discriminant methods, like

K Nearest Neighbor, Logistic Regression, and Naive Bayes to categorize the documents into semantically different classes.

The rest of this paper is organized as follows: Section 2 lists two representative work on dimensionality reduction: Latent Semantic Indexing *vs.* Fisher Linear Discriminant Analysis. Section 3 introduces our proposed algorithm. Some theoretical analysis is provided in Section 4. The experimental results are shown in Section 5. Finally, we give concluding remarks and future work in Section 6.

## 2 Related Work

Many text categorization methods have been proposed, such as K-Nearest Neighbor, naïve Bayes, Logistic Regression, or Support Vector Machine have been proposed. From different perspectives, these discriminant methods can be classified into agglomerative or divisive, hard or fuzzy, deterministic or stochastic. All these methods can be applied to the original data space or the lower dimensional representation subspace. For document space, the dimensionality is generally very high. Therefore, it is desirable to combine dimensionality reduction techniques and traditional discriminant methods. Many dimensionality reduction techniques have been proposed, the most representative two techniques are Latent Semantic Indexing and Fisher Linear Discriminant Analysis. We will introduce them here.

### 2.1 Latent Semantic Indexing

LSI [5] is one of the most popular document indexing method which produces low dimensional representations under a totally unsupervised mode. The basic idea of LSI is to extract the most representative features and at the same time the reconstruction error can be minimized. Given a set of documents  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^m$ , let  $\mathbf{a}$  be the transformation vector and  $y_i = \mathbf{a}^T \mathbf{x}_i$ . The objective function of LSI can be stated below:

$$\begin{aligned} \mathbf{a}_{opt} &= \arg \min_{\mathbf{a}} \|X - \mathbf{a}\mathbf{a}^T X\|^2 \\ &= \arg \max_{\mathbf{a}} \mathbf{a}^T X X^T \mathbf{a} \end{aligned}$$

with the constraint

$$\mathbf{a}^T \mathbf{a} = 1$$

where  $X$  is the term-document matrix,  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ . It is easy to see that the solutions are the eigenvectors of the matrix  $XX^T$ . It would be important to note that  $XX^T$  becomes the data covariance matrix if the data points have a zero mean, i.e.  $X\mathbf{e} = \mathbf{0}$  where  $\mathbf{e} = (1, \dots, 1)$ . In this case, LSI is identical to Principal Component Analysis [6]. From the above analysis, we see that LSI is able to capture the global geometrical information, however, the categorical information and the local geometrical information are ignored under this unsupervised mode.

## 2.2 Fisher Linear Discriminant Analysis

In supervised mode, if the labels are available, we can apply Fisher Linear Discriminant Analysis (FLDA) [7] to reduce the document space to a low dimensional space in which the documents of different classes are far from each other and at the same time the documents of a same class are close to each other. FLDA is optimal in the sense of utilizing categorical information. Suppose the data points belong to  $k$  classes. FLDA can be obtained by solving the following maximization problem:

$$\mathbf{a}_{opt} = \arg \max_{\mathbf{a}} \frac{|\mathbf{a}^T S_b \mathbf{a}|}{|\mathbf{a}^T S_w \mathbf{a}|}$$

$$S_b = \sum_{i=1}^k n_i (\mathbf{m}^i - \mathbf{m}) (\mathbf{m}^i - \mathbf{m})^T$$

$$S_w = \sum_{i=1}^k \left( \sum_{j=1}^{n_i} (\mathbf{x}_j^i - \mathbf{m}^i) (\mathbf{x}_j^i - \mathbf{m}^i)^T \right)$$

which leads to the following generalized maximum eigenvalue problem:

$$S_b \mathbf{a} = \lambda S_w \mathbf{a} \quad (1)$$

where  $\mathbf{m}$  is the total sample mean vector,  $n_i$  is the number of samples in the  $i^{th}$  class,  $\mathbf{m}^i$  is the average vector of the  $i^{th}$  class, and  $\mathbf{x}_j^i$  is the  $j^{th}$  sample in the  $i^{th}$  class. We call  $S_w$  the *within-class scatter matrix* and  $S_b$  the *between-class scatter matrix*. It's easy to see that for  $k$ -class classification, if FLDA is applied for dimensionality reduction, the remained dimensionality can be at most  $k - 1$ . Assume we have more than 3,000 samples for binary classification, by FLDA, only 1 dimension could be kept, two much information is lost! Since only the discriminative label information is kept, while the global (local) geometric structure information is lost, this totally supervised dimensionality reduction method is on the contrary extreme of LSI.

## 2.3 Locality Preserving Indexing

Recently, a new document indexing method called *Locality Preserving Indexing* (LPI) [9] was proposed. Different from LSI which aims to discover the global Euclidean structure, LPI aims to discover the local geometrical structure. Given an affinity matrix  $S$ , LPI can be obtained by solving the following minimization problem:

$$\begin{aligned} \mathbf{a}_{opt} &= \arg \min_{\mathbf{a}} \sum_{i=1}^m (\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_i)^2 S_{ij} \\ &= \arg \min_{\mathbf{a}} \mathbf{a}^T X L X^T \mathbf{a} \end{aligned}$$

with the constraint

$$\mathbf{a}^T XDX^T \mathbf{a} = 1$$

where  $L = D - S$  is the *graph Laplacian* [3] and  $D_{ii} = \sum_j S_{ij}$ .  $D_{ii}$  measures the local density around  $\mathbf{x}_i$ . The basis functions of LPI are the eigenvectors of the matrix  $(XDX^T)^{-1}XDX^T$  associated with the smallest eigenvalues. Even though LPI is also an unsupervised method, a simple theoretical analysis can show that LPI can have more discriminating power than LSI, it has demonstrated the power in the area of image processing [9]. As we will show later in this paper, it's not difficult to incorporate label information into the construction process of the affinity matrix and extended it to be a supervised version. To design a general and flexible framework for dimensionality reduction-discriminant analysis is the motivation of this work.

