



A new differential LSI space-based probabilistic document classifier

Liang Chen^{a,*}, Naoyuki Tokuda^b, Akira Nagai^c

^a *Computer Science Department, University of Northern British Columbia, Prince George, BC, Canada, V2N 4Z9*

^b *Sunflare Company, Shinjuku-Hirose Bldg., 7 Yotsuya 4-chome, Shinjuku-ku, Tokyo, Japan 160-0004*

^c *Advanced Media Network Center, Utsunomiya University, Utsunomiya, Tochigi, Japan 321-8585*

Received 26 October 2002; received in revised form 29 August 2003

Communicated by Wen-Lian Hsu

Abstract

We have developed a new effective probabilistic classifier for document classification by introducing the concept of differential document vectors and DLSI (differential latent semantic indexing) spaces. A combined use of the projections on and the distances to the DLSI spaces introduced from the differential document vectors improves the adaptability of the LSI (latent semantic indexing) method by capturing unique characteristics of documents. Using the intra- and extra-document statistics, both a simple posteriori calculation on a small example and an experiment on a large Reuters-21578 database demonstrate the advantage of the DLSI space-based probabilistic classifier over the LSI space-based classifier in classification performance.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Document classification; Latent semantic indexing; Differential document vector; Information retrieval

1. Introduction

This paper introduces a new efficient supervised document classification procedure whereby, given a training set comprising a large number of labeled documents preclassified into a finite number of appropriate clusters, one can generate a classifier to classify any of new documents into an appropriate cluster.

The vector space model is widely used in document classification, where each document is represented as a vector of terms. To represent a document by a docu-

ment vector, we assign weights to its components usually evaluating the frequency of occurrences of the corresponding terms. Then the standard pattern recognition and machine learning methods are employed for document classification [1,2].

In view of the inherent flexibility imbedded within any natural language, a staggering number of dimensions seem required to represent the featuring space of any practical document comprising the huge number of terms used. If a speedy classification algorithm can be developed [3], the first problem to be resolved is the dimensionality reduction scheme enabling the documents' term projection onto a smaller subspace.

Basically there are two types of approaches for projecting documents or reducing the documents' dimen-

* Corresponding author. The research of this author is supported by an NSERC Discovery Grant.

E-mail address: chenl@unbc.ca (L. Chen).

sions. One is based on local methods, often referred to as truncation, where we delete a number of “unimportant” or “irrelevant” terms from a document vector, the importance of a term being evaluated often by a weighting system based on its frequency of occurrences in the document. The method is called local because each document is projected into a different subspace but its effect is minimal in document vectors because the vectors are sparse [3]. The approaches of the other type are often termed global methods, where the terms to be deleted are chosen first, ensuring that all the document vectors are projected into the same subspace with the same terms being deleted from each document. In this process, a global method always loses some of the important features of adaptability to the unique characteristics of each document. How to improve this adaptability is our main task of the paper.

Like an eigen-decomposition method extensively used in image processing and image recognition [4, 5], the Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD) has proved to be an efficient method for the dimensionality reduction scheme in document analysis and feature extraction, providing a powerful tool for the classifier [3] when introduced into document retrieval with a good performance confirmed by empirical studies [6]. A distinct advantage of LSI-based dimensionality reduction lies in the fact that among all the projections on all the possible spaces having the same number of dimensions, the projection of the set of document vectors on the LSI space has a lowest possible least-square distance to the original document vectors. This implies that the LSI finds an optimal solution to dimensional reduction. In addition to the role of dimensionality reduction, the LSI with SVD is also effective in offering a dampening effect of synonymy and polysemy problems [3] which a simple scheme of deleting terms can not be expected to cope with. Also known as a word sense disambiguation problem [7] the source of synonymy and polysemy problem can be traced to inherent characteristics of context sensitive grammar of any natural language. Having the two advantages, the LSI has been found to provide a most popular dimensional reduction tool.

As pointed out by Schütze and Silverstein [3], the LSI is indeed a global dimensional reduction approach. Like all global projection schemes, LSI also encounters a difficulty in adapting to the unique

characteristics of each document; a method must be developed to improve an adverse performance of a document classifier due to this inability.

In this paper, we will show that exploiting both of the distances to, and the projections on, a reduced document space improves the performance as well as the robustness of the document classifier (a similar approach is taken by Moghaddam, Wahid, and Pentland for image recognition [8]). To do this, we introduce, as the major vector spaces, two reduced differential document spaces called differential LSI (or DLSI) spaces that are subspaces of document spaces formed respectively from the differences between normalized term-document vectors belonging to a same cluster, and from the differences between normalized term-document vectors belonging to different clusters. To evaluate the possibility of a document belonging to a cluster, the new classifier sets up a Bayesian posteriori probability function for the differential document vectors based on their projections on the DLSI spaces and their distances to the DLSI spaces, selecting the candidate having a highest probability.

