A Differential LSI Method for Document Classification

Liang Chen
Computer Science Department
University of Northern British Columbia
Prince George, BC, Canada V2N 4Z9
chenl@unbc.ca

Naoyuki Tokuda
R & D Center, Sunflare Company
Shinjuku-Hirose Bldg., 4-7 Yotsuya
Sinjuku-ku, Tokyo, Japan 160-0004
tokuda_n@sunflare.co.jp

Akira Nagai
Advanced Media Network Center
Utsunomiya University
Utsunomiya, Tochigi, Japan 321-8585
anagai@cc.utsunomiya-u.ac.jp

Abstract

We have developed an effective probabilistic classifier for document classification by introducing the concept of the differential document vectors and DLSI (differential latent semantics index) spaces. A simple posteriori calculation using the intra- and extra-document statistics demonstrates the advantage of the DLSI space-based probabilistic classifier over the popularly used LSI space-based classifier in classification performance.

1 Introduction

This paper introduces a new efficient supervised document classification procedure, whereby given a number of labeled documents preclassified into a finite number of appropriate clusters in the database, the classifier developed will select and classify any of new documents introduced into an appropriate cluster within the learning stage.

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 document 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(Li et al., 1991; Farkas, 1994; Svingen, 1997; Hyotyniemi, 1996; Merkl, 1998; Benkhalifa et al., 1999; Iwayama and Tokunaga, 1995; Lam and Low, 1997; Nigam et al., 2000).

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 (Schütze and Silverstein, 1997), the first problem to be resolved is the dimensionality reduction scheme enabling the documents' term projection onto a smaller subspace.

Like an eigen-decomposition method extensively used in image processing and image recognition (Sirovich and Kirby, 1987; Turk and Pentland, 1991), the Latent Semantic Indexing (LSI) method has proved to be a most efficient method for the dimensionality reduction scheme in document analysis and extraction, providing a powerful tool for the classifier (Schütze and Silverstein, 1997) when introduced into document retrieval with a good performance confirmed by empirical studies (Deerwester et al., 1990; Berry et al., 1999; Berry et al., 1995).The LSI method has also demonstrated its efficiency for automated cross-language document retrieval in which no query translation is required (Littman et al., 1998).

In this paper, we will show that exploiting both of the distances to, and the projections onto, the LSI space improves the performance as well as the robustness of the document classifier. To do this, we introduce, as the major vector space, the differential LSI (or DLSI) space which is formed from the differences between normalized intra- and extra-document vectors and normalized centroid vectors of clusters where the intra- and extra-document refers to the documents included within or outside of the given cluster respectively. The new classifier sets up a Baysian posteriori probability function for the differential document vectors based on their projections on DLSI space and their distances to the DLSI space, the document category with a highest probability is then selected. A similar approach is taken by Moghaddam and Pentland for image recognition (Moghaddam and Pentland, 1997; Moghaddam et al., 1998).

We may summarize the specific features introduced into the new document classification scheme based on the concept of the differential document vector and the DLSI vectors:

1. Exploiting the characteristic distance of the differential document vector to the DLSI space and the projection of the differential document onto the DLSI space, which we believe to denote the differences in word usage between the document and a cluster's centroid vector, the differential document vector is capable of capturing the relation between the particular document and the cluster.

2. A major problem of context sensitive semantic grammar of natural language related to synonymy and polysemy can be dampened by the major space projection method endowed in the LSIs used.

3. A maximum for the posteriori likelihood function making use of the projection of differential document vector onto the DLSI space and the distance to the DLSI space provides a consistent computational scheme in evaluating the degree of reliability of the document belonging to the cluster.

The rest of the paper is arranged as follows: Section 2 will describe the main algorithm for setting up the DLSI-based classifier. A simple example is computed for comparison with the results by the standard LSI based classifier in Section 3. The conclusion is given in Section 4.

2 Main Algorithm

2.1 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 $t_1, t_2, \cdots, t_m$.