From the above discussions, we see that FLDA and LSI are on the two extremes of the same spectrum. Being a supervised method, FLDA only captures the categorical information, while the geometrical information of the sample in the document space is totally ignored; Being an unsupervised approach, LSI only captures the global geometric information, while the discriminative categorical information is lost; What's more, both these two approaches do not consider any local neighbor relationships of the samples in the document space. Compared with these two extremes, LPI could capture sample local geometric structure and has more discriminative power than LSI. In this paper, with LPI as a basic component, we further generalize the construction process of the affinity matrix, leading to a unified framework, which could consider the chlorotical labels and geometric structure information simultaneously.

### 3 Supervised Locality Preserving Indexing

In this section, we describe the general framework of text categorization based on *Supervised Locality Preserving Indexing* (SLPI) which can be thought of as a combination of both FLDA and LSI. We begin with the motivations of our work.

#### 3.1 Motivations

In this section, we will provide some motivations about the reasoning of SLPI follow by traditional discriminant methods like K Nearest Neighbor.

Generally, the document space is of high dimensionality, typically ranging from several thousands to tens of thousands. Learning in such a high dimensional space is extremely difficult due to the *curse of dimensionality*. Thus, document categorization necessitates some form of dimensionality reduction. When using K Nearest Neighbor as a classifier on the dimension reduced data, one of the basic assumptions behind text categorization is that, if two data points are close to each in the high dimensional space, they tend to

be close to each other in the dimension reduced space. Therefore, the optimal document indexing method should be able to discover the local geometrical structure of the document space. To this end, the LPI algorithm is of particular interest.

Another consideration is the discriminating power. One can expect that the documents should be projected into the subspace in which the documents with different semantics can be well separated while the documents with common semantics can be clustered. As we will discuss in the next section, LPI can be easily modified to incorporate the categorical label information. Thus to gain the discriminating power approximate to FLDA. There are some other linear subspace learning algorithms such as informed projection [4], Linear Dependent Dimensionality Reduction [14], etc. However, none of them have shown to have discriminating power.

Finally, it would be interesting to note that LPI is fundamentally based on manifold theory [9][10]. It tries to find a linear approximation to the eigenfunctions of the Laplace Beltrami operator on the compact Riemannian manifold, see [10] for details. Therefore, LPI is capable of discovering the nonlinear structure of the document space to some extent.

### 3.2 Methodology

Given a set of documents  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m$ . Suppose  $\mathbf{x}_i$  has been normalized to 1, thus the dot product of two document vectors is exactly the cosine similarity of the two documents. The proposed SLPI algorithm is performed as follows:

**Constructing the adjacency graphs:** Let  $G_{unsupervised}$  and  $G_{supervised}$  denote graphs with  $n$  nodes. They are purposed to capture local geometric structure and the categorical information respectively. For  $G_{unsupervised}$ , The  $i$ -th node corresponds to the document  $\mathbf{x}_i$ . For  $G_{unsupervised}$ , we put an edge between nodes  $i$  and  $j$  if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are “close”, i.e.  $\mathbf{x}_i$  is among  $p$  nearest neighbors of  $\mathbf{x}_j$  or  $\mathbf{x}_j$  is among  $p$  nearest neighbors of  $\mathbf{x}_i$ ; while for  $G_{supervised}$ , two nodes have an edge only if they belong to the same category.

**Constructing the affinity matrix:** If node  $i$  and  $j$  are connected in  $G_{unsupervised}$ , put

$$S_{ij}^1 = \mathbf{x}_i^T \mathbf{x}_j$$

Otherwise, put  $S_{ij}^1 = 0$ . The weight matrix  $S^1$  of graph  $G_{unsupervised}$  models the local structure of the document space. We define  $D$  as a diagonal matrix whose entries are column (or row, since  $S^1$  is symmetric) sums of  $S^1$ ,  $D_{ii} = \sum_j S_{ji}^1$ . We also define  $L = D - S^1$ , which is called the Laplacian matrix in spectral graph theory [3]. If node  $i$  and  $j$  are connected in  $G_{supervised}$ , put  $S_{ij}^2 = \frac{1}{n_l}$ , Otherwise, put  $S_{ij}^2 = 0$ .  $n_l$  is the number of documents belong to the  $l$ th category.  $S^2$  is mainly used to record the label information.

The final affinity matrix  $S$  defined as

$$S = \pi \cdot S^1 + (1 - \pi) \cdot S^2$$

where the  $\pi$  is a tuning parameter, its assignment should depend on the properties of dataset. Tuning of  $\pi$  could be conducted by the standard procedures, *ie.* Cross-Validation. Also, it could be assigned *a priori* based on the prior knowledge.

**Data Preprocessing and SVD Projection:** We remove the weighted mean of  $\mathbf{x}$  from each  $\mathbf{x}$

$$\hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}, \quad \bar{\mathbf{x}} = \frac{1}{(\sum_i D_{ii})} \left( \sum_i \mathbf{x}_i D_{ii} \right)$$

and projected the document vector into the SVD subspace by throwing away those *zero* singular value.

$$\hat{X} = U \Sigma V^T$$

where  $\hat{X} = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n]$ . We denote the transformation matrix of SVD by  $W_{SVD}$ ,  $W_{SVD} = U$ . After SVD projection, the document vector  $\hat{\mathbf{x}}$  becomes  $\tilde{\mathbf{x}}$ :

$$\tilde{\mathbf{x}} = W_{SVD}^T \hat{\mathbf{x}}$$

After this step, the term-document matrix  $X$  becomes  $\tilde{X} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n]$ .