As it is always done in LSI-based models for information retrieval and/or document classification, we assume that the size of the training set is always quite large.

2. Main algorithm

2.1. Overall description

Given a document, the document vector and its normalized form can be directly set up exploiting the terms appearing in the document and their frequency of occurrences in the document, and possibly also in other documents. The centroid of a cluster is given by an average of the sums of the normalized vectors of its members. The cosine of a pair of normalized document vectors measures the angle of the pair of normalized document vectors.

To obtain an intra-DLSI space, or an I-DLSI space, we first set up a differential term by intra-document matrix where each column of the matrix denotes the difference between two documents belonging to a same cluster. Now exploiting the singular vector decomposition method, the major left singular vectors associated with the largest singular values are selected

as a major vector space called an intra-DLSI space, or an I-DLSI space. The I-DLSI space is effective in roughly describing the differential intra-document vectors, while the distance from a differential intra-document vector to the I-DLSI space can be effectively used as additive information to improve adaptability to the unique characteristics of the particular differential document vector. Given a new document to be classified, a best cluster to be recalled can be selected from among those candidate clusters to which the given document is most likely belong. Consequently, the candidate clusters should be chosen in such a way that the differences from their centroids to the given document vector have high probabilities of being differential intra-document vectors. The probability function for a differential document vector being a differential intra-document vector is calculated according to the projection on and the distance to the I-DLSI space from the differential document vector.

The extra-DLSI space, or the E-DLSI space can similarly be obtained by setting up a differential term by extra-document matrix where each column of the matrix denotes now a differential document vector between two document vectors belonging to different clusters. The extra-DLSI space can now be constructed by the major left singular vectors associated with the largest singular values. As in the intra-DLSI space, in addition to the global description capability, the space shares the improved adaptability to the unique characteristics of the particular differential document vector. Given a new document to be classified, a best candidate cluster to be recalled can be selected from among those clusters to which the given document is most unlikely not to belong. Consequently, the candidate clusters should be chosen in such a way that the differences from their centroids to the given document vector have low probabilities of being differential extra-document vectors. The probability function for a differential document vector being a differential extra-document vector is calculated according to projection on and distance to the E-DLSI space from the differential document vector.

Now integrating the concepts of the differential intra- and extra-document vectors, we set up a Bayesian posteriori likelihood function providing a most probable similarity measure of a document belonging to a cluster. As we stated already in Introduction, the projections of differential document vectors onto I-

DLSI, and E-DLSI spaces and the distances from the vectors to these spaces in our scheme have one advantage over the conventional LSI space-based approach: in addition to the role of the length of the projection of a differential document vector which is equivalent to that of the cosine of the angle of the projections of two document vectors in the LSI space-based approach, the distance of the differential document vector to the projected DLSI space allows the evaluation of the similarity measure of each individual document which the global method generally fails. We believe that by combining both the projections on as well as the distances to the DLSI spaces from differential vectors, our new scheme provides much richer information, capturing more unique characteristics of particular documents.

2.2. Basic concepts

A term is defined as a word or a phrase that appears at least in two documents. We exclude the so-called stop words such as “a”, “the”, “of” and so forth. Suppose we select and list the terms that appear in the documents as t_1, t_2, \dots, t_m .

For each document j in the collection, we assign each of the terms with a real vector $(a_{1j}, a_{2j}, \dots, a_{mj})^T$, with $a_{ij} = f_{ij} \times g_i$, where f_{ij} is the local weight of the term t_i in the document indicating the significance of the term in the document, while g_i is a global weight of all the documents, which is a parameter indicating the importance of the term in representing the documents. Local weights could be either raw occurrence counts, boolean, or logarithm of occurrence count. Global ones could be no weighting (uniform), domain specific, or entropy weighting. Both of the local and global weights are thoroughly studied in the literatures (e.g., [9]), and will not be discussed further in this paper. An example is given below:

$$f_{ij} = \log(1 + O_{ij}) \quad \text{and}$$

$$g_i = 1 - \frac{1}{\log n} \sum_{j=1}^N p_{ij} \log(p_{ij}),$$

where $p_{ij} = O_{ij}/d_i$, d_i is the total number of times that term t_i appears in the collection, O_{ij} the number of times the term t_i appears in the document j , and n the number of documents in the collection. The docu-

