Locality Preserving Indexing for Document Representation

Xiaofei He
Computer Science Dept.
University of Chicago
xiaofei@cs.uchicago.edu

Deng Cai
Tsinghua University
Beijing, China
cai_deng@yahoo.com

Haifeng Liu
Computer Science Dept.
University of Toronto
hfliu@cs.toronto.edu

Wei-Ying Ma
Microsoft Research Asia
Beijing, China
wyma@microsoft.com

ABSTRACT
Document representation and indexing is a key problem for document analysis and processing, such as clustering, classification and retrieval. Conventionally, Latent Semantic Indexing (LSI) is considered effective in deriving such an indexing. LSI essentially detects the most representative features for document representation rather than the most discriminative features. Therefore, LSI might not be optimal in discriminating documents with different semantics. In this paper, a novel algorithm called Locality Preserving Indexing (LPI) is proposed for document indexing. Each document is represented by a vector with low dimensionality. In contrast to LSI which discovers the global structure of the document space, LPI discovers the local structure and obtains a compact document representation subspace that best detects the essential semantic structure. We compare the proposed LPI approach with LSI on two standard databases. Experimental results show that LPI provides better representation in the sense of semantic structure.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods.*

General Terms
Algorithms, Measurement, Performance, Experimentation, Theory.

Keywords
Locality Preserving Indexing, Latent Semantic Indexing, Document Representation and Indexing, Similarity Measure, Dimensionality Reduction, Vector Space Model

1. INTRODUCTION
Document representation and indexing is a fundamental problem for efficient clustering, classification, and retrieval [1][2] [22]. A document can be represented as a point in the vector space $\mathbf{R}^n$ (given by the number of terms in the documents) [20]. Throughout this paper, we denote by *document space* the set of all document vectors. The document space is typically a subspace of $\mathbf{R}^n$, either linear or non-linear. n is typically very large while the intrinsic dimensionality of the document space might be much lower. Many dimensionality reduction techniques [1][2][5][7][14] [15] have been applied to document representation and indexing. Among these techniques, Latent Semantic Indexing (LSI) [7] by Singular Value Decomposition (SVD) is a well-known successful approach applied to document analysis [11][12] and information retrieval [16].

LSI is essentially a dimensionality reduction technique developed in the context of information retrieval in order to address the problems deriving from the use of synonymous, near-synonymous, and polysemous words as dimensions of document and query representations. Given a term-document matrix X, LSI applies SVD to project the document vectors into a subspace so that cosine similarity can accurately represent semantic similarity. LSI aims to find the best subspace approximation to the original document space in the sense of minimizing the global reconstruction error. In other words, LSI seeks to uncover the most representative features rather the most discriminative features for document representation. Therefore, LSI might not be optimal in discriminating documents with different semantics.

Some variants of LSI have been proposed recently, such as probabilistic LSI (PLSI) [14], iterative residual rescaling (IRR) [2], etc. PLSI is based on the likelihood principle, defines a generative data model, and directly minimizes word perplexity. It can also take advantage of statistical standard methods for model fitting, over-fitting control, and model combination. IRR is an alternative subspace-projection method that outperforms LSI by counteracting its tendency to ignore minority-class documents. This is done by repeatedly rescaling vectors to amplify the presence of documents poorly represented in previous iterations. These methods achieved good empirical results on standard databases. However, similar to the standard LSI, these methods effectively see only the global structure of the document space while in many cases the local structure is more important. Moreover, these methods do not take into account the discriminating structure which might be the most important for real world applications.

In this paper, we propose a new approach called Locality Preserving Indexing (LPI) to document representation, which aims to discover the *local* geometrical structure of the document space. Since the neighboring documents (data points in high dimensional space) probably relate to the same topic, LPI can have more discriminating power than LSI even though LPI is also unsupervised. To be specific, an adjacency graph is constructed to model the local structure of the document space. A semantic space for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
Copyright 2004 ACM 1-58113-881-4/04/0007...$5.00.

document representation is learned by using Locality Preserving Projections (LPP) [13] which is a recently proposed algorithm for linear dimensionality reduction. From the perspective of discrimination, we provide a theoretical analysis to show that LPP is an optimal approximation to Linear Discriminant Analysis (LDA) [10] based on the assumption that the neighboring documents are probably related to the same topic.

It is worthwhile to highlight several aspects of our proposed algorithm here:

1. By using SVD, LSI effectively sees only the linear structure of the document space. In contrast, LPI is capable of discovering the nonlinear structure of the document space to some extent due to its locality preserving character [13].
2. While the document space is generally embedded in an ambient space $\mathbf{R}^n$, there is no convincing evidence that the document space is Euclidean, or *flat*. Therefore, it is more natural and reasonable to assume that the document space is a manifold, either linear or nonlinear. In this sense, LPI is particularly applicable since it essentially discovers the local geometrical structure of the data manifold. Specifically, LPI is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the data manifold [4][13].
3. In [13], some examples on synthetic and real world data sets show that LPP has more discriminating power than PCA. SVD is similar in spirit to PCA when it is used for dimensionality reduction. Consequently, LPI has more discriminating power than LSI. It can be used as a preprocessing for document clustering, classification and retrieval.