**SLPI Projection:** Compute the eigenvectors and eigenvalues for the generalized eigenproblem:

$$\tilde{X} L \tilde{X}^T \mathbf{a} = \lambda \tilde{X} D \tilde{X}^T \mathbf{a} \quad (2)$$

Let  $W_{SLPI} = [\mathbf{a}_1, \dots, \mathbf{a}_k]$  be the solutions of equation (2), ordered according to their eigenvalues,  $\lambda_1 < \lambda_2 < \dots < \lambda_k$ . Thus, the embedding is as follows:

$$\mathbf{x} \rightarrow \mathbf{y} = W^T \hat{\mathbf{x}}$$

$$W = W_{SVD} W_{SLPI} \quad \text{and} \quad \hat{\mathbf{x}} = \mathbf{x} - \frac{1}{(\sum_i D_{ii})} \left( \sum_i \mathbf{x}_i D_{ii} \right)$$

where  $\mathbf{y}$  is a  $k$ -dimensional representation of the document  $\mathbf{x}$ .  $W$  is the transformation matrix.

**Discriminant Analysis in the Lower Dimensional Semantic Space:** Now we get lower dimensional representations of the original documents. In the reduced semantic space, those documents belonging to the same category or have similar semantic structure are close to one another. The traditional discrimination methods (we choose K Nearest Neighbor, Naive Bayes, and Logistic Regression in this paper) can be applied in the reduced semantic space.

## 4 Theoretical Analysis

In this section we give the theoretical analysis of our proposed approach. First, we will discuss the relationship between SLPI, LSI and FLDA. We will show that under specific settings of the parameters and the adjacency graphs, LSI and FLDA could also be viewed as a specific example of SLPI. Then, we talk about the issues of different possibilities to construct the affinity matrix.

### 4.1 Relationship between SLPI and LSI

If we set  $\pi = 1$ , SLPI is essentially obtained from a graph model  $G_{unsupervised}$  and the affinity matrix  $S = S^1$ . The graph structure represents the geometrical structure of the document space. In our algorithm, a nearest neighbor graph is constructed to discover the *local* manifold structure. Intuitively, SLPI with a complete graph should discover the *global* structure. In this subsection, we present a theoretical analysis on the relationship between SLPI and LSI under this setting. Specifically, we show that LSI can be viewed as a specific example of SLPI with a complete graph.

LSI is fundamentally based on SVD (Singular Value Decomposition). For a rank  $r$  term-document matrix  $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ , LSI decompose the  $X$  using SVD as follow:

$$X = U\Sigma V^T$$

Where  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$  and  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$  are the singular values of  $X$ ,  $U = [\mathbf{u}_1, \dots, \mathbf{u}_r]$  and  $\mathbf{u}_i$  is called left singular vectors,  $V = [\mathbf{v}_1, \dots, \mathbf{v}_r]$  and  $\mathbf{v}_i$  is called right singular vectors. LSI use the first  $k$  vectors in  $U$  as the transformation matrix to embed the original document into a  $k$  dimensional subspace. It can be easily checked that the column vectors of  $U$  are the eigenvectors of  $XX^T$ . Thus, LSI tries to solve the maximum eigenvalue problem:

$$XX^T \mathbf{a} = \lambda \mathbf{a}$$

In SLPI with the tuning parameter  $\pi = 1$ , recall that the weight on an edge linking  $\mathbf{x}_i$  and  $\mathbf{x}_j$  is set to their inner product  $\mathbf{x}_i^T \mathbf{x}_j$ . Thus, the affinity matrix  $S$  of the complete graph can be written as  $X^T X$ . Since we first apply SVD to remove the components corresponding to the zero singular value, the matrix  $XX^T$  is of full rank. The generalized minimum eigenvalue problem of SLPI can be written as follows:

$$\begin{aligned} & XLX^T \mathbf{a} = \lambda XDX^T \mathbf{a} \\ \Rightarrow & X(D - W)X^T \mathbf{a} = \lambda XDX^T \mathbf{a} \\ \Rightarrow & XWX^T \mathbf{a} = (1 - \lambda)XDX^T \mathbf{a} \\ \Rightarrow & XX^T XX^T \mathbf{a} = (1 - \lambda)XDX^T \mathbf{a} \end{aligned} \tag{3}$$

Since the diagonal matrix  $D$  is close to the identity matrix,  $XDXT \approx XX^T$ . The *minimum* eigenvalues of equation (3) correspond to the *maximum* eigenvalues of the following equation:

$$XX^TXX^T\mathbf{a} = \lambda XX^T\mathbf{a}$$

Since  $XX^T$  is of full rank, we get:

$$XX^T\mathbf{a} = \lambda\mathbf{a}$$

which is just the eigenvalue problem of LSI. The above analysis shows that SLPI with a complete graph is actually similar to LSI. Both of them discover the global structure. The only difference is that there is a diagonal matrix  $D$  in SLPI which reflects the importance of the different document vectors (In fact, if we do not center the data, SLPI is exactly reduced to LSI under this setting). In LSI, every document vector is treated equally important. In SLPI, the weight of document  $\mathbf{x}_i$  is  $D_{ii}$ . We define  $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$  as the center vector of these document vectors. In complete graph situation, we have