ment vector $(a_{1j}, a_{2j}, \dots, a_{mj})$ can be normalized as $(b_{1j}, b_{2j}, \dots, b_{mj})$ by the following formula:

$$b_{ij} = a_{ij} / \sqrt{\sum_{l=1}^m a_{lj}^2}. \quad (1)$$

The normalized centroid vector $(c_1, c_2, \dots, c_m)^T$ of a cluster can similarly be calculated in terms of the normalized vector as

$$c_i = s_i / \sqrt{\sum_{j=1}^m s_j^2}, \quad (2)$$

where $(s_1, s_2, \dots, s_m)^T$ is a mean vector of the member documents in the cluster which are normalized as T_1, T_2, \dots, T_k . $(s_1, s_2, \dots, s_m)^T$ can be expressed as

$$(s_1, s_2, \dots, s_m)^T = \frac{1}{k} \sum_{j=1}^k T_j.$$

The normalized centroid vector of a cluster is regarded as a normalized document vector of the cluster.

A differential document vector is defined as $T_i - T_j$ where T_i and T_j are normalized document vectors satisfying some criteria as given above.

A differential intra-document vector D_I is the differential document vector defined as $T_i - T_j$, where T_i and T_j are two normalized document vectors belonging to a same cluster. A differential extra-document vector D_E is the differential document vector defined as $T_i - T_j$, where T_i and T_j are two normalized document vectors belonging to two different clusters.

It is important to notice that, in the above intra- and extra document vectors, we can select the centroid vector of a cluster as the T_i or T_j . This is because a centroid vector is regarded as one of a normalized document vector of a cluster.

The differential term by intra- and extra-document matrices D_I and D_E are, respectively, defined as a matrix, each column of which comprises a differential intra- and extra-document vector, respectively.

2.3. The posteriori model

Any differential term by document m -by- n matrix of D , say, of rank $r \leq q = \min(m, n)$, whether it is a differential term by intra-document matrix D_I or a differential term by extra-document matrix D_E ,

can be decomposed by SVD into a product of three matrices: $D = USV^T$, such that U (left singular matrix) and V (right singular matrix) are an m -by- q and q -by- n unitary matrices with the first r columns of U and V being the eigenvectors of DD^T and D^TD , respectively. Here S is called a singular matrix expressed by $S = \text{diag}(\delta_1, \delta_2, \dots, \delta_q)$, where δ_i are nonnegative square roots of the eigen values of DD^T , $\delta_i > 0$ for $i \leq r$ and $\delta_i = 0$ for $i > r$.

The diagonal elements of S are sorted in the decreasing order of magnitude. To obtain a new reduced matrix S_k , we simply keep the k -by- k leftmost-upper corner matrix ($k < r$) of S and delete other terms; we similarly obtain the two new matrices U_k and V_k by keeping the left most k columns of U and V , respectively. The product of U_k, S_k and V_k^T provide a reduced matrix D_k of D which is approximately equal to D .

How we choose an appropriate value of k , a reduced degree of dimension from the original matrix, depends on the type of applications. Generally we choose $k \geq 100$ for $1000 \leq n \leq 3000$, and the corresponding k is normally smaller for the differential term by intra-document matrix than that for the differential term by extra-document matrix, because the differential term by extra-document matrix normally has more columns than the differential term by intra-document matrix has.

Each of differential document vector q could find a projection on the k -dimensional fact space spanned by the k columns of U_k . The projection can easily be obtained by $U_k^T q$.

Noting that the mean of the differential intra-(extra-)document vectors are approximately 0, we may assume that the differential vectors formed follow a high-dimensional Gaussian distribution so that the likelihood of any differential vector x will be given by

$$P(x|D) = \frac{\exp[-1/2d(x)]}{(2\pi)^{n/2} |\Sigma|^{1/2}},$$

where $d(x) = x^T \Sigma^{-1} x$, and Σ is the covariance of the distribution computed from the training set expressed $\Sigma = \frac{1}{n} DD^T$.

Since δ_i^2 constitutes the eigen values of DD^T , we have $S^2 = U^T DD^T U$, and thus we have $d(x) = nx^T (DD^T)^{-1} x = nx^T U S^{-2} U^T x = ny^T S^{-2} y$, where $y = U^T x = (y_1, y_2, \dots, y_n)^T$.