The rest of this paper is organized as follows: Section 2 describes the Locality Preserving Projections for learning a semantic subspace. Section 3 introduces Locality Preserving Indexing for document representation. Theoretical analysis of LPP and its connections to LDA are discussed in Section 4. The experimental results are shown in Section 5. Finally, we provide concluding remarks and future work in Section 6.

2. LEARNING A SEMANTIC SUBSPACE

In this section, we give a brief description of Locality Preserving Projections (LPP) [13], the core algorithm used for document representation and indexing in this paper. Different from LSI which assumes that the document space is a Euclidean space, LPI assumes that the document space is a manifold. Note that, Euclidean space is actually a linear manifold which is a special manifold.

Let $\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_m$ denote the set of document vectors in $\mathbf{R}^n$. We denote by X the term-document matrix whose column vectors are documents. Let $\mathbf{w}$ denote the transformation vector. Thus, the optimal projections preserving locality can be obtained by solving the following minimization problem [13]:

$$\min_{\mathbf{w}} \sum_{ij} \left(\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{x}_j\right)^2 S_{ij} \qquad (1)$$

where S_{ij} evaluate the local structure of the document space. It can be simply defined as follows:

$$S_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_j \\ & \text{or } \mathbf{x}_j \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

The objective function with our choice of symmetric weights S_{ij} ($S_{ij} = S_{ji}$) incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close" then y_i (= $\mathbf{w}^T\mathbf{x}_i$) and y_j (= $\mathbf{w}^T\mathbf{x}_j$) are close as well. S_{ij} can be thought of as a similarity measure between objects. Some more sophisticated definition of S can be found in [13]. The objective function can be reduced to:

$$\begin{aligned} & \frac{1}{2}\sum_{ij}(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j)^2 S_{ij} \\ &= \sum_i \mathbf{w}^T\mathbf{x}_i D_{ii} \mathbf{w}^T\mathbf{x}_i - \sum_{ij} \mathbf{w}^T\mathbf{x}_i S_{ij} \mathbf{w}^T\mathbf{x}_j \\ &= \mathbf{w}^T X(D-S)X^T\mathbf{w} \\ &= \mathbf{w}^T XLX^T\mathbf{w} \end{aligned} \qquad (3)$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, and D is a diagonal matrix; its entries are column (or row, since S is symmetric) sums of S, $D_{ii} = \sum_j S_{ji}$.. $L = D - S$ is the Laplacian matrix [6]. Matrix D provides a natural measure on the data points. The bigger the value D_{ii} (corresponding to y_i) is, the more "important" is y_i. Therefore, we impose a constraint as follows:

$$\begin{aligned} & \mathbf{y}^T D\mathbf{y} = 1 \\ \Rightarrow\ & \mathbf{w}^T XDX^T\mathbf{w} = 1 \end{aligned} \qquad (4)$$

Finally, the minimization problem reduces to finding:

$$\arg\min_{\substack{\mathbf{w} \\ \mathbf{w}^T XDX^T\mathbf{w}=1}} \mathbf{w}^T XLX^T\mathbf{w} \qquad (5)$$

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

$$XLX^T\mathbf{w} = \lambda XDX^T\mathbf{w} \qquad (6)$$

Note that the two matrices XLX^T and XDX^T are both symmetric and positive semi-definite. Also, the obtained projections $\mathbf{w}$ are actually the optimal linear approximation to the eigenfunctions of the Laplace Beltrami operator on the manifold [13]. Therefore, LPI is capable of discovering the intrinsic manifold structure to some extent.

3. LOCALITY PRESERVING INDEXING

3.1 The Problem

The problem of document indexing and representation is the following. Given a set of documents $\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_m$ in $\mathbf{R}^n$, find a lower dimensional representation $\mathbf{y}_i$ of $\mathbf{x}_i$ such that $\| \mathbf{y}_i - \mathbf{y}_j \|$ reflects the semantic relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. Here, we assume that the documents reside on a linear subspace or non-linear submanifold of $\mathbf{R}^n$.

3.2 The Algorithm

In Section 2, we have described LPP, a method for learning a locality preserving subspace. Based on LPP, we describe our method for document representation and indexing.

In the document analysis and processing problems one is often confronted with the fact that the dimension of the document vector (n) is much larger than the number of documents (m). Thus, the $n \times n$ matrix XDX^{T} is singular. Also, when n is very large, the computational complexity of the eigenproblem (6) is high. To overcome these computational problems, we first project the documents to a PCA subspace so that the resulting matrix XDX^{T} is nonsingular and with small dimensions. Another consideration of using PCA as preprocessing is for noise reduction. The algorithmic procedure of LPI is stated below:

1. **PCA Projection:** We project the document set $\{\mathbf{x}_i\}$ into the PCA subspace by throwing away the smallest principal components. We denote the transformation matrix of PCA by W_{PCA}.
2. **Constructing the adjacency graph**: Let G denote a graph with n nodes. The i^{th} node corresponds to the document $\mathbf{x}_i$. 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 k nearest neighbors of $\mathbf{x}_i$ or $\mathbf{x}_i$ is among k nearest neighbors of $\mathbf{x}_j$. Note that, if the documents have been classified into different semantic classes, one might construct an adjacency graph based on the class labels. That is, we can put an edge between two nodes if and only if they have the same class label.
3. **Choosing the weights**: If node i and j are connected, put

$$S_{ij} = \mathbf{x}_i^T \mathbf{x}_j \tag{7}$$

   Otherwise, put $S_{ij} = 0$. The weight matrix S of graph G models the local structure of the document space.