$$\begin{aligned} D_{ii} &= \sum_{j=1}^n S_{ij} = \sum_{j=1}^n (X^T X)_{ij} = \sum_{j=1}^n \mathbf{x}_i^T \mathbf{x}_j \\ &= \mathbf{x}_i^T \sum_{j=1}^n \mathbf{x}_j = n\mathbf{x}_i^T \bar{\mathbf{x}} = \delta \mathbf{x}_i^T \frac{\bar{\mathbf{x}}}{\|\bar{\mathbf{x}}\|} \end{aligned}$$

where  $\delta = n\|\bar{\mathbf{x}}\|$  is a constant. Note that all the  $\mathbf{x}_i$  are normalized to 1, thus they are distributed on a unit hypersphere.  $\bar{\mathbf{x}}/\|\bar{\mathbf{x}}\|$  is also on this unit hypersphere. Thus,  $D_{ii}$  evaluates the cosine of the angle between vector  $\mathbf{x}_i$  and  $\bar{\mathbf{x}}$ . In other words, the  $D_{ii}$  evaluates the cosine similarity between document  $\mathbf{x}_i$  and the center. The closer to the center the document is, the larger weight it has. Some previous researches [15] show that such  $D$  will improve the result and our experiments will also show this.

## 4.2 Relationship between SLPI and FLDA

In SLPI with the tuning parameter  $\pi = 0$ , we have the affinity matrix  $S = S^2$ , and the corresponding weights:

$$W_{ij} = \begin{cases} \frac{1}{n_l}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } l^{\text{th}} \text{ class;} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

and

$$D_{ij} = \begin{cases} \sum_j W_{ij}, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

It is easy to check that the row sum of  $W$  is 1, therefore the diagonal matrix  $D$  is exactly the identity matrix  $I$ .

$$L = D - W = I - W$$

With some algebraic steps [9], we can show that: if the sample mean is zero, the eigenproblem of equation (1) is equivalent to

$$XLX^T \mathbf{a} = \lambda XX^T \mathbf{a} \quad (5)$$

This analysis shows that if the affinity matrix  $S$  in SLPI is defined as the  $W$  in Equation (4), the result of SLPI will be identical to the FLDA.

### 4.3 The Construction of Affinity Matrix $S$

The previous two sections show that LSI can FLDA could both be viewed as a special case of SLPI under specific settings. The key difference between them is the construction of affinity matrix (the weighted matrix of graph). The LSI tries to discover the global structure (with the complete weighted graph). The FLDA is performed in supervised mode, thus the graph can be constructed to reflect the label information. From a general perspective, the SLPI can be performed in either supervised, unsupervised or semi-supervised manner, depending on what kind of information we want to capture.

The construction of the affinity matrix is also very important in spectral clustering [13][12] and spectral embedding [1]. It includes two steps: constructing the graph and setting the weight. In our algorithm, we construct a  $p$ -nearest neighbor graph and choose the dot product (cosine similarity) as the weight for  $S^1$  and a simple category dependent approach for  $S^2$ . There are also some other choices for  $S^1$  as discussed below. We put an edge between nodes  $i$  and  $j$  if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are “close”. There are two variations:

**$p$ -nearest neighbors:** Nodes  $i$  and  $j$  are connected by an edge if  $\mathbf{x}_i$  is among  $p$  nearest neighbors of  $\mathbf{x}_j$  or  $\mathbf{x}_j$  is among  $p$  nearest neighbors of  $\mathbf{x}_i$ .

Advantages: simpler to choose, tends to lead to connected graphs.

Disadvantages: less geometrically intuitive.

**$\epsilon$  neighbors:** Nodes  $i$  and  $j$  are connected by an edge if  $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$ .

Advantages: geometrically motivated, the relationship is naturally symmetric.

Disadvantages: often leads to graphs with several connected components, difficult to choose  $\epsilon$ .

In the step of setting the weight, there are several choices for both the  $S^1$  and  $S^2$  (For simplicity, we just write  $S$  to represent them):

0-1 weighting:  $S_{ij} = 1$  if and only if node  $i$  and  $j$  are connected by an edge. This is the simplest weighting method and very easy to compute.

**Gaussian kernel weighting:** If node  $i$  and  $j$  are connected, put

$$S_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$$

The gaussian kernel weighting is also called heat kernel weighting. It has intrinsic connection to the Laplace Beltrami operator on differentiable functions on a manifold [1].

**Polynomial kernel weighting:** If node  $i$  and  $j$  are connected, put

$$S_{ij} = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$$

The parameter  $d$  in the equation indicate the degree of the polynomial kernel. Order  $d$  polynomial kernel can discover non-linear structure with polynomial basis functions of order  $d$ .

**Dot product weighting:** If node  $i$  and  $j$  are connected, put

$$S_{ij} = \mathbf{x}_i^T \mathbf{x}_j$$

Note that if  $\mathbf{x}$  is normalized to 1, the dot product of two vector equivalent the cosine similarity of the two vector. The dot product weighting can discover the linear Euclidean structure of document space.

## 5 Experiments and Study Design

In this section, several experiments were performed to show the effectiveness of our method. Two standard document collections were used in our experiments, i.e Reuters-21578 and TDT2. We compared the performance of Supervised Locality Preserving Indexing and Latent Semantic Indexing for different discrimination methods.

### 5.1 Data Corpora

Table 1: 30 semantic categories from Reuters-21578 used in our experiments

category	num of doc	category	num of doc	category	num of doc
earn	3713	money-supply	87	iron-steel	37
acq	2055	gnp	63	ipi	36
crude	321	cpi	60	nat-gas	33
trade	298	cocoa	53	veg-oil	30
money-fx	245	alum	45	tin	27
interest	197	grain	45	cotton	24
ship	142	copper	44	bop	23
sugar	114	jobs	42	wpi	20
coffee	110	reserves	38	pet-chem	19
gold	90	rubber	38	livestock	18

Reuters-21578 corpus<sup>1</sup> contains 21578 documents in 135 categories. In our experiments, we discarded those documents with multiple category labels, and selected the largest 30 categories. It left us with 8067 documents in 30 categories as described in table 1.

