Orthogonal Locality Preserving Indexing

Deng Cai
Computer Science Department
University of Illinois at Urbana Champaign
dengcai2@cs.uiuc.edu

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

ABSTRACT

We consider the problem of document indexing and representation. Recently, Locality Preserving Indexing (LPI) was proposed for learning a compact document subspace. Different from Latent Semantic Indexing which is optimal in the sense of global Euclidean structure, LPI is optimal in the sense of local manifold structure. However, LPI is extremely sensitive to the number of dimensions. This makes it difficult to estimate the intrinsic dimensionality, while inaccurately estimated dimensionality would drastically degrade its performance. One reason leading to this problem is that LPI is non-orthogonal. Non-orthogonality distorts the metric structure of the document space. In this paper, we propose a new algorithm called Orthogonal LPI. Orthogonal LPI iteratively computes the mutually orthogonal basis functions which respect the local geometrical structure. Moreover, our empirical study shows that OLPI can have more locality preserving power than LPI. We compare the new algorithm to LSI and LPI. Extensive experimental results show that Orthogonal LPI obtains better performance than both LSI and LPI. More crucially, it is insensitive to the number of dimensions, which makes it an efficient data preprocessing method for text clustering, classification, retrieval, etc.

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

Orthogonal Locality Preserving Indexing, Locality Preserving Indexing, Document Representation and Indexing, Similarity Measure, Dimensionality Reduction, Vector Space Model

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

1. INTRODUCTION

There are two fundamental problems in document processing: how to represent the documents and how to evaluate their similarity. If we denote by *document space* the set of all the documents, different indexing algorithms see different structures of the document space. The Vector Space Model (VSM) might be one of the most popular model for document representation. Each document is represented as a *bag of words*. Correspondingly, the document space is associated with a Euclidean structure and the inner product (or, cosine similarity) is used as the standard similarity measure for documents. Unfortunately, VSM suffers from some problems such as *synonymy* and *polysemy*.

Data representation is fundamentally related to the problem of manifold learning [4][18][20] which is an emerging research area. Given a set of high-dimensional data points, manifold learning techniques aim at discovering the geometric properties of the data space, such as its Euclidean embedding [4][18][20], intrinsic dimensionality [13], connected components [21], homology [16], etc. Particularly, learning representation is closely related to the embedding problem, while clustering can be thought of as finding connected components. Finding an Euclidean embedding of the document space is the primary focus of our work in this paper. Manifold learning techniques can be classified into linear and non-linear techniques. For document processing, we are especially interested in linear techniques due to the consideration of computational complexity. However, our algorithm presented in this paper can be easily extended to nonlinear case. The typical linear techniques for document representation include Latent Semantic Indexing [7], Iterative Residual Rescaling [1][2], Non-negative Matrix Factorization [22], and Locality Preserving Indexing [11].

LSI is originally motivated to deal with the problem of *synonymy* and *polysemy*. The mathematics behind LSI is the Singular Value Decomposition (SVD). The basis functions obtained by SVD are the eigenvectors of the matrix XX^T, where X is the term-document matrix. It would be important to note that LSI is different from Principal Component Analysis (PCA) in that XX^T is generally not the data covariance matrix. In fact, this occurs only when the documents has a zero mean. One of the main advantages of LSI is that its basis functions are orthogonal. Therefore, the metric structure in the LSI subspace can be well preserved. LSI received a lot of attentions during these years and many variants of LSI have been proposed [12][14][19].

LSI is optimal in the sense of preserving the global geometric structure of the document space (inner product). However, it might not be optimal in the sense of discrimination. Specifically, LSI might not be optimal in separating documents with different topics. Recently, LPI is proposed to discover the discriminant structure of the document space. It has shown that it can have more discrimina-

tive power than LSI. A reasonable assumption behind LPI is that, close inputs should have similar topics. The detailed discriminant analysis of LPI can be found in [11]. Different from LSI, LPI is non-orthogonal. Therefore, it can not preserve the metric structure of the document space and suffers from the problem of dimensionality estimation. In fact, inaccurate estimation of the intrinsic dimensionality of the document space would drastically degrade LPI's performance. In the worst case, it can even produce worse performance than in the original representation space.

In this paper, we propose a new algorithm called **Orthogonal Locality Preserving Indexing**. Orthogonal LPI is fundamentally based on LPI. It shares the same locality preserving character as LPI, but at the same time it requires the basis functions to be orthogonal. Orthogonal basis functions preserve the metric structure of the document space. In fact, if we use all the dimensions obtained by Orthogonal LPI, the projective map is simply a rotation map which does not distort the metric structure. Therefore, the performance of Orthogonal LPI is not sensitive to the number of dimensions. While for LPI, since it does not preserve the metric structure, its performance can be much worse than that in the original document space if the dimensionality is inaccurately estimated. Moreover, our empirical study shows that Orthogonal LPI can have more locality preserving power than LPI. Since it has been shown that the locality preserving power is directly related to the discriminating power [11], the Orthogonal LPI is expected to have more discriminating power than LPI.