For each document j in the collection, we assign each of the terms with a real vector $(a_{1j}, a_{2j}, \cdots, a_{mj})$, with $a_{ij} = f_{ij} \times g_i$, where f_{ij} is the local weighting 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 logarithms of occurrence counts. 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 (Raghavan and Wong, 1986; Luhn, 1958; van Rijsbergen, 1979; Salton, 1983; Salton, 1988; Lee et al., 1997), and will not be discussed further in this paper. An example will be given below:

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

where $p_{ij} = \frac{O_{ij}}{d_i}$, 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 document vector $(a_{1j}, a_{2j}, \cdots, a_{mj})$ can be normalized as $(b_{1j}, b_{2j}, \cdots, b_{mj})$ by the following formula:

$$b_{ij} = a_{ij} / \sqrt{\sum_{k=1}^{m} a_{kj}^2}. \tag{1}$$

The normalized centroid vector $C = (c_1, c_2, \cdots, c_m)$ of a cluster can be calculated in terms of the normalized vector as $c_i = s_i / \sqrt{\sum_{j=1}^{m} s_i^2}$, where $(s_1, s_2, \cdots, s_m)^T$

is a mean vector of the member documents in the cluster which are normalized as $T_1, T_2, \cdots, T_k$; i.e., $(s_1, s_2, \cdots, s_m)^T = \frac{1}{k}\sum_{j=1}^{k} T_j$. We can always take C itself as a normalized 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 of 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 of different clusters.

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

2.2 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 respectively with the first r columns of U and V being the eigenvectors of DD^T and D^TD respectively. Here S is called singular matrix expressed by $S = \text{diag}(\delta_1, \delta_2, \cdots, \delta_q)$, where δ_i are nonnegative square roots of 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 , deleting 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 approximately equals 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 $\bar{x}$ of the differential intra- (extra-) document vectors are approximately 0, we may assume that the differential vectors formed follows a high-dimensional Gaussian distribution so that the likelihood of any differential vector x will be given by

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

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 eigenvalues of DD^T, we have $S^2 = U^TDD^TU$, and thus we have $d(x) = nx^T(DD^T)^{-1}x = nx^TUS^{-2}U^Tx = ny^TS^{-2}y$, where $y = U^Tx = (y_1, y_2, \cdots, y_n)^T$.

Because S is a diagonal matrix, $d(x)$ can be represented by a simpler form as: $d(x) = n\sum_{i=1}^{r} y_i^2/\delta_i^2$. It is most convenient to estimate it 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 be estimated by

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

where $y = U_k^T x$, $\epsilon^2(x) = ||x||^2 - \sum_{i=1}^{k} y_i^2$, $\rho = \frac{1}{r-k}\sum_{i=k+1}^{r}\delta_i^2$, 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 equation (2), the term $\sum \frac{y_i^2}{\delta_i^2}$ describes the projection of x onto the DLSI space, while $\epsilon(x)$ approximates the distance from x to DLSI space.

When both $P(x|D_I)$ and $P(x|D_E)$ are computed, the Baysian 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 $1 - P(D_I)$.

2.3 Algorithm

2.3.1 Setting up the DLSI Space-Based Classifier

1. By preprocessing documents, identify terms either of the word and noun phrase from 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 column is an differential intra-document vector[2].

5. Decompose D_I, by an SVD algorithm, into $D_I = U_I S_I V_I^T$ ($S_I = \text{diag}(\delta_{I,1}, \delta_{I,2}, \cdots)$), 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:
$P(x|D_I) =$
$$\frac{n_I^{1/2} \exp\left(-\frac{n_I}{2}\sum_{i=1}^{k_I}\frac{y_i^2}{\delta_{I,i}^2}\right) \cdot \exp\left(-\frac{n_I \epsilon^2(x)}{2\rho_I}\right)}{(2\pi)^{n_I/2}\prod_{i=1}^{k_I}\delta_{I,i}\cdot \rho_I^{(r_I-k_I)/2}}, \quad (3)$$
where $y = U_{k_I}^T x$, $\epsilon^2(x) = ||x||^2 - \sum_{i=1}^{k_I} y_i^2$, $\rho_I = \frac{1}{r_I - k_I}\sum_{i=k_I+1}^{r_I}\delta_{I,i}^2$, 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 column is an extra- differential document vector.