The TDT2 corpus<sup>2</sup> consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of 11201 on-topic documents which are classified into 96 semantic categories. In this dataset, we also removed those documents appearing in two or more categories and use the largest 30 categories thus leaving us with 9394 documents in 30 categories as described in table 2. Each document is represented as a term-frequency vector. We simply removed the stop word and no further preprocessing was done.

Table 2: 30 semantic categories from TDT2 used in our experiments

category	num of doc	category	num of doc	category	num of doc
20001	1844	20048	160	20096	76
20015	1828	20033	145	20021	74
20002	1222	20039	141	20026	72
20013	811	20086	140	20008	71
20070	441	20032	131	20056	66
20044	407	20047	123	20037	65
20076	272	20019	123	20065	63
20071	238	20077	120	20005	58
20012	226	20018	104	20074	56
20023	167	20087	98	20009	52

## 5.2 Evaluation Measures

Based on different underlying principles and assumptions, some classifiers can be used for multi-class classifications directly, while some others do not. To ease the convenience of comparison, we assume all the discrimination methods as binary classifiers. The category assignments of a binary classifier can be evaluated using a two-way contingency table for each category, which has four cells, where cell  $a$  counts the documents correctly assigned to this category; cell  $b$  counts the documents incorrectly assigned to this category; cell  $c$  counts the documents incorrectly rejected from this category; cell  $d$  counts the documents correctly rejected from this category. The conventional performance measures evaluations are defined from this contingency table. In this paper, we are mainly interested in the

<sup>1</sup>Reuters-21578 corpus at <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

<sup>2</sup>Nist Topic Detection and Tracking corpus at <http://www.nist.gov/speech/tests/tdt/tdt98/index.html>

recall ( $r$ ), precision ( $p$ ), accuracy ( $Acc$ ) and error ( $Err$ ).

$$\begin{cases} r = \frac{a}{a+c} & \text{if } a+c > 0, \text{ otherwise undefined;} \\ p = \frac{a}{a+b} & \text{if } a+b > 0, \text{ otherwise undefined;} \\ Acc = \frac{a+d}{n} & \text{where } n = a+b+c+d > 0; \\ Err = \frac{b+c}{n} & \text{where } n = a+b+c+d > 0; \end{cases} \quad (6)$$

For evaluation performance average across categories, both *micro-averaging* and *macro-averaging*. Macro-averaged performance scores are computed by first computing the scores for the per-category contingency tables and then averaging these scores to compute the global measures. Micro-averaged performance scores are computed by first creating a global contingency table whose cell values are the sums of the corresponding cells in the per-category contingency tables, and then use this global contingency table to compute the scores. There is an important difference between these two scores. Macro-average performance scores gives equal weight to every category, and therefore is considered a per-document average. Likewise, micro-average performance scores give equal weight to every document, and is therefore a per-document average.

Since some of the performance measures may be misleading if we treat them separately. Such as, a trivial algorithms may get a perfect recall of 100% with a very low precision. Usually, a discriminant method exhibits a trade-off between recall and precision when the intrinsic threshold are changing from one extreme to another. To compensate this, the  $F_1$  measure is also used for evaluation, the definition is

$$F_1 = \frac{2pr}{p+r}$$

It also has both the micro- and macro- versions.

### 5.3 Discrimination Methods and Study Design

We use *K Nearest Neighbor* (KNN), Naive Bayes Classifier (NB), and Logistic Regression (LR) as the discriminant methods to evaluate performance of SLPI. Each of the methods can be used to assign objects to one of several classes, though we use them in a strictly binary mode, to assign document to one of two categories, which we refer to simply as Class 1 and Class 2, respectively. For each category, In our application, Class 1 is the set of documents that belong to given category, and Class 2 is the set of documents that do not belong to this category. The rest of this section outlines all the methods used in our study.

Although the methods differ greatly, they all produce classifiers that are discriminative. That is, given an object, the classifier produces a score or likelihood,  $p_1$ , that the object

belongs to Class 1, as well as a score,  $p_2$ , that the object belongs to Class 2. A natural decision rule is to assign an object to Class 1 if  $p_1 > p_2$ , and to Class 2 otherwise. More generally, the error rates can be adjusted by introducing a decision threshold,  $t$ , so that an object is assigned to Class 1 if and only if  $p_1/p_2 > t$ . Of course, this decision rule is equivalent to  $f(p_1/p_2) > t$ , where  $f$  is any monotonically increasing function. Often,  $f$  is the logarithmic function, in which case the rule becomes  $l_1 - l_2 > t$ , where  $l_k = \log p_k$ .

**Naive Bayes:** At an abstract level, most of the discrimination methods considered use the same method to classify an object,  $\mathbf{x}$ . First, Bayes Rule is used to compute  $p_k(\mathbf{x})$ , the posterior probability that  $\mathbf{x}$  belongs to class  $k$ :

$$p_k(\mathbf{x}) = \frac{\pi_k f_k(\mathbf{x})}{\sum_j \pi_j f_j(\mathbf{x})}$$

Here,  $\pi_k$  is the prior probability that  $\mathbf{x}$  belongs to class  $k$ , and  $f_k(\mathbf{x})$  is the conditional probability of  $\mathbf{x}$  assuming class  $k$ . Typically,  $\pi_k$  is estimated as the proportion of data points in class  $k$ . Once the posterior probabilities are estimated, they are used in a likelihood ratio test to assign  $\mathbf{x}$  to a class. In our application, for which  $K = 2$ ,  $\mathbf{x}$  is assigned to Class 1 if and only if  $p_1(\mathbf{x})/p_2(\mathbf{x}) > t$ , where  $t$  is a decision threshold.