The rest of the paper is organized as follows: in Section 2, we give a brief review of LSI and LPI. Section 3 introduces our algorithm. We provide a theoretical justification of our algorithm in Section 4. Extensive experimental results on document similarity, local structure of document space and clustering are presented in Section 5. Finally, we provide some concluding remarks and suggestions for future work in Section 6.

2. A BRIEF REVIEW OF LSI AND LPI

LSI is one of the most popular algorithms for document indexing. It is fundamentally based on SVD (Singular Value Decomposition). Given a set of documents $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^m$, they can be represented as a term-document matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$. Suppose the rank of X is r, LSI decompose the X using SVD as follow:

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

Where $\Sigma = diag(\sigma_1, \cdots, \sigma_r)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ are the singular values of X, $U = [\mathbf{a}_1, \cdots, \mathbf{a}_r]$ and $\mathbf{a}_i$ is called left singular vectors, $V = [\mathbf{v}_1, \cdots, \mathbf{v}_r]$ 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. The basic idea of LSI is to extract the most representative features and at the same time the reconstruction error can be minimized. 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:

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

with the constraint

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

Since XX^T is symmetric, the basis functions of LSI are orthogonal. 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 $\mathbf{e} = (1, \cdots, 1)$. In such a case, LSI is identical to Principal Component Analysis [9]. More details on theoretical interpretations of LSI using SVD can refer to [3][8][17].

Different from LSI which aims to extract the most representative features, LPI aims to extract the most discriminative features. Given a similarity matrix S, LPI can be obtained by solving the following minimization problem:

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

with the constraint

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

where $L = D - S$ is the *graph Laplacian* [6] and $D_{ii} = \sum_j S_{ij}$. D_{ii} measures the local density around $\mathbf{x}_i$. LPI constructs the similarity matrix S as:

$$S_{ij} = \begin{cases} \mathbf{x}_i^T\mathbf{x}_j, & \text{if } \mathbf{x}_i \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_j \\ & \text{or } \mathbf{x}_j \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_i \\ 0, & \text{otherwise.} \end{cases}$$

Thus, the objective function in LPI 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 (= $\mathbf{a}^T\mathbf{x}_i$) and y_j (= $\mathbf{a}^T\mathbf{x}_j$) are close as well [11]. Finally, the basis functions of LPI are the eigenvectors associated with the smallest eigenvalues of the following generalized eigen-problem:

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

XDX^T is non-singular after some pre-processing steps on X in LPI, thus, the basis functions of LPI can also be regarded as the eigenvectors of the matrix $(XDX^T)^{-1}XLX^T$ associated with the smallest eigenvalues. Since $(XDX^T)^{-1}XLX^T$ is not symmetric in general, the basis functions of LPI are non-orthogonal.

Once the eigenvectors are computed, let $A_k = [\mathbf{a}_1, \cdots, \mathbf{a}_k]$ be the transformation matrix. Thus, the Euclidean distance between two data points in the reduced space can be computed as follows:

$$\begin{aligned} dist(\mathbf{y}_i, \mathbf{y}_j) &= \|\mathbf{y}_i - \mathbf{y}_j\| \\ &= \|A^T\mathbf{x}_i - A^T\mathbf{x}_j\| \\ &= \|A^T(\mathbf{x}_i - \mathbf{x}_j)\| \\ &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T AA^T(\mathbf{x}_i - \mathbf{x}_j)} \end{aligned}$$

If A is an orthogonal matrix, $AA^T = I$ and the metric structure is preserved.

3. THE ALGORITHM

In this Section, we introduce a novel algorithm for document indexing and representation, called Orthogonal LPI. The theoretical justifications of our algorithm will be presented in Section 4.

In the document analysis and processing problems one is often confronted with the fact that the dimension of the document vector (m) is much larger than the number of documents (n). Thus, the $m \times m$ matrix XDX^T is singular. To overcome this problem, we can apply PCA to project the documents into a subspace without losing any information and the matrix XDX^T becomes non-singular.

The algorithmic procedure of OLPI is stated below:

1. **PCA Projection**: We project the document set x_i into the PCA subspace by throwing away the components corresponding to zero eigenvalue. We denote the transformation matrix

of PCA by W_{PCA}. By PCA projection, the extracted features are statistically uncorrelated and the rank of the new data matrix is equal to the number of features (dimensions).

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 p nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among p nearest neighbors of $\mathbf{x}_i$. 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
$$S_{ij} = \mathbf{x}_i^T \mathbf{x}_j$$
Otherwise, put S_{ij} = 0. The weight matrix S of graph G models the local structure of the document space.

4. **Computing the Orthogonal Locality Preserving Projections**: We define D as a diagonal matrix whose entries are column (or row, since S is symmetric) sums of S, $D_{ii} = \sum_j S_{ji}$. We also define $L = D - S$, which is called Laplacian matrix in spectral graph theory [6]. Let $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k\}$ be the orthogonal locality preserving projections, we define:
$$A^{(k-1)} = [\mathbf{a}_1, \cdots, \mathbf{a}_{k-1}]$$
$$B^{(k-1)} = \left[A^{(k-1)}\right]^T (XDX^T)^{-1} A^{(k-1)}$$
The orthogonal locality preserving vectors $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k\}$ can be iteratively computed as follow:
   - Compute $\mathbf{a}_1$ as the eigenvector of $(XDX^T)^{-1}XLX^T$ associated with the smallest eigenvalue.
   - Compute $\mathbf{a}_k$ as the eigenvector of
$$M^{(k)} = \left\{ I - (XDX^T)^{-1} A^{(k-1)} \left[B^{(k-1)}\right]^{-1} \left[A^{(k-1)}\right]^T \right\} \cdot (XDX^T)^{-1} XLX^T$$
   associated with the smallest eigenvalue of $M^{(k)}$.