Because S is a diagonal matrix, $d(x)$ can be expressed by a simpler form as: $d(x) = n \sum_{i=1}^r y_i^2 / \delta_i^2$. This allows us to estimate it more conveniently as

$$\hat{d}(x) = n \left(\sum_{i=1}^k y_i^2 / \delta_i^2 + \frac{1}{\rho} \sum_{i=k+1}^r y_i^2 \right),$$

where

$$\rho = \frac{1}{r-k} \sum_{i=k+1}^r \delta_i^2.$$

In practice, δ_i ($i > k$) could be estimated by fitting a function (say, $1/i$) to the available δ_i ($i \leq k$), or we could let $\rho = \delta_{k+1}^2/2$ since we only need to compare the relative probability. Because the columns of U are orthogonal vectors, $\sum_{i=k+1}^r y_i^2$ could be estimated by $\|x\|^2 - \sum_{i=1}^k y_i^2$. Thus, the likelihood function $P(x|D)$ could now be estimated by

$$\hat{P}(x|D) = \frac{n^{1/2} \exp(-\frac{n}{2} \sum_{i=1}^k \frac{y_i^2}{\delta_i^2}) \cdot \exp(-\frac{n\varepsilon^2(x)}{2\rho})}{(2\pi)^{n/2} \prod_{i=1}^k \delta_i \cdot \rho^{(r-k)/2}}, \quad (3)$$

where

$$y = U_k^T x, \quad \varepsilon^2(x) = \|x\|^2 - \sum_{i=1}^k y_i^2,$$

$$\rho = \frac{1}{r-k} \sum_{i=k+1}^r \delta_i^2,$$

and r is the rank of matrix D . In practice, ρ may be chosen as $\delta_{k+1}^2/2$, and n may be substituted for r . Note that in Eq. (3), the term $\sum(y_i^2/\delta_i^2)$ describes the projection of x onto the DLSI space, while $\varepsilon(x)$ approximates the distance from x to DLSI space.

When both $P(x|D_I)$ and $P(x|D_E)$ are computed, the Bayesian posteriori function can be computed as:

$$P(D_I|x) = \frac{P(x|D_I)P(D_I)}{P(x|D_I)P(D_I) + P(x|D_E)P(D_E)},$$

where $P(D_I)$ is set to $1/n_c$ where n_c is the number of clusters in the database,¹ while $P(D_E)$ is set to $1 - P(D_I)$.

¹ $P(D_I)$ can also be set to be an average number of recalls divided by the number of clusters in the data base if we do not require that the clusters are non-overlapped.

2.4. Algorithm

2.4.1. Setting up the DLSI space-based classifier

- (1) By preprocessing the documents, identify terms either of the words and noun phrases, excluding stop words.
- (2) Construct the system terms by setting up the term list as well as the global weights.
- (3) Normalize the document vectors of all the collected documents, as well as the centroid vectors of each cluster.
- (4) Construct the differential term by intra-document matrix $D_I^{m \times n_I}$, such that each of its columns is an differential intra-document vector.²
- (5) Decompose D_I , by an SVD algorithm, into $D_I = U_I S_I V_I^T$, $S_I = \text{diag}(\delta_{I,1}, \delta_{I,2}, \dots)$, followed by the composition of $D_{I,k_I} = U_{k_I} S_{k_I} V_{k_I}^T$ giving an approximate D_I in terms of an appropriate k_I , then evaluate the likelihood function:

$$P(x|D_I) = \frac{n_I^{1/2} \exp(-\frac{n_I}{2} \sum_{i=1}^{k_I} \frac{y_i^2}{\delta_{I,i}^2}) \exp(-\frac{n_I \varepsilon^2(x)}{2\rho_I})}{(2\pi)^{n_I/2} \prod_{i=1}^{k_I} \delta_{I,i} \cdot \rho_I^{(r_I-k_I)/2}}, \quad (4)$$

where

$$y = U_{k_I}^T x, \quad \varepsilon^2(x) = \|x\|^2 - \sum_{i=1}^{k_I} y_i^2,$$

$$\rho_I = \frac{1}{r_I - k_I} \sum_{i=k_I+1}^{r_I} \delta_{I,i}^2,$$

and r_I is the rank of matrix D_I . In practice, r_I may be set to n_I , and ρ_I to $\delta_{I,k_I+1}^2/2$ if both n_I and m are sufficiently large.