The various classification methods differ primarily in their assumptions about the prior probabilities,  $f_k(\mathbf{x})$ . In the Naive Bayes classifier, it is assumed (simplistically) that the features of the vector  $\mathbf{x}$  can be treated as independent random variables, so that  $f_k$  can be factored into a product of distributions, one for each component of  $\mathbf{x}$ . Thus, if  $\mathbf{x} = (x_1, \dots, x_n)$ , then

$$f_k(\mathbf{x}) = \prod_i f_{ki}(x_i)$$

where  $f_{ki}(x_i)$  is the (marginal) probability of  $x_i$  for class  $k$ . Despite this simplistic assumption, Naive Bayes classifiers often perform surprisingly well [8].

Originally, univariate Gaussians were used to estimate  $f_{ki}$ , though kernel density estimates are now common [8]. In our application, multinomial distribution is used for likelihood, while Dirichlet distribution is adopted for the prior.

**Logistic Regression:** Logistic Regression can be viewed as a generalization of Linear Discriminant Analysis (LDA). Like LDA, Logistic Regression provides a linear decision boundary between classes. The main difference is that Logistic Regression is in a sense more direct. Instead of first fitting multivariate Gaussians to each class (which requires estimating  $O(n^2)$  parameters), Logistic Regression fits a linear decision directly to the data (which requires estimating only  $n$  parameters). More specifically, it assumes that for each feature vector  $\mathbf{x}$ , the log likelihood ratio is given by the equation  $\log[p_1(\mathbf{x})/p_2(\mathbf{x})] = \mathbf{x} \bullet \mathbf{w} + b$ , for some vector  $\mathbf{w}$  and some constant  $b$ . This equa-

tion is known to hold for a wide range of class density distributions. For instance, multivariate Gaussian distributions with equal covariance matrices. It is also known to hold for gamma distributions, exponential distributions, binomial distributions, Poisson distributions, and more generally, for any member of the general class of distributions known as the exponential family. In this sense, Logistic Regression is more general than LDA, from a statistical perspective, Logistic Regression could be treated as a semiparametric method.

It also requires estimating fewer parameters than LDA, as noted above. However, finding maximum likelihood estimates for the parameters of Logistic regression is more complex, since there are no closed-form formulas for them. Instead a set of nonlinear equations must be solved, using optimization techniques such as the Newton-Raphson algorithm [8].

**Nearest Neighbor Classifiers:** K Nearest Neighbors (KNN) is different from the other methods considered in this paper in that it is *non-parametric*, so there are no parameters to estimate. Instead, the training data itself is used as a sample estimate of the underlying distributions of the two classes. The classification method is simple. Given an object,  $\mathbf{x}$ , to be classified, find the  $K$  objects (or neighbors) in the training data that are closest to it. If most of these neighbors belong to Class 1, then  $\mathbf{x}$  is assigned to Class 1; otherwise, it is assigned to Class 2. When  $K = 1$ , this method is called “Nearest Neighbor,” or 1NN. In this case,  $\mathbf{x}$  is assigned to the same class as the data point that is closest to it. One can view KNN as a voting system, in which some neighbors vote for Class 1 and others vote for Class 2. In its simplest incarnation, a simple majority vote is used, but this is not necessary. For example, one might require a super-majority vote in which  $\mathbf{x}$  is assigned to Class 1 if and only if  $2/3$  of its neighbors are in Class 1. More generally,  $\mathbf{x}$  can be assigned to Class 1 if  $K_1/K_2 > t$ , for some  $t$ . Here,  $K_1$  is the number of K-nearest neighbors in Class 1, and  $K_2$  is the number in Class 2 (so  $K_1 + K_2 = K$ ). Note that this is a discriminative classifier, where  $K_1$  is interpreted as the likelihood that  $\mathbf{x}$  is in Class 1,  $K_2$  as the likelihood it is in Class 2, and  $t$  is the decision threshold.

In using KNN, one must choose a value for  $K$ . If there is no reason to prefer any particular value *a priori*, then one can train classifiers using several values of  $K$ , and choose the value that minimizes the testing error. That is the approach taken in this paper. One must also choose a measure of “distance” (or similarity) between two objects. If the objects are vectors, then several choices naturally suggest themselves, including Euclidean distance, correlation coefficient, and the angle between the vectors. cosine similarity are used in our case.

For these three discrimination methods, they can be divided into three types: parametric, semiparametric and nonparametric. For our study, we test the generalization performance of these methods across different dimensionality reduction approaches. Since

SLPI is a very flexible generic framework, we have very large degree of freedom to control the model setting. According to the previous theoretical analysis, we know that LSI can FLDA could both be viewed as a special case of SLPI under specific parameter settings, it guarantees that there must be a nonempty set of parameter settings, which makes the performance of SLPI no worse than both LSI and FLDA. Besides these quantitative measurements, we are also very interested in analyzing the qualitative aspects, *ie.* embedding space, optimal dimensionality, and model sensitivity.

## 5.4 SLPI Embedding vs. LSI Embedding

Our theoretical analysis shows that SLPI is able to map the documents belong to the same category and have close relationship in the document space as close to each other as possible. This motivates us to perform text categorization in the SLPI subspace rather than directly in the original space. In this subsection, we first present some embedding results by using SLPI and LSI.