4. **Eigenmap**: Compute the eigenvectors and eigenvalues for the generalized eigenvector problem:

$$XLX^T\mathbf{w} = \lambda XDX^T\mathbf{w} \tag{8}$$

   where D is a diagonal matrix whose entries are column (or row, since S is symmetric) sums of S, $D_{ii} = \sum_j S_{ji}$. $L = D - S$ is the Laplacian matrix. The i^{th} column of matrix X is $\mathbf{x}_i$.

Let $W_{LPP} = [\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{k-1}]$ be the solutions of equation (8), ordered according to their eigenvalues, $\lambda_0<\lambda_1<\cdots<\lambda_{k-1}$. Thus, the embedding is as follows:

$$\begin{aligned} \mathbf{x} &\rightarrow \mathbf{y} = W^T\mathbf{x} \\ W &= W_{PCA}W_{LPP} \end{aligned} \tag{9}$$

where $\mathbf{y}$ is a k-dimensional representation of the document $\mathbf{x}$. W is the transformation matrix. This linear mapping best preserves the manifold's estimated intrinsic geometry.

4. THEORETICAL ANALYSIS

As we described earlier, LSI is fundamentally based on SVD while LPI is fundamentally based on LPP [13]. In this section, we give a theoretical analysis of LSI and LPI. We begin with a brief review of SVD.

4.1 Singular Value Decomposition

Let $X = [\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_m]^{\mathrm{T}}$ be a $n \times m$ data matrix, $\mathbf{x}_i \in \mathbf{R}^{\mathbf{n}}$. By SVD, X can be decomposed as follow:

$$X = USV^{\mathrm{T}} \tag{10}$$

where $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ and $\mathbf{u}_i$ is the eigenvector of XX^T, $\mathbf{u}_i \in \mathbf{R}^{\mathbf{n}}$. $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ and $\mathbf{v}_j$ is the eigenvector of X^TX, $\mathbf{v}_j \in \mathbf{R}^{\mathbf{m}}$. S is a $n \times m$ matrix such that $\sigma_i = S_{ii}$ is the i^{th} largest singular value and $S_{ij} = 0$ for $i \neq j$. $\mathbf{u}_i$ and $\mathbf{v}_j$ are called left singular vectors and right singular vectors, respectively. The following points are well known or easily derived in matrix theory:

1. σ_i is the square root the i^{th} largest eigenvalues of XX^{T} (or, $X^{\mathrm{T}}X$).
2. U and V are both orthonormal matrices in that $U^{\mathrm{T}}U=UU^{\mathrm{T}}=I$ and $V^{\mathrm{T}}V=VV^{\mathrm{T}}=I$.
3. $\mathbf{u}_i$ ($i = 1,2,\ldots,n$) forms a orthonormal basis for the input space $\mathbf{R}^{\mathbf{n}}$. The matrix U can be thought of as a rotation transformation.

Clearly, the number of non-zero singular values is determined by the rank of X, say k. Thus, equation (10) can be rewritten as follows:

$$X = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}} \tag{11}$$

Thus, the document vectors can be reduced to a k-dimensional subspace spanned by $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ without losing any information. Also, it is easy to see that $\sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}$ is the best rank-p ($p \leq k$) approximation to X in terms of Frobenius matrix norm. See [3][9][18] for theoretical interpretations of LSI using SVD.

4.2 Geometry of Document Space --- Global vs. Local

From section 4.1 we see that the projection of LSI can be simply computed by using SVD. The left singular vectors $\mathbf{u}_i$ are the basis functions of the reduced subspace.

Let A be a transformation matrix. The low dimensional representation of X is $Y = A^{\mathrm{T}}X$. The optimal linear transformation which preserves inner product can be obtained by solving the following minimization problem:

$$\min_A \left\| X^{\mathrm{T}}X - Y^{\mathrm{T}}Y \right\|_F \tag{12}$$

where $\|M\|_F$ is the Frobenius matrix norm such that $\|M\|_F = \sqrt{\sum_{ij} M_{ij}^2}$. By SVD, we have $X^{\mathrm{T}}X = VSU^{\mathrm{T}}USV^{\mathrm{T}} = VS^2V^{\mathrm{T}}$. Therefore, $Y = S_kV_k^T$ is the k-dimensional representation of X which best preserves inner product, where $S_k = diag(\sigma_1, \ldots, \sigma_k)$ and $V_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$. Correspondingly, the optimal transformation preserving inner product is $A = U_k$ where $U_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$.