- (6) Construct the term by extra-document matrix $D_E^{m \times n_E}$, such that each of its columns is an extra-differential document vector.
- (7) Decompose D_E , by exploiting the SVD algorithm, into $D_E = U_E S_E V_E^T$, $S_E = \text{diag}(\delta_{E,1}, \delta_{E,2}, \dots)$, then with a proper k_E , define the $D_{E,k_E} = U_{k_E} S_{k_E} V_{k_E}^T$ to approximate D_E . We now define the likelihood function as:

² For a cluster with s elements, we may include at most $m - 1$ differential intra-document vectors in D_I if the linear dependency among columns is to be avoided.

$$P(x|D_E) = \frac{n_E^{1/2} \exp\left(-\frac{n_E}{2} \sum_{i=1}^{k_E} \frac{y_i^2}{\delta_{E,i}^2}\right) \exp\left(-\frac{n_E \varepsilon^2(x)}{2\rho_E}\right)}{(2\pi)^{n_E/2} \prod_{i=1}^{k_E} \delta_{E,i} \cdot \rho_E^{(r_E - k_E)/2}}, \quad (5)$$

where

$$y = U_{k_E}^T x, \quad \varepsilon^2(x) = \|x\|^2 - \sum_{i=1}^{k_E} y_i^2,$$

$$\rho_E = \frac{1}{r_E - k_E} \sum_{i=k_E+1}^{r_E} \delta_{E,i}^2,$$

r_E is the rank of matrix D_E . In practice, r_E may be set to n_E , and ρ_E to $\delta_{E,k_E+1}^2/2$ if both n_E and m are sufficiently large.

(8) Define the posteriori function:

$$P(D_I|x) = \frac{P(x|D_I)P(D_I)}{P(x|D_I)P(D_I) + P(x|D_E)P(D_E)}, \quad (6)$$

$P(D_I)$ is set to $1/n_c$ where n_c is the number of clusters in the database and $P(D_E)$ is set to $1 - P(D_I)$.

As explained in Section 2.2, centroid vectors *can* be used in defining the differential document vectors. In practice, however, when an application involves large document sets, we do not use the centroid vectors of clusters in constructing the differential document vectors in the matrices D_I and D_E so that these matrices remain sparse allowing us to take an advantage of those SVD algorithms for large sparse matrices.

2.4.2. Automatic classification by DLSI space-based classifier

- (1) A document vector is set up by generating the terms as well as their frequencies of occurrence in the document, so that a normalized document vector N is obtained for the document by Eq. (1). For each of the clusters of the database, repeat the procedure of items (2)–(4) below.
- (2) Using the document to be classified, construct a differential document vector $x = N - C$, where C is the normalized vector of the center or centroid of the cluster.

- (3) Calculate the intra-document likelihood function $P(x|D_I)$, and calculate the extra-document likelihood function $P(x|D_E)$ for the document.
- (4) Calculate the Bayesian posteriori probability function $P(D_I|x)$.
- (5) Select the cluster having a largest $P(D_I|x)$ as the recall candidate.

3. Experiments

3.1. Simple example

3.1.1. Problem description

We demonstrate our algorithm by means of a numerical example below. Suppose we have the following 8 documents in the database:

- T_1 : Algebra and Geometry Education System.
- T_2 : The Software of Computing Machinery.
- T_3 : Analysis and Elements of Geometry.
- T_4 : Introduction to Modern Algebra and Geometry.
- T_5 : Theoretical Analysis in Physics.
- T_6 : Introduction to Elements of Dynamics.
- T_7 : Modern Alumina.
- T_8 : The Foundation of Chemical Science.

And we know in advance that they belong to 4 clusters, namely, $T_1, T_2 \in C_1$, $T_3, T_4 \in C_2$, $T_5, T_6 \in C_3$ and $T_7, T_8 \in C_4$ where C_1 belongs to Computer related field, C_2 to Mathematics, C_3 to Physics, and C_4 to Chemical science. We will show below how DLSI- and LSI-based classifier perform in classifying the following new document:

N : “The Elements of Computing Science.”

The DLSI-based classifier and the LSI-based classifier will be set up for this example.

First, we can easily set up the document vectors of the database giving the term by document matrix by simply counting the frequency of occurrences; then we could further obtain the normalized form as in Table 1.

The document vector for the new document N is given by:

$$(0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0)^T,$$

and in normalized form by

(0, 0, 0, 0, 0.577350269, 0, 0, 0.577350269, 0, 0, 0, 0, 0.577350269, 0, 0, 0)ᵀ.

3.1.2. DLSI space-based classifier

Firstly, we calculate the normalized form of the centroid C_i ($i = 1, 2, 3, 4$) of each cluster by Eq. (2).

Following the procedure of the previous section, it is easy to construct both the differential term by intra-document matrix and the differential term by extra-document matrix. Let us denote the differential term by intra-document matrix by $D_I = (T_1 - C_1, T_3 - C_2, T_5 - C_3, T_7 - C_4)$ and the differential term by extra-document matrix by $D_E = (T_2 - C_2, T_4 - C_3, T_6 - C_4, T_8 - C_1)$, respectively. Then we can decompose them into $D_I = U_I S_I V_I^T$ and $D_E = U_E S_E V_E^T$ by using SVD algorithm.