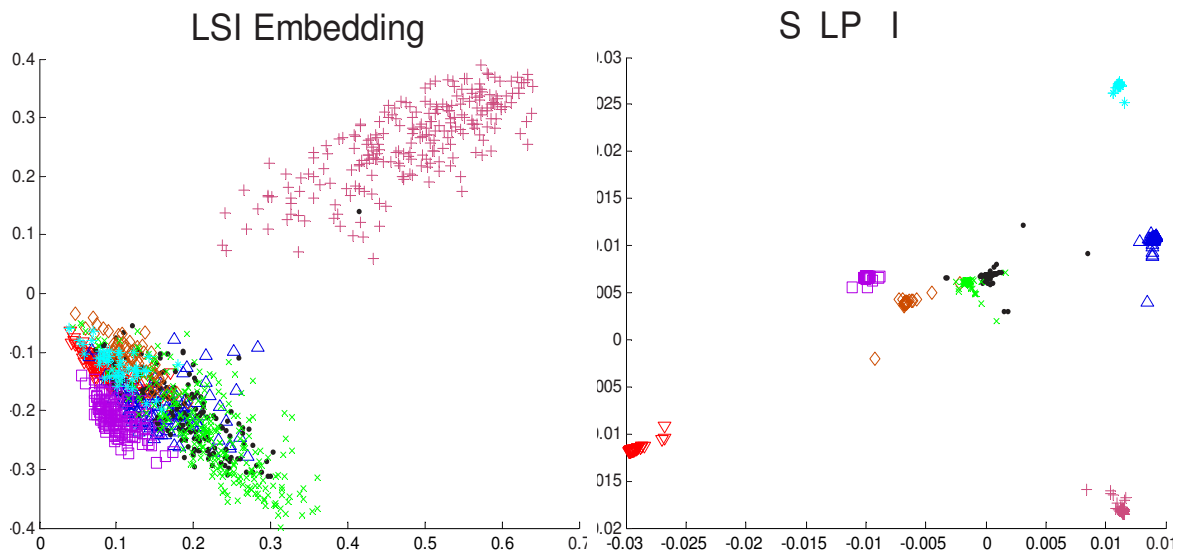


Figure 1: 2 Dimensional visualization of LSI and SLPI on the TDT 2 corpus, each color represents a topic. There are altogether 8 categories in this corpus

Figure (1) showed the 2-D embedding results on the TDT2 corpus. The experiments were conducted on 8 categories. As can be seen, SLPI was more powerful than LSI as to separating the documents with different semantics. From the embedding visualization of LSI, we see that LSI is mainly dominated by the two categories with the most intense variance, while the distribution of the other categories is generally ignored. Therefore, it's very likely that we will get a low error rate and a very low F1 measure in the LSI embedding space. In comparison, SLPI generates a very nice embedding, from the figure,

SLPI considers all the categories fairly well. The tuning parameter  $\pi$  is manually setup as 0.9, which means we will mainly consider the geometric structure information. The experiments on the other settings also show similar patterns.

## 5.5 Text Categorization Performance

To demonstrate how SLPI improves the performance of document categorization, we compared three discrimination methods on two data sets, i.e. Reuters-21578 and TDT2. Note that, SLPI needs to construct a graph on the documents. In this experiment, we set the parameter  $p$  to 15. When constructing the two affinity matrix, we use cosin similarity. For a 5 categories subset, the corresponding matrix is visualized in figure 2

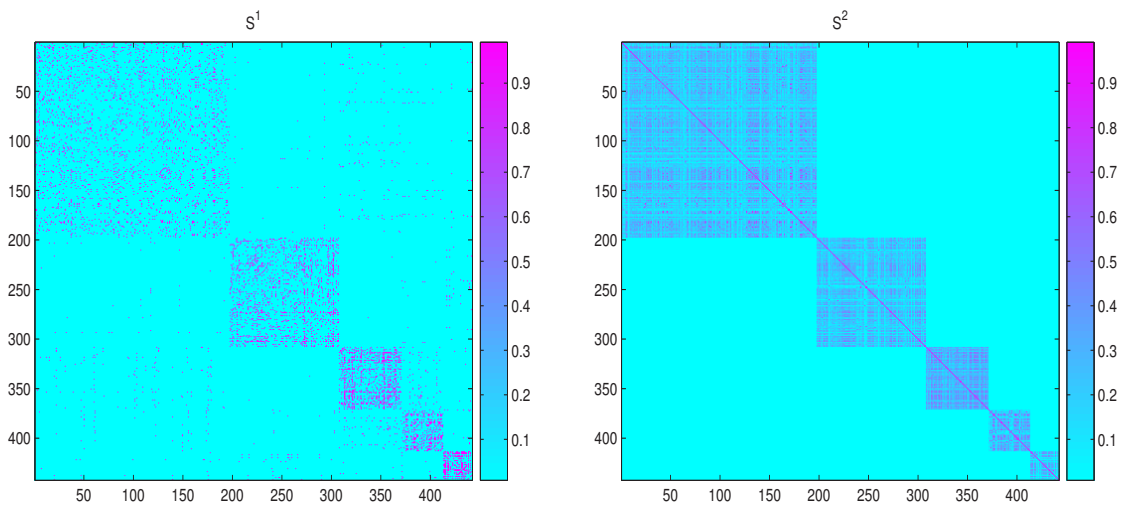


Figure 2: Visualization of the unsupervised component  $S^1$  and the supervised component  $S^2$

From figure 2, we can see that in  $S^1$ , the geometric information can be kept, while categorical information is recorded in  $S^2$ . On the Reuters corpus, we first evaluate the average performance of K Nearest Neighbor classifier. The evaluations were conducted on a small dataset, ranging from 2 to 10. For each given category number  $k$ , 50 tests were conducted on different randomly chosen categories, and the macro-F1, macro-precision, macro-recall were calculated over these 50 tests. For each test, K Nearest Neighbor algorithm was applied 10 times with different value  $K$  and the best result was used. The averaged macro-F1 score of LSI is 0.88, while that of SLPI is 0.92.