LSI uses the matrix U_k to perform linear dimensionality reduction. In document analysis, inner product is one of the most frequently used similarity measures to discover the semantic structure of the document space. It preserves the global structure in spirit. How-

ever, in many real world applications, the local structure is more important especially when nearest neighbor search is involved. Moreover, LSI aims to discover the linear subspace on which the documents possibly reside. However, there is no convincing evidence that the document space is actually a linear subspace of the input space. A more naturally and reasonable assumption is that the document space is a sub-manifold embedded in the ambient space. It can be either linear or non-linear. Recently there have been some renewed interests in the problem of developing low dimensional representations when data arises from sampling a probability distribution on a manifold. Some typical manifold learning algorithms include ISOMAP [21], Locally Linear Embedding [19], Laplacian Eigenmaps [4], Locality Preserving Projections (LPP, [13]), etc. All of them aim to discover the local manifold structure. The former three are non-linear algorithms while the last one is linear. Also, the maps obtained by the former three are only defined on the training data points, while the maps obtained by LPP are defined everywhere. In this paper, we apply LPP to learn a low dimensional semantic space for document representation.

4.3 Discriminant Analysis of LPP

Traditionally, document indexing and representation have been explored extensively from the perspectives of geometry and statistics. We see little discriminant analysis for document indexing. One reason is that, for document indexing and representation, the labels of the documents are not available. When the labels are available, we can apply Linear Discriminant Analysis (LDA) 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. LDA is optimal in the sense of discrimination. In this subsection, we show that LPP provides an optimal approximation to LDA. Our basic assumption is that the neighboring documents are probably related to the same topic.

Suppose the documents belong to l classes. LDA can be obtained by solving the following maximization problem [10]:

$$\begin{aligned}
&\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \frac{\left|\mathbf{w}^T S_B \mathbf{w}\right|}{\left|\mathbf{w}^T S_W \mathbf{w}\right|} \\
&S_B = \sum_{i=1}^{l} n_i \left(\mathbf{m}^{(i)} - \mathbf{m}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}\right)^T \\
&S_W = \sum_{i=1}^{l} n_i E\left[\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)^T\right]
\end{aligned} \tag{13}$$

which leads to the following eigenvector problem:

$$S_B \mathbf{w} = \lambda S_W \mathbf{w} \tag{14}$$

where $\mathbf{w}$ is the transformation vector. $\mathbf{m}$ is the total sample mean vector. n_i is the number of samples in the i^{th} class. $\mathbf{m}^{(i)}$ are the average vectors of the i^{th} class, and $\mathbf{x}^{(i)}$ is the random vector associated to the i^{th} class. We call S_W the *within-class scatter matrix* and S_B the *between-class scatter matrix*.

We can rewrite the matrix S_W as follows:

$$\begin{aligned}
S_W &= \sum_{i=1}^{l} n_i E\left[\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)^T\right] \\
&= \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{x}^{(i)} - \mathbf{m}^{(i)}\right)^T\right) \\
&= \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\left(\mathbf{x}^{(i)}(\mathbf{x}^{(i)})^T - 2\mathbf{m}^{(i)}(\mathbf{x}^{(i)})^T + \mathbf{m}^{(i)}(\mathbf{m}^{(i)})^T\right)\right) \\
&= \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\mathbf{x}^{(i)}(\mathbf{x}^{(i)})^T - n_i\mathbf{m}^{(i)}(\mathbf{m}^{(i)})^T\right) \\
&= \sum_{i=1}^{l} X_i L_i X_i^T
\end{aligned} \tag{15}$$

where $X_i L_i X_i^T$ is the data covariance matrix of the i^{th} class. $X_i = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ is a $d \times n_i$ matrix whose column vectors belong to the i^{th} class, and $L_i = I - 1/n_i\ \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}}$ is a $n_i \times n_i$ matrix where I is the identity matrix and $\mathbf{e}_i = (1,1,\ldots,1)^{\mathrm{T}}$ is a n_i dimensional vector. To further simplify the above equation, we define:

$$\begin{aligned}
&X = \left(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\right) \\
&W_{ij} = \begin{cases} 1/n_k & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } k^{th} \text{ class} \\ 0 & \text{otherwise} \end{cases} \\
&L = I - W
\end{aligned} \tag{16}$$

Thus, we get:

$$S_W = XLX^T \tag{17}$$

Let $\mathbf{e} = (1,1,\ldots,1)^{\mathrm{T}}$ be a n dimensional vector. Similarly, we can compute the matrix S_B as follows:

$$\begin{aligned}
S_B &= \sum_{i=1}^{l} n_i \left(\mathbf{m}^{(i)} - \mathbf{m}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}\right)^T \\
&= \sum_{i=1}^{l} n_i \mathbf{m}^{(i)}(\mathbf{m}^{(i)})^T - 2\mathbf{m}\left(\sum_{i=1}^{l} n_i \mathbf{m}^{(i)}\right) + \left(\sum_{i=1}^{l} n_i\right)\mathbf{m}\mathbf{m}^T \\
&= XWX^T - 2n\mathbf{m}\mathbf{m}^T + n\mathbf{m}\mathbf{m}^T \\
&= XWX^T - n\mathbf{m}\mathbf{m}^T \\
&= XWX^T - X(\tfrac{1}{n}\mathbf{e}\mathbf{e}^T)X^T \\
&= X(W - \tfrac{1}{n}\mathbf{e}\mathbf{e}^T)X^T \\
&= X(W - I + I - \tfrac{1}{n}\mathbf{e}\mathbf{e}^T)X^T \\
&= -XLX^T + X(I - \tfrac{1}{n}\mathbf{e}\mathbf{e}^T)X^T \\
&= -XLX^T + C
\end{aligned} \tag{18}$$

where $C = X(I - \frac{1}{n}\mathbf{e}\mathbf{e}^T)X^T$ is the data covariance matrix. Note that, if the data has a zero mean, we can write the covariance matrix as $C = XX^T$. Thus, the projections of LDA are given by the maximum eigenvalue solutions to the following generalized eigenvector problem:

category	num of documents
earn	3713
acq	2055
crude	321
trade	298
money-fx	245
interest	197
ship	142
sugar	114
coffee	110
gold	90
money-supply	87
gnp	63
cpi	60
cocoa	53
alum	45
grain	45
copper	44
jobs	42
reserves	38
rubber	38

Figure 1. 20 semantic categories from Reuters-21578 used in our experiments.

$$\begin{aligned} & S_B\mathbf{w} = \lambda S_W\mathbf{w} \\ \Rightarrow\ & (C - XLX^T)\mathbf{w} = \lambda XLX^T\mathbf{w} \\ \Rightarrow\ & C\mathbf{w} = (1+\lambda)XLX^T\mathbf{w} \end{aligned} \tag{19}$$

which is equivalent to the minimum eigenvalue solutions to the following eigenproblem:

$$XLX^T\mathbf{w} = \lambda C\mathbf{w} \tag{20}$$

Based on our assumption that neighboring documents are probably related to the same topic, the similarity matrix S in (2) gives an optimal approximation to the similarity matrix W in (16). Also, the covariance matrix C can be approximated by the matrix XDX^T since D is a diagonal matrix and $D \approx I$. Therefore, even though the label information is not available, we can still discover the discriminating structure to some extent by using LPP.

5. EXPERIMENTAL RESULTS

In this section, several experiments were performed to show the effectiveness of our proposed algorithm. Two standard document collections were used in our experiments, *i.e* Reuters-21578 and TDT2. We compared our proposed algorithm LPI with LSI.

5.1 Similarity Evaluation on Reuters-21578

The accuracy of similarity measure plays a crucial role in most of the information processing tasks, such as document clustering, classification, retrieval, etc. In this subsection, we evaluate the accuracy of similarity measure using two different indexing algorithms, *i.e.* LPI and LSI. The similarity measure we used is the cosine similarity.

	baseline		LSI		LPI	
Doc. Subsets	dims	AvP(%)	dims	Δ AvP	dims	Δ AvP
agreement	753	65.78	6	1.33	4	16.95
american	337	55.94	6	0.03	5	14.3
bank	780	36.91	8	6.36	10	10.51
control	226	78.95	209	0.59	12	**-2.26**
domestic	246	56.75	235	0.22	5	**-0.27**
export	254	70.1	254	0	5	5.46
exports	295	53.74	290	0.02	8	5.54
five	761	72.42	538	0.74	60	**-13**
foreign	456	44.21	451	0.01	8	3.98
growth	319	46.16	73	2.61	5	10.14
income	333	81.83	177	0.89	60	**-1.9**
increase	548	46.77	488	0.11	19	1.46
industrial	232	49.16	7	3.38	2	16.42
industry	361	50.67	355	0.02	5	13.62
international	679	51.08	6	1.67	5	21.43
investment	527	68.39	527	0	5	**-2.27**
losses	234	88.27	132	0.95	45	**-1.63**
money	247	53.11	231	0.06	10	**-12**
national	379	37	15	9.82	9	20.28
prices	551	57.16	550	0	6	10.19
production	335	57.85	330	0.01	6	10.06
public	282	56.3	274	0.01	9	**-3.44**
rates	300	51.65	290	0.04	4	**-4.42**
report	299	62.01	299	0	8	6.43
services	234	55.18	4	4.26	4	15.96
sources	256	52.74	252	0.03	8	11.68
talks	274	68.72	272	0	6	15.08
tax	594	95.67	245	0.16	35	**-6.11**
trade	550	47.09	512	0.29	5	**-0.7**
world	349	61.26	344	0.01	9	9.01

Figure 2. Precision improvement and the corresponding dimensionality for each document subset using the Reuters-21578 document corpus. The first column contains the keywords generating the document subset. The LPI, LSI and baseline algorithms are compared.