We now choose the number k in such a way that $\delta_k - \delta_{k+1}$ remains sufficiently large. For this problem, we find that we would have to choose $k_I = k_E = 1$ or $k_I = k_E = 3$. Now using Eqs. (4), (5) and (6), we can calculate the $P(x|D_I)$, $P(x|D_E)$ and finally $P(D_I|x)$ for each differential document vector $x = N - C_i$ ($i = 1, 2, 3, 4$) as shown in Table 2. The C_i having a largest $P(D_I|N - C_i)$ is chosen as the cluster to which

the new document N belongs. Because both n_I, n_E are actually quite small, we may here set

$$\rho_I = \frac{1}{r_I - k_I} \sum_{i=k_I+1}^{r_I} \delta_{I,i}^2,$$

$$\rho_E = \frac{1}{r_E - k_E} \sum_{i=k_E+1}^{r_E} \delta_{E,i}^2.$$

The last row of Table 2 clearly shows that cluster C_2 , that is, “Mathematics” is the best possibility regardless of the parameters $k_I = k_E = 1$ or $k_I = k_E = 3$ chosen, showing the robustness of the computation.

3.1.3. LSI space-based classifier

As we have already explained in Introduction, the LSI-based classifier works as follows: First, employ an SVD algorithm on the term by document matrix to set up an LSI space, then the classification is completed within the LSI space.

For the current simple example, the LSI-based classifier could be set up as follows:

- (1) Use the SVD to decompose the normalized term by document matrix as shown in Table 1.
- (2) Select the LSI space.

Table 1
The normalized document vectors

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
Algebra	0.5	0	0	0.5	0	0	0	0
Alumina	0	0	0	0	0	0	0.707106781	0
Analysis	0	0	0.577350269	0	0.577350269	0	0	0
Chemical	0	0	0	0	0	0	0	0.577350269
Compute	0	0.577350269	0	0	0	0	0	0
Dynamics	0	0	0	0	0	0.577350269	0	0
Education	0.5	0	0	0	0	0	0	0
Element	0	0	0.577350269	0	0	0.577350269	0	0
Foundation	0	0	0	0	0	0	0	0.577350269
Geometry	0.5	0	0.577350269	0.5	0	0	0	0
Introduction	0	0	0	0.5	0	0.577350269	0	0
Machine	0	0.577350269	0	0	0	0	0	0
Modern	0	0	0	0.5	0	0	0.707106781	0
Physics	0	0	0	0	0.577350269	0	0	0
Science	0	0	0	0	0	0	0	0.577350269
Software	0	0.577350269	0	0	0	0	0	0
System	0.5	0	0	0	0	0	0	0
Theory	0	0	0	0	0.577350269	0	0	0

Table 2
Classification with DLSI space-based classifier

x	$k_I = k_E = 1$				$k_I = k_E = 3$			
	$N - C_1$	$N - C_2$	$N - C_3$	$N - C_4$	$N - C_1$	$N - C_2$	$N - C_3$	$N - C_4$
$U_{k_I}^T x$	-0.085338834	-0.565752063	-0.368120678	-0.077139955	-0.085338834	-0.556196907	-0.368120678	-0.077139955
					-0.404741071	-0.403958563	-0.213933843	-0.250613624
					-0.164331163	0.249931018	0.076118938	0.35416984
$P(x D_I)$	0.000413135	0.000430473	0.00046034	0.000412671	3.79629E-5	7.03221E-5	3.83428E-5	3.75847E-5
$U_{k_I}^T x$	-0.281162007	0.022628465	-0.326936108	0.807673935	-0.281162007	-0.01964297	-0.326936108	0.807673935
					-0.276920807	0.6527666	0.475906836	-0.048681069
					-0.753558043	-0.619983845	0.258017361	-0.154837357
$P(x D_E)$	0.002310807	0.002065451	0.002345484	0.003140447	0.003283825	0.001838634	0.001627501	0.002118787
$P(D_I x)$	0.056242843	0.064959115	0.061404975	0.041963635	0.003838728	0.012588493	0.007791905	0.005878172

Table 3
Cosine of the projections of the centers and the document in LSI space

	$\text{Cosine}(C_1, U_k^T N)$	$\text{Cosine}(C_2, U_k^T N)$	$\text{Cosine}(C_3, U_k^T N)$	$\text{Cosine}(C_4, U_k^T N)$
$k = 6$	0.359061035	0.359993858	0.527874697	0.320188454
$k = 2$	0.15545458	0.78629345	0.997897255	0.873890479

- (3) Locate the approximate documents for all the documents T_1, T_2, \dots, T_8 in the LSI space.
- (4) Find the centroid of each cluster.
- (5) Calculate the similarity of the document vector for the new document N to be classified and each of the centroid vectors based on the Cosine formula, find the cluster having a largest similarity with the document vector to be classified.