5. **OLPI Embedding**: Let $W_{OLPI} = [\mathbf{a}_1, \cdots, \mathbf{a}_l]$, the embedding is as follows:
$$\mathbf{x} \rightarrow \mathbf{y} = W^T \mathbf{x}$$
$$W = W_{PCA} W_{OLPI}$$
where $\mathbf{y}$ is a l-dimensional representation of the document $\mathbf{x}$. W is the transformation matrix.

4. JUSTIFICATIONS

In this section, we provide theoretical justifications of our proposed algorithm.

4.1 Optimal Orthogonal Embedding

We begin with the following definition:

DEFINITION 1. *Let $\boldsymbol{a} \in \mathbb{R}^m$ be a projective map. The* **Locality Preserving Function** *f is defined as follows:*
$$f(\boldsymbol{a}) = \frac{\boldsymbol{a}^T XLX^T \boldsymbol{a}}{\boldsymbol{a}^T XDX^T \boldsymbol{a}}$$

The Locality Preserving Function $f(\mathbf{a})$ evaluates the locality preserving power of the projective map $\mathbf{a}$. Directly minimizing this function will lead to the original LPI algorithm. Our OLPI algorithm tries to find a set of orthogonal basis vectors which minimizes the locality preserving function.

Thus the objective function of OLPI is:
$$\mathbf{a}_1 = \arg\min_{\mathbf{a}} \frac{\mathbf{a}^T XLX^T \mathbf{a}}{\mathbf{a}^T XDX^T \mathbf{a}}$$
$$\mathbf{a}_k = \arg\min_{\mathbf{a}} \frac{\mathbf{a}^T XLX^T \mathbf{a}}{\mathbf{a}^T XDX^T \mathbf{a}}$$
with the constraint
$$\mathbf{a}_k^T \mathbf{a}_1 = \mathbf{a}_k^T \mathbf{a}_2 = \cdots = \mathbf{a}_k^T \mathbf{a}_{k-1} = 0$$

Since XDX^T is positive definite after PCA projection, for any $\mathbf{a}$, we can always normalize it such that $\mathbf{a}^T XDX^T \mathbf{a} = 1$, and the ratio of $\mathbf{a}^T XLX^T \mathbf{a}$ and $\mathbf{a}^T XDX^T \mathbf{a}$ keeps unchanged. Thus, the above minimization problem is equivalent to minimizing the value of $\mathbf{a}^T XLX^T \mathbf{a}$ with an additional constraint as follows,
$$\mathbf{a}^T XDX^T \mathbf{a} = 1$$

Note that, the above normalization is only for simplifying the computation. Once we get the optimal solutions, we can re-normalize them to get a othonormal basis vectors.

It is easy to check that $\mathbf{a}_1$ is the eigenvector of the generalized eigen-problem:
$$XLX^T \mathbf{a} = \lambda XDX^T \mathbf{a}$$
associated with the smallest eigenvalue. Since XDX^T is non-singular, $\mathbf{a}_1$ is the eigenvector of the matrix $(XDX^T)^{-1}XLX^T$ associated with the smallest eigenvalue.

In order to get the k-th basis vector, we minimize the following objective function:
$$f(\mathbf{a}_k) = \frac{\mathbf{a}_k^T XLX^T \mathbf{a}_k}{\mathbf{a}_k^T XDX^T \mathbf{a}_k} \tag{1}$$
with the constraints:
$$\mathbf{a}_k^T \mathbf{a}_1 = \mathbf{a}_k^T \mathbf{a}_2 = \cdots = \mathbf{a}_k^T \mathbf{a}_{k-1} = 0, \quad \mathbf{a}_k^T XDX^T \mathbf{a}_k = 1$$

We can use the Lagrange multipliers to transform the above objective function to include all the constraints
$$C^{(k)} = \mathbf{a}_k^T XLX^T \mathbf{a}_k - \lambda(\mathbf{a}_k^T XDX^T \mathbf{a}_k - 1) - \mu_1 \mathbf{a}_k^T \mathbf{a}_1 - \cdots - \mu_{k-1} \mathbf{a}_k^T \mathbf{a}_{k-1}$$

The optimization is performed by setting the partial derivative of $C^{(k)}$ with respect to $\mathbf{a}_k$ to zero:
$$\frac{\partial C^{(k)}}{\partial \mathbf{a}_k} = 0$$
$$\Rightarrow 2XLX^T \mathbf{a}_k - 2\lambda XDX^T \mathbf{a}_k - \mu_1 \mathbf{a}_1 \cdots - \mu_{k-1} \mathbf{a}_{k-1} = 0 \tag{2}$$

Multiplying the left side of (2) by $\mathbf{a}_k^T$, we obtain
$$2\mathbf{a}_k^T XLX^T \mathbf{a}_k - 2\lambda \mathbf{a}_k^T XDX^T \mathbf{a}_k = 0$$
$$\Rightarrow \lambda = \frac{\mathbf{a}_k^T XLX^T \mathbf{a}_k}{\mathbf{a}_k^T XDX^T \mathbf{a}_k} \tag{3}$$

Comparing to (1), λ exactly represents the expression to be minimized.