7. Decompose D_E, by exploiting the SVD algorithm, into $D_E = U_E S_E V_E^T$ ($S_E = \text{diag}(\delta_{E,1}, \delta_{E,2}, \cdots)$), 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,
$P(x|D_E) =$
$$\frac{n_E^{1/2} \exp\left(-\frac{n_E}{2}\sum_{i=1}^{k_E}\frac{y_i^2}{\delta_{E,i}^2}\right) \cdot \exp\left(-\frac{n_E \epsilon^2(x)}{2\rho_E}\right)}{(2\pi)^{n_E/2}\prod_{i=1}^{k_E}\delta_{E,i}\cdot \rho_E^{(r_E-k_E)/2}}, \quad (4)$$
where $y = U_{k_E}^T x$, $\epsilon^2(x) = ||x||^2 - \sum_{i=1}^{k_E} y_i^2$, $\rho_E = \frac{1}{r_E - k_E}\sum_{i=k_E+1}^{r_E}\delta_{E,i}^2$, 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:
$$P(D_I|x) = \frac{P(x|D_I)P(D_I)}{P(x|D_I)P(D_I) + P(x|D_E)P(D_E)}, \quad (5)$$
$P(D_I)$ is set to $1/n_c$ where n_c is the number of clusters in the database and $P(D_E)$ is set to $1 - P(D_I)$.

2.3.2 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 from equation (1).

   For each of the clusters of the data base, repeat the procedure of item 2-4 below.

2. Using the document to be classified, construct a differential document vector $x = N - C$, where

[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] For a cluster with s elements, we may include at most $m-1$ differential intra-document vectors in D_I to avoid the linear dependency among columns

C is the normalized vector giving 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 Examples and Comparison

3.1 Problem Description

We demonstrate our algorithm by means of numerical examples 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, as an example, below how we will set up the classifier to classify the following new document:

N: "The Elements of Computing Science."

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 simply by counting the "common" words among documents.

We will now set up the DLSI-based classifier and LSI-based classifier 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: $(0,0,0,0,1,0,0,1,0,0,0,0,0,0,1,0,0,0)^T$, and in normalized form by $(0,0,0,0,0.577350269,0,0,0.577350269,0,0,0,0,0,0,0.577350269,0,0,0)^T$.

3.2 DLSI Space-Based Classifier

The normalized form of the centroid of each cluster is shown in Table 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 $D_I^{18\times 4} = (T_1 - C_1, T_3 - C_2, T_5 - C_3, T_7 - C_4)$ and the differential term by extra-document matrix by $D_E^{18\times 4} = (T_2 - C_2, T_4 - C_3, T_6 - C_4, T_8 - C_1)$ respectively. Note that the T_i's and C_i's can be found in the matrices shown in tables 1 and 2.

Now that we know D_I and D_E, 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, where