The normalized term by document matrix D of Table 1 is decomposed into USV^T by an SVD algorithm.

As in the DLSI computation, the number k is chosen such that $\delta_k - \delta_{k+1}$ is sufficiently large. For this problem, we have to choose $k = 6$ or $k = 2$ to set up the classifier, for testing the result. Noting that the similarity is calculated as the angle between vectors in the LSI space, the dimension of the LSI space should at least be 2 so that k should be larger than 1. Once k is chosen, the term by document matrix D could be approximated by $D_k = U_k S_k V_k^T$. Thus, the projection of the document T_i onto the LSI space is calculated to be $S_k V_k^T e_i$ (see [6], where e_i is defined as the i th column of an 8×8 identity matrix). Within the LSI space, we should calculate the centroid vector of each cluster. Since the similarities are calculated by the

cosines of angles in the LSI space, the centroid vector of each cluster should be calculated as a mean of the normalized member vectors in the LSI space. The similarity between a centroid vector and the document in the LSI space is easily calculated by the cosine of the angle between in the space, expressed as

$$\text{Cosine}(C_i, N) = \frac{C_i U_k^T N}{\|C_i\|_2 \|U_k^T N\|_2}.$$

The results of the similarities are shown in Table 3. The result of the table implies that for both cases of $k = 2$ and $k = 6$, the most likely cluster to which the document N belongs to is C_3 , namely “Physics”.

3.1.4. Conclusion of the simple example

For this simple example, the DLSI space-based approach finds the most reasonable cluster for the document “The elements of computing science” from the classifiers using either 1 or 3 dimensions for the DLSI-I and DLSI-E spaces.

But the LSI approach fails to predict this for both of the classifiers we have computed using 2- and 4-dimensional LSI spaces. It is worth to note that for the particular example, 1 or 3 dimensions for DLSI space, and 2 or 4 dimensions for the LSI space seem to be the most reasonable dimensions to choose.

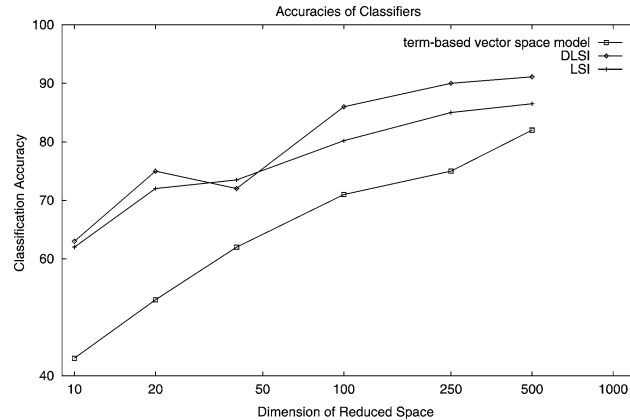


Fig. 1. Classification accuracies on Reuters-21578 test data.

We should note that a conventional matching method of “common” words does not work in this example, because the words “compute” and, “science” in the new document appear in C_1 and C_4 separately, while the word “elements” occur in both C_2 and C_3 simultaneously, giving no indication on the appropriate candidate of classification by simply counting the “common” words among documents. By calculating the cosine of the document N and C_i in the original document space, we have $\cos(C_1, N) = \cos(C_3, N) = \cos(C_4, N) = 0.23570226$, and $\cos(C_2, N) = 0.207630918$. This means that N is more likely to be in the classes of C_1 , C_3 and C_4 than in class C_2 . Thus, we see that the simple term-based vector space model also does not work.

3.2. Experiment on a large database

To test the performances of DLSI-based classifier, LSI-based classifier, and simple term-based vector space model on a large database, we have used a large Reuters-21578 database.³ To split the 21578 articles into training and testing sets, we have used “Mod-Lewis” split [10].⁴ We further discard the documents with multiple labels of categories, and the categories

³ <http://www.daviddlewis.com/resources/testcollections/reuters21578/>.

⁴ It firstly excludes those that were not classified by the indexers, then places the articles which appeared on April 7, 1987 or before in the training set, and others in the test set.

that do not occur at least once in both the training and testing sets.