Multiplying the left side of (2) successively by $\mathbf{a}_1^T(XDX^T)^{-1}$, $\cdots$, $\mathbf{a}_{k-1}^T(XDX^T)^{-1}$, we now obtain a set of $k-1$ equations:

$$\begin{aligned}
&\mu_1\mathbf{a}_1^T(XDX^T)^{-1}\mathbf{a}_1+\cdots+\mu_{k-1}\mathbf{a}_1^T(XDX^T)^{-1}\mathbf{a}_{k-1}\\
=&2\mathbf{a}_1^T(XDX^T)^{-1}XLX^T\mathbf{a}_k\\
&\mu_1\mathbf{a}_2^T(XDX^T)^{-1}\mathbf{a}_1+\cdots+\mu_{k-1}\mathbf{a}_2^T(XDX^T)^{-1}\mathbf{a}_{k-1}\\
=&2\mathbf{a}_2^T(XDX^T)^{-1}XLX^T\mathbf{a}_k\\
&\cdots\cdots\\
&\mu_1\mathbf{a}_{k-1}^T(XDX^T)^{-1}\mathbf{a}_1+\cdots+\mu_{k-1}\mathbf{a}_{k-1}^T(XDX^T)^{-1}\mathbf{a}_{k-1}\\
=&2\mathbf{a}_{k-1}^T(XDX^T)^{-1}XLX^T\mathbf{a}_k
\end{aligned}$$

We define:

$$\mu^{(k-1)}=[\mu_1,\cdots,\mu_{k-1}]^T,\quad A^{(k-1)}=[\mathbf{a}_1,\cdots,\mathbf{a}_{k-1}]$$

$$B^{(k-1)}=\left[B_{ij}^{(k-1)}\right]=\left[A^{(k-1)}\right]^T(XDX^T)^{-1}A^{(k-1)}$$

$$B_{ij}^{(k-1)}=\mathbf{a}_i^T(XDX^T)^{-1}\mathbf{a}_j$$

Using this simplified notation, the previous set of $k-1$ equations can be represented in a single matrix relationship

$$B^{(k-1)}\mu^{(k-1)}=2\left[A^{(k-1)}\right]^T(XDX^T)^{-1}XLX^T\mathbf{a}_k$$

thus

$$\mu^{(k-1)}=2\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T(XDX^T)^{-1}XLX^T\mathbf{a}_k \quad (4)$$

Let us now multiply the left side of (2) by $(XDX^T)^{-1}$

$$\begin{aligned}
&2(XDX^T)^{-1}XLX^T\mathbf{a}_k-2\lambda\mathbf{a}_k-\mu_1(XDX^T)^{-1}\mathbf{a}_1\\
&-\cdots-\mu_{k-1}(XDX^T)^{-1}\mathbf{a}_{k-1}=0
\end{aligned}$$

This can be expressed using matrix notation as

$$\begin{aligned}
&2(XDX^T)^{-1}XLX^T\mathbf{a}_k-2\lambda\mathbf{a}_k\\
&-(XDX^T)^{-1}A^{(k-1)}\mu^{(k-1)}=0
\end{aligned}$$

With equation (4), we obtain

$$\begin{aligned}
&\left\{I-(XDX^T)^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\}\\
&\cdot(XDX^T)^{-1}XLX^T\mathbf{a}_k=\lambda\mathbf{a}_k
\end{aligned}$$

As shown in (3), λ is just the criterion to be minimized, thus $\mathbf{a}_k$ is the eigenvector of

$$\begin{aligned}
M^{(k)}=&\left\{I-(XDX^T)^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\}\\
&\cdot(XDX^T)^{-1}XLX^T
\end{aligned}$$

associated with the smallest eigenvalue of $M^{(k)}$.

Finally, we get the optimal orthogonal basis vectors. The orthogonal basis of OLPI preserves the metric structure of the document space.

Recall in LPI [11], the basis vectors of LPI is the first k eigenvectors associated with the smallest eigenvalues of the eigen-problem:

$$XLX^T\mathbf{b}=\lambda XDX^T\mathbf{b} \quad (5)$$

Thus, the basis vectors satisfy the following constraint:

$$\mathbf{b}_i^TXDX^T\mathbf{b}_j=0\quad(i\neq j)$$

The transformation of LPI is non-orthogonal. Actually, it is XDX^T-orthogonal.