Table 3 showed the experimental results using the the Reuters corpus with only 5 categories, there are altogether 3,313 features and 442 documents. From this table, it's

not hard to find that SLPI has the best generalization performance, the LSI generally seeks a performance a little better than discriminant analysis without any dimensionality reduction, this is mainly because some noise may be removed by LSI when the document were projected into a low dimensional space. While SLPI's performance is significantly better than LSI, since the all SLPI based methods'  $F1$  scores are much higher. If a method's  $F$  measures are consistently better than the other's, this method is superior to that one. Another thing to note is that, both LSI and SLPI, they are more effective on the relative "weak" classifiers, such as Naive Bayes. However, for the "strong" classifier, like Logistic Regression, they are not so helpful.

Table 3: Comparison using different dimensionality reduction techniques

Reuter-21578 corpus									
Method	LSI			SLPI			ORG		
	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall
KNN	0.908	0.940	0.902	0.940	0.950	0.922	0.907	0.936	0.903
NB	0.340	0.823	0.231	0.593	0.887	0.445	0.212	0.152	0.589
LR	0.957	0.981	0.939	0.962	0.982	0.940	0.951	0.985	0.927

After conducting LSI or SLPI, how to determine the dimensions of the subspace is still an open problem. In  $k$  category situation, we also compute their best performance on different dimensions in each test. We iterate all the dimensions for the best categorization performance and average all the 50 best results. In real situation, it might not be possible to iterate all the dimensions to get the best performance (denoted as optimal dimension). From our observation, the The optimal dimension in LSI is much higher than that of SLPI. Also, the variance of the dimensions obtained by using LSI is much higher than that obtained by using SLPI. For SLPI, the optimal number is close to  $k - 1$  to  $k + 5$ , where  $k$  is the number of categories. However, for LSI, its optimal dimension is 5 to 10 times larger than that of SLPI, and the variance is almost unpredictable. From this point of view, it demonstrated that SLPI is more powerful than LSI in finding the intrinsic dimensionality of the document space, therefore SLPI is more suitable for text categorization.

**Discussion:** The experimental results showed that LSI seems not to be promising in dimensionality reduction for text categorization because there is almost no significant improvement of the  $F1$  score than that on the original dimension. Since this work is only for a course project, with limited CPU resources and project deadline, we only conduct a very preliminary result to show that SLPI is a promising approach for text categorization. More detailed analysis, including both Macro-scores and Micro-scores, different affinity matrix, different settings of the tuning parameter, is absolutely interesting and necessary. The following work is expected to be finished in one month after this course.

## 6 Conclusions and Future Work

A powerful technique for dimensionality reduction based on Supervised Locality Preserving Indexing is proposed in this paper. Based on the analysis of the theoretical properties of SLPI, analysis on the relation between LSI, SLPI and FLDA indicates that the affinity graph and the tuning parameter is the key to distinguish these algorithms. We also show that under specific parameter settings, the FLDA and LSI could both be viewed as special case of SLPI, which theoretically guarantees the performance of SLPI. A preliminary experiments on Reuters-21578 and TDT2 showed that SLPI works better than LSI based text categorization method. Moreover, the linearity of SLPI makes it more applicable for categorization analysis than the nonlinear approaches.

**Future Work:** Several questions remain to be investigated in our future work: first, even though SLPI is a more flexible framework for dimensionality reduction, it is based on manifold embedding, a further extension to the latent variable probabilistic model may make it more useful, especially for tasks like cross-collection analysis, topic detection, etc. Also, from a Bayesian perspective, the affinity matrix could be viewed as a kind of data dependent priors, how to represent these priors explicitly would be an interesting topic for further exploration.

## Acknowledgement

The author to this paper would like to thank Deng Cai's helpful comments and discussions, Deng is the original author of the LPI method, he is also the coauthor of this interesting project. Also, I would thank professor Chengxiang Zhai and all the students in the CS 598 course, the brainstorming and presentation is really exciting and useful.

## References

- [1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, Vancouver, British Columbia, Canada, 2001.
- [2] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*, Vancouver, British Columbia, Canada, 2003.
- [3] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. 1997.
- [4] David Cohn. Informed projections. In *Advances in Neural Information Processing Systems 15*, Vancouver, British Columbia, Canada, 2002.

- [5] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.
- [7] R.A Fisher. The use of multiple measurements in taxonomic problems. *Annal of Eugenics*, 7:179–188, 1936.
- [8] H. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Series in Statistics, 2001.
- [9] Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. Locality preserving indexing for document representation. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 96–103. ACM Press, 2004.
- [10] Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, 16, 2003.
- [11] J. Lim, J. Ho, M-H. Yang, K-C. Lee, and D. Kriegman. Image clustering with metric, local linear structure and affinity symmetry. In *Proceedings of the 8th European Conference on Computer Vision*, 2004.
- [12] Andrew Y. Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, Vancouver, British Columbia, Canada, 2001.
- [13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000.
- [14] Nathan Srebro and Tommi Jaakkola. Linear dependent dimensionality reduction. In *Advances in Neural Information Processing Systems 16*, Vancouver, British Columbia, Canada, 2003.
- [15] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM Press, 2003.
- [16] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2004.