Now the LSI-based and the new DLSI-based classification methods are applied on the above data sets. We follow the usual experimental setup on this collection [11,12] to split the classification task into binary decision. A binary classifier is trained for each of the categories to determine whether a document belongs to it or not. For each classifier, we use the articles in a category as the positive examples and the rest as the opposite. As the positive clusters are always much larger than the opposite ones, subsets of the opposite clusters are selected randomly in the training sets so that there is an equal number of articles in a positive cluster and its opposite.

Notice that the number k 's in SVD decompositions of LSI & DLSI algorithms represent the dimensions of the reduced DLSI and LSI spaces, Fig. 1 shows the accuracy of each classifier achieves in relation to the dimension for the reduced LSI/DLSI space. Naturally, the classifiers are generated using only the training set, and the error rates are evaluated using only testing set. By randomly choosing the terms to be used for representing document vectors, where the number of the terms is now the dimension of the document space, we are able to use simple term-based vector space model as a classifier to classify the documents. The accuracy of the simple term-based vector space model is also given in Fig. 1.

We can see that the DLSI method exhibits lower error rates in most of the cases.

4. Conclusion and remarks

Basically the length of a differential vector of two normalized vector carries the same amount of information as the angle of the two vectors does. But a vector itself is a quantity that has not only a magnitude but also a direction. We have made full use of the differential vectors for pairs of normalized vectors rather than the mere scalar cosines of the angles of vectors in our document classification procedure, providing a more effective tool to the document classification. This should be the main reason that our DLSI-based classification model performs much better than a simple vector document model.

Just like the LSI method, a DLSI space can be regarded as a subspace for dimensionality reduction in document analysis. In the LSI-based scheme, only the projection of a document on a reduced space is used, while completely ignoring the effect of the distance from the document to the reduced spaces which constitutes a unique characteristic of the individual document. By introducing the concepts of differential intra- and extra-document vectors, our DLSI-based classification model is able to consider both of the projections on and the distances to the DLSI spaces from the differential vectors, improving the adaptability of the conventional LSI-based method to the unique characteristics of the individual documents which is a common weakness of the global projection schemes including the LSI approach. Consequently, the new classifier demonstrates an improved and robust performance in document classification than the LSI-based cosine approach.

The experiments demonstrate convincingly that the performance of the DLSI-based model outperforms both the standard LSI space-based approach and the simple term-based vector space model.

Acknowledgements

The authors are most grateful to the reviewers for many helpful comments to the original version of the paper. An early version of this paper has been presented at The Sixth International Workshop on Information Retrieval with Asian Languages, Sapporo, Japan, July 7, 2003.

References

- [1] J. Farkas, Generating document clusters using thesauri and neural networks, in: *Canadian Conference on Electrical and Computer Engineering*, Vol. 2, 1994, pp. 710–713.
- [2] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, *Machine Learning* 39 (2/3) (2000) 103–134.
- [3] H. Schütze, C. Silverstein, Projections for efficient document clustering, in: *Proc. SIGIR'97*, 1997, pp. 74–81.
- [4] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, *J. Opt. Soc. Amer. A* 4 (3) (1987) 519–524.
- [5] M. Turk, A. Pentland, Eigenfaces for recognition, *J. Cognitive Neurosci.* 3 (1) (1991) 71–86.
- [6] M.W. Berry, Z. Drmac, E.R. Jessup, Matrices, vector spaces, and information retrieval, *SIAM Rev.* 41 (2) (1999) 335–362.
- [7] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: *Proc. ACL-95*, 1995, pp. 189–196.
- [8] B. Moghaddam, W. Wahid, A. Pentland, Beyond eigenfaces: Probabilistic matching for face recognition, in: *Proc. IEEE Internat. Conf. on Automatic Face & Gesture Recognition*, Nara, Japan, April 1998, pp. 30–35.
- [9] D.L. Lee, H. Chuang, K. Seamons, Document ranking and the vector-space model, *IEEE Software* 14 (2) (1997) 67–75.
- [10] D.D. Lewis, Representation and learning in information retrieval, Ph.D. thesis, Computer Science Dept., Univ. of Massachusetts, Amherst, MA, 1992, Technical Report 91-93.
- [11] A. Aizawa, Linguistic techniques to improve the performance of automatic text categorization, in: *Proc. 6th Natural Language Processing Pacific Rim Symposium (NLPRS2001)*, 2001, pp. 307–314.
- [12] C.M. Tan, Y.F. Wang, C.D. Lee, The use of bigrams to enhance text categorization, *J. Inform. Process. Management* 30 (4) (2002) 529–546.