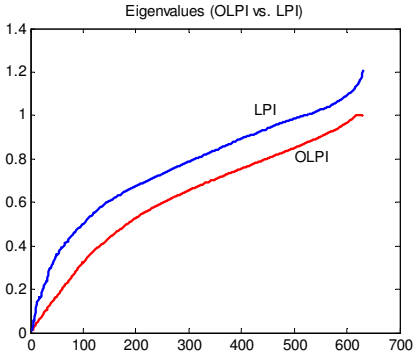


Figure 1: The eigenvalues of LPI and OLPI

4.2 Locality Preserving Power

Both LPI and OLPI try to preserve the local geometric structure. They find the basis vectors by minimizing the Locality Preserving Function:

$$f(\mathbf{a})=\frac{\mathbf{a}^TXLX^T\mathbf{a}}{\mathbf{a}^TXDX^T\mathbf{a}} \quad (6)$$

$f(\mathbf{a})$ reflects the locality preserving power of the projective map $\mathbf{a}$.

In LPI, based on the Rayleigh Quotient format of the eigen-problem (Eqn. 5) [10], the value of $f(\mathbf{a})$ is exactly the eigenvalue of Eqn. (5) corresponding to eigenvector $\mathbf{a}$. Therefore, the eigenvalues of LPI reflect the locality preserving power of LPI. In OLPI, as we show in Eqn. (3), the eigenvalues of OLPI also reflect its locality preserving power. This observation motivates us to compare the eigenvalues of LPI and OLPI.

Fig. 1 shows the eigenvalues of LPI and OLPI. The data set used for this study is the document set "air" in Table 2 (please see Section 5.2.1 for details). As can be seen, the eigenvalues of OLPI is consistently smaller than those of LPI, which indicates that OLPI can have more locality preserving power than LPI. We also did experiments on the other 29 document sets in Table 2 and get the similar results.

Since it has been shown in [11] that the locality preserving power is directly related to the discriminating power, we expect that the OLPI based applications on document processing can obtain better performance than those based on LPI.

5. EXPERIMENTAL RESULTS

In this section, several experiments on TDT2 data corpus were performed to show the effectiveness of our proposed algorithm. We compared our proposed algorithm Orthogonal LPI with LSI and LPI.

5.1 Data Corpus

The TDT2 corpus[1] 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 as described in table 1. Each document is represented as a term-frequency vector. We simply removed the stop

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

Table 1: 20 semantic categories from TDT2 used in our experiments

category	num of doc	category	num of doc
20001	1844	20048	160
20015	1828	20033	145
20002	1222	20039	141
20013	811	20086	140
20070	441	20032	131
20044	407	20047	123
20076	272	20019	123
20071	238	20077	120
20012	226	20018	104
20023	167	20087	98

word and no further preprocessing was done. Each document vector is normalized to 1 and the Euclidean distance is used as the distance measure.

5.2 Similarity Evaluation

5.2.1 Data Preparation

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 $(i = 1, 2, \cdots, 30)$. For each keyword q_i, let D_i denote the set of the documents containing q_i. Let $D = D_1 \cup \cdots \cup D_{30}$. 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 256 to 805 with an average of 507, and the number of topics ranged from 6 to 20 with an average of 16.7 (Table 2). 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 and LPI 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 [11].

Table 2: 30 document subsets

Query	Num of Doc	Query	Num of Doc
air	658	impact	430
british	516	information	641
building	356	legal	565
control	647	material	762
cooperation	348	money	805
court	736	peace	600
decision	764	police	486
domestic	421	robert	351
drug	284	russia	613
fire	320	school	440
food	494	smoking	256
growth	569	technology	349
health	447	trade	622
history	444	violence	349
human	400	women	530

5.2.2 Similarity Measure

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 three different indexing algorithms, i.e. OLPI, LPI and LSI. The similarity measure we used is the cosine similarity.

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

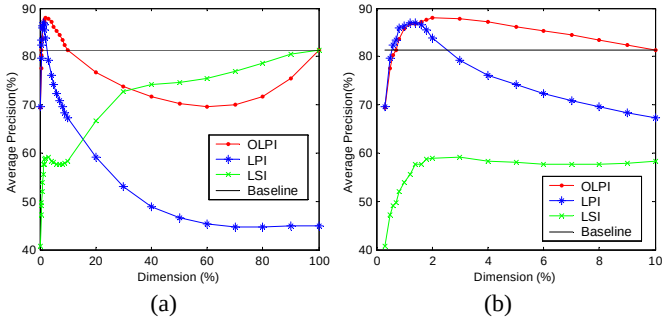


Figure 2: The average precisions of LSI, LPI and OLPI vary with the dimensionality reduction rate. The optimal performances obtained by both LPI and OLPI are better than LSI. Also, OLPI is less sensitive to the dimensionality reduction rate than LPI.

For each document subset D_i (or, $D_{OLPI,i}$, $D_{LPI,i}$, $D_{LSI,i}$), 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, 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:

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

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 we used here is the same as that used in [1][11].

5.2.3 Result

The experimental results on similarity are reported in this subsection. We compared OLPI (corresponding to document set D_{OLPI}) to LPI (corresponding to document set D_{LPI}), 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 OLPI, LPI and LSI varies with the number of dimensions. We compared their results on different dimensions. Figure 2 shows the average precision over 30 document sets (Table 2) with different dimensionality reduction rates. In Figure 2(a), the rate ranges from 0.3% to 100%. In Figure 2(b), the rate ranges from 0.3% to 10%. Figure 2(b) provide us with a better view of the performance changes when the dimensionality is small.