$$U_I = \begin{pmatrix} 0.25081 & 0.0449575 & -0.157836 & -0.428217 \\ 0.130941 & 0.172564 & 0.143423 & 0.0844264 \\ -0.240236 & 0.162075 & -0.043428 & 0.257507 \\ -0.25811 & -0.340158 & -0.282715 & -0.166421 \\ -0.237435 & -0.125328 & 0.439997 & -0.15309 \\ 0.300435 & -0.391284 & 0.104845 & 0.193711 \\ 0.0851724 & 0.0449575 & -0.157836 & 0.0549164 \\ 0.184643 & -0.391284 & 0.104845 & 0.531455 \\ -0.25811 & -0.340158 & -0.282715 & -0.166421 \\ 0.135018 & 0.0449575 & -0.157836 & -0.0904727 \\ 0.466072 & -0.391284 & 0.104845 & -0.289423 \\ -0.237435 & -0.125328 & 0.439997 & -0.15309 \\ 0.296578 & 0.172564 & 0.143423 & -0.398707 \\ -0.124444 & 0.162075 & -0.043428 & -0.0802377 \\ -0.25811 & -0.340158 & -0.282715 & -0.166421 \\ -0.237435 & -0.125328 & 0.439997 & -0.15309 \\ 0.0851724 & 0.0449575 & -0.157836 & 0.0549164 \\ -0.124444 & 0.162075 & -0.043428 & -0.0802377 \end{pmatrix},$$

$$S_I = \text{diag}(0.800028, 0.765367, 0.765367, 0.583377),$$

$$V_I = \begin{pmatrix} 0.465291 & 0.234959 & -0.824889 & 0.218762 \\ -0.425481 & -2.12675\text{E-}9 & 1.6628\text{E-}9 & 0.904967 \\ -0.588751 & 0.733563 & -0.196558 & -0.276808 \\ 0.505809 & 0.637715 & 0.530022 & 0.237812 \end{pmatrix},$$

$$U_E = \begin{pmatrix} 0.00466227 & -0.162108 & 0.441095 & 0.0337051 \\ -0.214681 & 0.13568 & 0.0608733 & -0.387353 \\ 0.0265475 & -0.210534 & -0.168537 & -0.529866 \\ -0.383378 & 0.047418 & -0.195619 & 0.0771912 \\ 0.216445 & 0.397068 & 0.108622 & 0.00918756 \\ 0.317607 & -0.147782 & -0.27922 & 0.0964353 \\ 0.12743 & 0.0388027 & 0.150228 & -0.240946 \\ 0.27444 & -0.367204 & -0.238827 & -0.0825893 \\ -0.383378 & 0.047418 & -0.195619 & 0.0771912 \\ -0.0385053 & -0.38153 & 0.481487 & -0.145319 \\ 0.19484 & -0.348692 & 0.0116464 & 0.371087 \\ 0.216445 & 0.397068 & 0.108622 & 0.00918756 \\ -0.337448 & -0.0652302 & 0.351739 & -0.112702 \\ 0.069715 & 0.00888817 & -0.208929 & -0.350841 \\ -0.383378 & 0.047418 & -0.195619 & 0.0771912 \\ 0.216445 & 0.397068 & 0.108622 & 0.00918756 \\ 0.12743 & 0.0388027 & 0.150228 & -0.240946 \\ 0.069715 & 0.00888817 & -0.208929 & -0.350841 \end{pmatrix},$$

$$S_E = \text{diag}(1.67172, 1.47695, 1.45881, 0.698267),$$

$$V_E = \begin{pmatrix} 0.200663 & 0.901144 & -0.163851 & 0.347601 \\ -0.285473 & -0.0321555 & 0.746577 & 0.600078 \\ 0.717772 & -0.400787 & -0.177605 & 0.540952 \\ -0.60253 & -0.162097 & -0.619865 & 0.475868 \end{pmatrix}.$$

We now choose the number k in such a way that $\delta_k - \delta_{k+1}$ remains sufficiently large. Let us choose $k_I = k_E = 1$ and $k_I = k_E = 3$ to test the classifier. Now using equations (3), (4) and (5), 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 3. 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 $\rho_I = \frac{1}{r_I - k_I}\sum_{i=k_I+1}^{r_I} \delta_{I,i}^2$, and $\rho_E = \frac{1}{r_E - k_E}\sum_{i=k_E+1}^{r_E} \delta_{E,i}^2$. The last row of Table 3 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.

3.3 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.

Using the LSI-based classifier, our experiment show that, it will return C_3, namely "Physics", as the most likely cluster to which the document N belongs. This is obviously a wrong result.