5.1.1 Data Preparation

Reuters-21578 is used as our data collection. Documents that appear in two or more categories were removed, thus leaving us with 8293 documents. The collection consists of 65 semantic categories (topics). The numbers of documents in different categories range from 1 to 3713. We kept the largest 20 categories which contain 7800 documents in total, as listed in Figure 1. From the <title> field of 300 TREC ad hoc topics (topic 251-550), we chose 30 keywords that appear in our data collection with highest frequencies, say, q_i (i = 1, 2, …, 30). For each keyword q_i, let D_i denote the set of the documents containing q_i. Let $D = D_1 \cup \ldots \cup D_{30}$. Finally, we get 30 document subsets and each subset contains multiple topics. Note that, these subsets are not necessarily disjoint. The numbers of documents of these 30 document subsets ranged from 226 to 802 with an average of 408, and the number of topics ranged from 12 to 20 with an average of 17.5. We removed the stop words. No further preprocessing was done. For

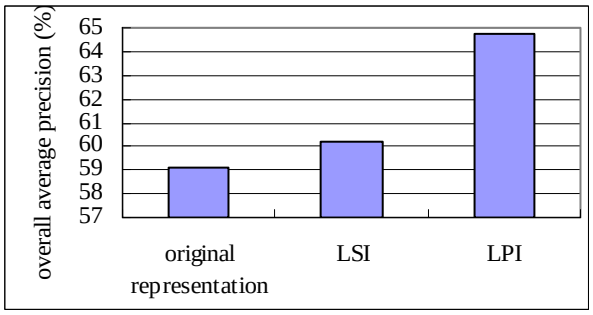

Figure 3. The overall average precision using the Reuters-21578 document corpus.

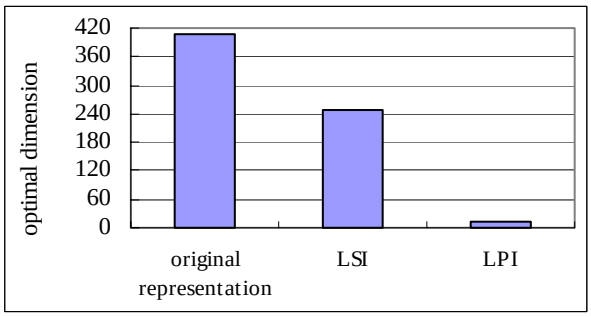

Figure 4. The optimal dimensions of the original representation space, LSI subspace and LPI subspace using the Reuters-21578 document corpus.

the i^{th} subset, the documents are represented as vectors in a n_i dimensional vector space using the Term Frequency (TF) indexing scheme. The reason for generating such 30 document subsets is to split the data collection into small subsets so that we can compare our algorithm to LSI on each subset. In fact, the keywords can be thought of as queries in information retrieval. Thus, the comparison can be thought of as being performed on different queries.

5.1.2 Experimental Design

For the original document set D, we compute its lower dimensional representations D_{LPI} and D_{LSI} by using LPI and LSI respectively. Similarly, D_{LPI} consists of 30 subsets, $D_{LPI,1}$, …, $D_{LPI,30}$. D_{LSI} also consists of 30 subsets, $D_{LSI,1}$, …, $D_{LSI,30}$. We take the number of nearest neighbors for the LPI algorithm to be 7.

For each document subset D_i (or, $D_{LPI,i}$, $D_{LSI,i}$), we evaluate the similarity measure between the documents in D_i. Intuitively, we expect that similarity should be higher for any document pair related to the same topic (intra-topic pair) than for any pair related to different topics (cross-topic pair). Therefore, we adopted the *average precision* used in TREC [1], regarding an intra-topic pair as a relevant document and the similarity value as the ranking score. Specifically, we denote by p_i the document pair which has the i^{th} highest similarity value among all pairs of documents in the document set D_i. For each *intra-topic pair* p_k, its precision is evaluated as follows:

$$precision(p_k) = \frac{\#\text{ of intra-topic pairs } p_j \text{ where } j \le k}{k} \quad (21)$$

The average of the precision values over all intra-topic pairs in D_i was computed as the average precision of D_i. Note that, the definition of precision (21) we used here is the same as that used in [1].

5.1.3 Results

The experimental results are reported in this section. We compared LPI (corresponding to document set D_{LPI}) to LSI (corresponding to document set D_{LSI}) and the original document representation (corresponding to document set D as baseline algorithm). In general, the performance of LPI and LSI varies with the number of dimensions. We showed the best results obtained by them. For each document subset, Figure 2 listed the average precision and the dimensionality by using the baseline algorithm, and the precision improvement and the dimensionality by using LPI and LSI. In our experiments, the number of terms (m > 6,000) is larger than the number of documents (n). So for the baseline algorithm, we reduced the document space to an n-dimensional subspace using SVD without losing any information. As can be seen, LPI achieved higher accuracy than LSI on 19 document subsets, while it failed on the other 11 subsets.

Figure 3 showed the average of the "best average precisions". The overall average precisions for LPI, LSI and baseline algorithms are 64.78%, 60.22%, and 59.10%, respectively. LSI achieved little improvement (1.12%) over the baseline algorithm, while LPI achieved 5.68% improvement. Moreover, the average dimension in the original representation space of the 30 document subsets is 399.7. The average optimal dimensions for LSI and LPI are 246 and 12.7, respectively, as shown in Figure 4.