As can be seen from Figure 2, the best performances of both LPI and OLPI are better than baseline. However, the LPI is very sensitive to the dimensionality, which makes the dimensionality estimation extremely crucial in LPI. When the dimensionality is inaccurately estimated, the performance of LPI can be much worse than the baseline. The orthogonal basis functions of OLPI preserve the metric structure of the document space. Moreover, it has more locality preserving power (or, discriminating power) than LPI, as

we have shown in Figure 1. Also, it can be seen that, both LPI and OLPI outperform LSI when the dimensionality is small.

By using LPI or OLPI, we can obtain an extremely low dimensional representation for documents, which might facilitate some real world applications such as clustering, classification and retrieval. And the insensitivity to the dimension makes OLPI more applicable than LPI.

5.3 Discriminating Power

As pointed in [11], LPI is an unsupervised approximation to LDA algorithm which is supervised. Orthogonal LPI share the similar objective function with LPI. It is intrinsically similar with LPI. Thus Orthogonal LPI also has discriminating power. Meanwhile, the orthogonal basis in OLPI make it less sensitive to the reduced dimensionality.

In many cases, the data points (documents) may lack of labels. Specifically, for each data point, we do not know to what specific topic it is related to. However, it might be possible to discover the discriminant structure hidden in the data points. In other words, it might be possible to know if two data points are related to the same topic. In the context of learning theory, it is often assumed that if two points $\mathbf{x}_1$, $\mathbf{x}_2$ are close in the intrinsic geometry of the data space, then they are related to the same topic [5]. In this section, we evaluate the discriminating power of OLPI, LPI and LSI. The dataset we used here is the same as that used in Section 5.2.1.

For each document subset, we project the documents into a subspace by using OLPI, LPI, LSI and baseline algorithm. For the baseline algorithm, we simply use SVD to remove those components corresponding to zero eigenvalue. In other words, the baseline algorithm preserves inner product and there is no information loss, while the dimensionality is reduced. Let n denote the number of data points in the subset and c denote the number of semantic classes contained in this subset. For each semantic class, let p_i denote the number of data points in the i-th class. Let $\mathbf{x}_i^j$ the j-th sample in the i-th semantic class. For each data point $\mathbf{x}_i^j$, we find its p_i nearest neighbors in the subspace. Among these p_i data points, those sharing the same label as $\mathbf{x}_i^j$ are called relevant examples. Thus, we can compute the accuracy for $\mathbf{x}_i^j$ as follows:

$$accuracy(\mathbf{x}_i^j) = \frac{\text{\# of relevant examples}}{p_i}$$

Correspondingly, the average precision can be computed:

$$average\ accuracy = \frac{1}{n}\sum_i \sum_j precision(\mathbf{x}_i^j)$$

As before, the accuracy varies with the dimensionality reduction rates as shown in Figure 3. The rate in Figure 3(a) ranges from 0.3% to 100% and the rate in Figure 3(b) ranges from 0.3% to 10%. As can be seen, the optimal performances obtained by both LPI and OLPI are better than LSI. Also, OLPI is less sensitive to the dimensionality reduction rate than LPI. Therefore, OLPI can work more stably than LPI in the real world applications, such as document clustering.

5.4 Clustering Evaluation

Document clustering is one of most crucial techniques to organize the documents in an unsupervised manner. In this subsection, we investigate the use of indexing algorithms for document clustering.

We chose K-means as our clustering algorithm and compared four methods. These four methods are listed below:

- K-means on original term-document matrix (K-means) - This method is treated as our baseline

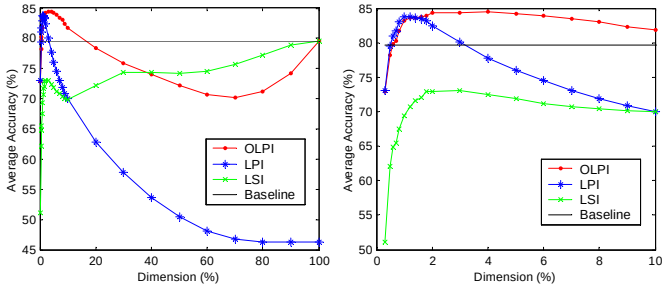


Figure 3: The average accuracies of LSI, LPI and OLPI vary with the dimensionality reduction rate. The optimal performances obtained by both LPI and OLPI are better than LSI. Also, OLPI is less sensitive to the dimensionality reduction rate than LPI.

- K-means after LSI (LSI)
- K-means after LPI (LPI)
- K-means after OLPI (OLPI)

Note that, the two methods LPI and OLPI need to construct a graph on the documents. In this experiment, we used the same graph for these two methods and the parameter p (number of nearest neighbors) was set to 7.

5.4.1 Evaluation Metric

The clustering result is evaluated by comparing the obtained label of each document with that provided by the document corpus. Two metrics, the accuracy (AC) and the normalized mutual information metric ($\overline{MI}$) are used to measure the clustering performance [22]. Given a document $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}$$

where n is the total number of documents and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [15].

Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c_j' \in C'} p(c_i, c_j') \cdot log_2 \frac{p(c_i, c_j')}{p(c_i) \cdot p(c_j')}$$

where $p(c_i)$ and $p(c_j')$ are the probabilities that a document arbitrarily selected from the corpus belongs to the clusters c_i and c_j', respectively, and $p(c_i, c_j')$ is the joint probability that the arbitrarily selected document belongs to the clusters c_i as well as c_j' at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}(C, C')$ ranges from 0 to 1.

$\overline{MI} = 1$ if the two sets of clusters are identical, and $\overline{MI} = 0$ if the two sets are independent.

5.4.2 Clustering Results

The evaluations were conducted with different number of clusters, ranging from 2 to 10. For each given cluster number k, 50 tests were conducted on different randomly chosen categories, and the average performance was computed over these 50 tests. For each test, K-means algorithm was applied 10 times with different start points and the best result in terms of the objective function of K-means was recorded.

Figure 4 shows the average accuracy and average mutual information for different number of classes (different k). Both LPI and OLPI reach their best performance at very low dimensionality. After the optimal dimension, the performance of LPI decreases drastically. For OLPI, its performance fluctuates slightly and is always above the performance of the baseline. For LSI, the clustering performance does not outperform the baseline.

5.5 Discussions

We summarize the experiments below:

1. 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.

2. The discriminating power and orthogonal basis functions are two important factors in acquiring a good document representation method. OLPI combines the advantages of LSI and LPI. Thus, it is expected to be a natural alternative to LPI.

6. CONCLUSIONS AND FUTURE WORK

We have proposed a new algorithm for document indexing and representation, called Orthogonal Locality Preserving Indexing. The new algorithm combines the advantages of both Latent Semantic Indexing and Locality Preserving Indexing. As shown in our experiment results, Orthogonal LPI can have as much discriminative power as the standard LPI, while it does not suffer from the problem of dimensionality estimation.

Several questions remain unclear and will be investigated in our future work:

1. In most of previous work on document indexing, it is assumed that the data space is connected. Correspondingly, the data space has an intrinsic dimensionality. However, this might not be the case for real world data. The data space can be disconnected and different components can have different dimensionality. It remains unclear how often such a case may occur and how to deal with it.

2. Orthogonal LPI is linear, but it can be also performed in reproducing kernel Hilbert space which gives rise to nonlinear maps. It is unclear if the document space is linear or nonlinear. If it is nonlinear, better performance may be obtained by nonlinear techniques. The difficulty is that the document space is always embedded in an extremely high-dimensional ambient space whose bases correspond to the terms. Thus, correctly identifying the geometric structure of the document space require a large amount of sample points.

7. REFERENCES

[1] R. Ando. Latent semantic space: Iterative scaling improves precision of inter-document similarity measurement. In *Proceedings of ACM SIGIR*, 2000.

[2] R. Ando and L. Lee. Iterative residual rescaling: An analysis and generalization. In *Proceedings of ACM SIGIR*, 2001.

[3] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Latent semantic indexing is an optimal special case of multidimensional scaling. In *Proceedings of ACM SIGIR*, 1992.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 2001.

[5] M. Belkin, P. Niyogi, and V. Sindhwani. On maniold regularization. Technical report tr-2004-05, Computer Science Department, The University of Chicago, 2004.

[6] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. 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] C. H. Ding. A similarity-based probability model for latent semantic indexing. In *Proceedings of ACM SIGIR*, 1999.

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

[10] G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[11] X. He, D. Cai, H. Liu, and W.-Y. Ma. Locality preserving indexing for document representation. In *Proceedings of ACM SIGIR*, 2004.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR*, 1999.

[13] B. Kegl. Intrinsic dimension estimation using packing numbers. In *Advances in Neural Information Processing Systems 15*, 2002.

[14] E. Kokiopoulou and Y. Saad. Polynomial filtering in latent semantic indexing for information retrieval. In *Proceedings of ACM SIGIR*, 2004.

[15] L. Lovasz and M. Plummer. *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest, 1986.

[16] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. Technical report tr-2004-08, Department of Computer Science, University of Chicago, 2004.

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

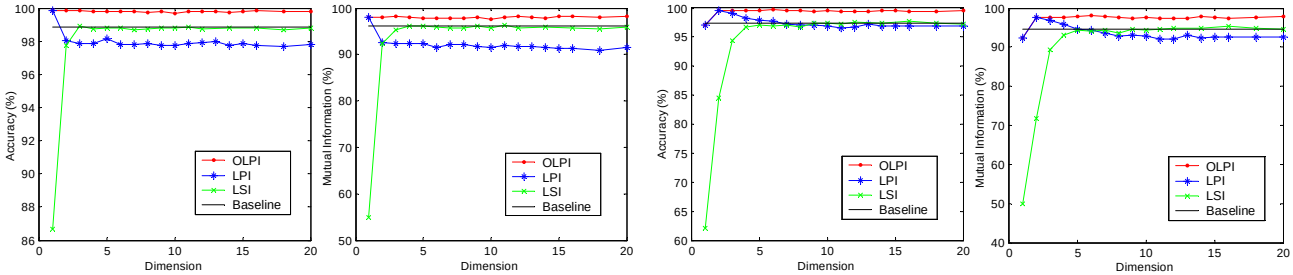
[18] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[19] C. Tang, S. Dwarkadas, and Z. Xu. On scaling latent semantic indexing for large peer-to-peer systems. In *Proceedings of ACM SIGIR*, 2004.

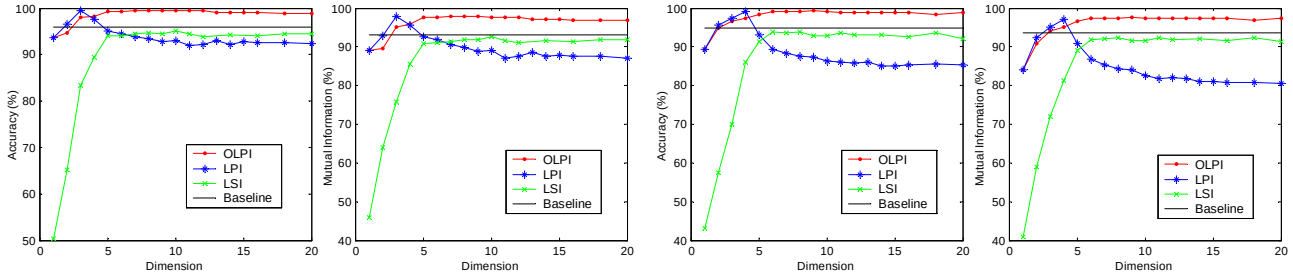
[20] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[21] U. von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in Neural Information Processing Systems 17*, 2004.

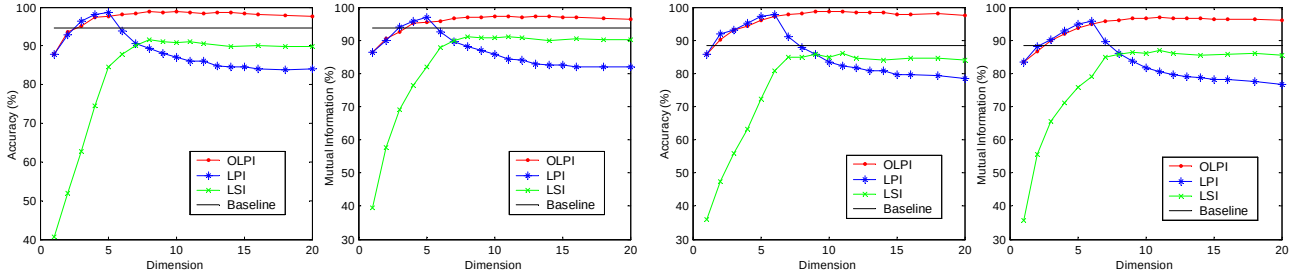
[22] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of ACM SIGIR*, 2003.



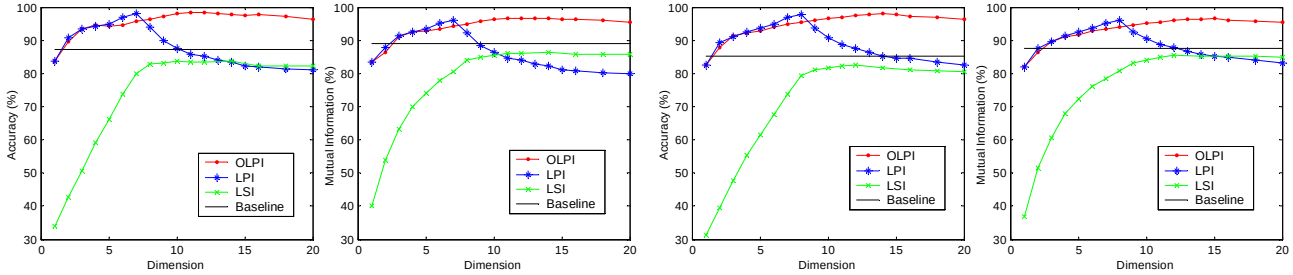
(a) 2 Classes (b) 3 Classes



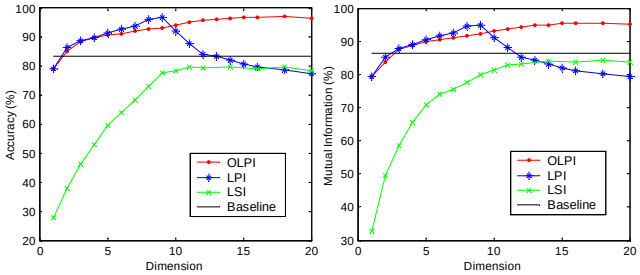
(c) 4 Classes (d) 5 Classes



(e) 6 Classes (f) 7 Classes



(g) 8 Classes (h) 9 Classes



(i) 10 Classes

Figure 4: The average accuracy and average mutual information for different number of classes. The clustering performance was evaluated at different dimensionality (1∼20) for LSI, LPI and OLPI. Both LPI and OLPI reach their best performance at very low dimensionality. After the optimal dimension, the performance of LPI decreases drastically. For OLPI, its performance fluctuates slightly and is always above the performance of the baseline. For LSI, the clustering performance does not outperform the baseline.