5.2 Similarity Evaluation on TDT2

In this subsection, we evaluated the accuracy of similarity measures using the TDT2 document dataset[*].

5.2.1 Data Preparation

The TDT2 corpus 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 experiment, those documents appearing in two or more categories were removed, and only the largest 20 categories were kept, thus leaving us with 8741 documents in total. The sizes of these 20 categories are as follows: 1844, 1828, 1222, 811, 441, 407, 272, 238, 226, 167, 160, 145, 141, 140, 131, 123, 123, 120, 104, and 98. Using the same strategy described in Section 5.1.1, we split this data collection into 30 subsets. The numbers of documents of these 30 document subsets ranged from 256 to 805 with an average of 507, and the number of topics ranged from 6 to 20 with an average of 16.7.

5.2.2 Performance Evaluations and Comparisons

In this test, we used the same experimental design as described in Section 5.1.2. We evaluated the accuracy of similarity measure in the LPI subspace, LSI subspace and original representation space. Figure 5 listed the average precision and dimensionality for each document subset by using LPI, LSI and the baseline algorithms.

[*] Nist Topic Detection and Tracking corpus at http://www.nist.gov/speech/tests/tdt/tdt98/index.html

As can be seen, LPI achieved higher accuracy than LSI on all the document subsets.

Figure 6 showed the average of the "best average precisions". The overall average precisions for LPI, LSI and baseline algorithms are 90.6%, 81.5%, and 81.4%, respectively. As can be seen, LPI significantly outperformed LSI on this data corpus. The average optimal dimensions for LPI, LSI and baseline are 9.1, 469.6, and 488.6, respectively, as shown in Figure 7. LPI successfully encoded the discriminating information in a very low dimensional subspace.

	baseline		LSI		LPI	
Doc. Subsets	dims	AvP(%)	dims	Δ AvP	dims	Δ AvP
air	632	90.62	596	0.07	6	6.5
british	502	90.66	470	0.04	3	3.64
building	346	63.36	344	0.01	9	14.91
control	629	64.13	624	0	6	12.92
cooperation	337	91.73	337	0	4	5.19
court	715	93.91	670	0.02	9	4.19
decision	743	78.92	734	0	9	12.89
domestic	406	81.68	351	0.35	4	11.74
drug	272	91.87	254	0.35	4	6.76
fire	308	79.06	300	0.06	7	6.93
food	482	82.27	482	0	5	6.2
growth	554	92.34	360	1.16	29	1.21
health	429	83.82	424	0	9	8.29
history	419	73.62	396	0.3	11	21.03
human	383	65.4	372	0.04	7	13.58
impact	417	79.55	415	0	9	16.39
information	617	91.94	616	0	8	5.35
legal	542	94.61	541	0	7	3.96
material	735	80.99	735	0	11	10.41
money	770	70.43	770	0	7	17.63
peace	583	79.38	550	0.01	7	8.57
police	473	63.24	470	0.01	8	5.76
robert	337	85.38	337	0	6	10.67
russia	595	95.14	590	0	3	3.64
school	422	68.54	422	0	8	12.33
smoking	247	99	173	0.05	51	0.23
technology	332	78.9	332	0	4	7.04
trade	596	75.64	596	0	6	16.07
violence	337	72.66	337	0	7	11.39
women	499	83.08	491	0.01	9	9.41

Figure 5. Precision improvement and the corresponding dimensionality for each document subset using the TDT2 document corpus. The first column contains the keywords generating the document subset. The LPI, LSI and baseline algorithms are compared.

5.3 Discussion

The experiments above reveal a number of interesting points:

1. Both LPI and LSI performed better in the optimal subspace than in the original space.
2. The optimal representation subspace obtained by LPI has smaller dimensionality (12.7 for Reuters-21578 and 9.1 for TDT2) than LSI (246 for Reuters-21578 and 469.6 for TDT2) while LPI achieved much better results. This shows that LPI is more powerful than LSI as to discovering the intrinsic dimensionality of the document space. One reason is that, by preserving the local structure of the document space, LPI can discover the discriminating structure to some extent, as we show in Section 4.
3. The low dimensionality of the document subspace obtained in our experiments show that dimensionality reduction is indeed necessary as a preprocessing for document clustering, classification, retrieval, etc.
4. The improvement achieved by LPI on the TDT2 corpus is much higher than that on the Reuters-21578 corpus. This is probably because that the TDT2 has a higher baseline performance. Therefore, our assumption that the neighboring documents are probably related to the same topic can hold with high probability.

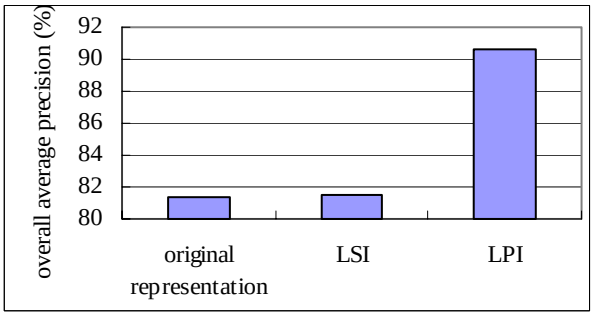

Figure 6. The overall average precision using the TDT2 document corpus.

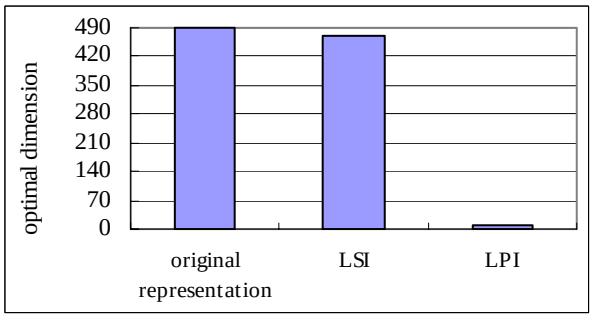

Figure 7. The optimal dimensions of the original representation space, LSI subspace and LPI subspace using the TDT2 document corpus.

6. CONCLUSIONS

A novel algorithm called Locality Preserving Indexing is proposed in this paper. Different from LSI which discovers the linear structure of the document space, LPI is capable of discovering the non-linear structure of the document space to some extent. The locality preserving character makes LPI insensitive to noise and outliers. Theoretical analysis of LPI and its connections to LDA are provided. Based on the assumption that neighboring docu-

ments are probably related to the same topic, we conclude that LPI is an optimal approximation to LDA when the label information is not available. Therefore, even though LPI is unsupervised, it can to some extent discover the discriminating structure of the document space. Experimental results on Reuters-21578 and TDT2 show the effectiveness of our algorithm.

Though dimensionality reduction has proved to be useful, it remains unclear how to estimate the intrinsic dimensionality of the document space. SVD is guaranteed to discover the intrinsic dimensionality if the document space is linear. However, there is no convincing evidence that it is actually linear. Also, LPI seems to be superior to LSI for similarity measure, as shown in our experiments. However, it is unclear how LPI works in the real world applications, such as information retrieval.

7. REFERENCES

[1] R. K. Ando, "Latent Semantic Space: Iterative Scaling improves precision of inter-document similarity measurement", in *Proc. of the 23th International ACM SIGIR*, Athens, Greece, 2000.

[2] R. K. Ando, and L. Lee, "Iterative Residual Rescaling: An Analysis and Generalization of LSI", in *Proc. of the 24th International ACM SIGIR*, New Orleans, LA, 2001.

[3] B. T. Bartell, G. W. Cottrell, and R. K. Belew, "Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling", in *Proc. of 15th International ACM SIGIR*, Copenhagen, Denmark, 1992.

[4] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering", *Advances in Neural Information Processing Systems 14*, Vancouver, Canada, 2001.

[5] E. Bingham and H. Mannila, "Random Projection in dimensionality reduction: applications to image and text data", *Proc. Of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 245-250, 2001.

[6] Fan R. K. Chung, *Spectral Graph Theory*, Regional Conferences Series in Mathematics, number 92, 1997.

[7] 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.

[8] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag New York, Inc., 1996.

[9] C. H. Ding, "A similarity-based probability model for Latent Semantic Indexing", in *Proc. of the 22th International ACM SIGIR*, 1999.

[10] Richard O. Duda, Peter E. Hart and David G. Stork, Pattern Classification (2nd Edition), Wiley-Interscience, 2000.

[11] S. T. Dumais and J. Nielsen, "Automating the assignment of submitted manuscripts to reviewers", *in Proc. of the 15th ACM SIGIR*, Copenhagen, Denmark, 1992.

[12] P. W. Foltz and S. T. Dumais, "Personalized information delivery: An analysis of information filtering methods", *Communications of the ACM*, 35(12):51-60, 1992.

[13] Xiaofei He and Partha Niyogi, "Locality Preserving Projections", in *Advances in Neural Information Processing Systems 16*, Vancouver, Canada, 2003.

[14] T. Hofmann, "Probabilistic Latent Semantic Indexing", in *Proc. of the 22th International ACM SIGIR*, Berkeley, California, 1999.

[15] C. L. Isbell and P. Viola, "Restructuring Sparse High Dimensional Data for Effective Retrieval", *Advances in Neural Information Systems*, 1999.

[16] T. G. Kolda and D. P. O'Leary, "A Semidiscrete matrix decomposition for latent semantic indexing in information retrieval", *ACM Transactions on Information Systems*, 16(4):322-346, 1998.

[17] K. Lang, "Learning to filter netnews", *Proc. Of the 12th Int. Conf. on Machine Learning*, 1995.

[18] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: a probabilistic analysis," in *Proc. 17th ACM Symp. Principles of Database Systems*, Seattle, 1998.

[19] S. T. Roweis, L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, vol 290, 22 December 2000.

[20] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[21] J. B. Tenenbaum, Vin De Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, Vol 290, 22 December 2000.

[22] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization", in *Proc. of the 26th International ACM SIGIR*, Toronto, Canada, 2003.