3.4 Conclusion of the Example

For this simple example, the DLSI space-based approach finds the most reasonable cluster for the document "The elements of computing science", while the LSI approach fails to do so.

4 Conclusion and Remarks

We have made use of the differential vectors of two normalized vectors rather than the mere scalar cosine of the angle of the two vectors in document classification procedure, providing a more effective means of document classifier. Obviously the concept of differential intra- and extra-document vectors imbeds a richer meaning than the mere scalar measure of cosine, focussing the characteristics of each document wheere the new classifier demonstrates an improved and robust performance in document classification than the LSI-based cosine approach. Our model considers both of the projections and the distances of the differential vectors to the DLSI spaces, 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. The simple experiment demonstrates convincingly that the performance of our model outperforms the standard LSI space-based approach. Just as the cross-language ability of LSI, DLSI method should also be able to be used for document classification of docuements in multiple languages. We have tested our method using larger collection of texts, we will give details of the results elsewhere. .

References

M. Benkhalifa, A. Bensaid, and A Mouradi. 1999. Text categorization using the semi-supervised fuzzy c-means algorithm. In *18th International Conference of the North American Fuzzy Information Processing Society*, pages 561–565.

Michael W. Berry, Susan T. Dumais, and G. W. O'Brien. 1995. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37:573–595.

Michael W. Berry, Zlatko Drmac, and Elizabeth R. Jessup. 1999. Matrices, vector spaces, and information retrieval. *SIAM Rev.*, 41(2):335–362.

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: The normalized cluster centers

	C_1	C_2	C_3	C_4
Algebra	0.353553391	0.311446376	0	0
Alumina	0	0	0	0.5
Analysis	0	0.359627298	0.40824829	0
Chemical	0	0	0	0.40824829
Compute	0.40824829	0	0	0
Dynamics	0	0	0.40824829	0
Education	0.353553391	0	0	0
Element	0	0.359627298	0.40824829	0
Foundation	0	0	0	0.40824829
Geometry	0.353553391	0.671073675	0	0
Introduction	0	0.311446376	0.40824829	0
Machine	0.40824829	0	0	0
Modern	0	0.311446376	0	0.5
Physics	0	0	0.40824829	0
Science	0	0	0	0.40824829
Software	0.40824829	0	0	0
System	0.353553391	0	0	0
Theory	0	0	0.40824829	0

Table 3: Classification with DLSI space-based classifier

	$k_I = k_E = 1$				$k_I = k_E = 3$			
x:	$N-C_1$	$N-C_2$	$N-C_3$	$N-C_4$	$N-C_1$	$N-C_2$	$N-C_3$	$N-C_4$
$U_{k_I}^T x$	-0.085338834	-0.565752063	-0.368120678	-0.077139955	-0.085338834 -0.404741071 -0.164331163	-0.556196907 -0.403958563 0.249931018	-0.368120678 -0.213933843 0.076118938	-0.077139955 -0.250613624 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 -0.276920807 -0.753558043	-0.01964297 0.6527666 -0.619983845	-0.326936108 0.475906836 0.258017361	0.807673935 -0.048681069 -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

Scott Deerwester, Susan T. Dumais, Grorge W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Jennifer Farkas. 1994. Generating document clusters using thesauri and neural networks. In *Canadian Conference on Electrical and Computer Engineering*, volume 2, pages 710–713.

H. Hyotyniemi. 1996. Text document classification with self-organizing maps. In *STeP '96 - Genes, Nets and Symbols. Finnish Artificial Intelligence Conference*, pages 64–72.

M. Iwayama and T. Tokunaga. 1995. Hierarchical bayesian clustering for automatic text classification. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 2, pages 1322–1327.

Wai Lam and Kon-Fan Low. 1997. Automatic document classification based on probabilistic reasoning: Model and performance analysis. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2719–2723.

D. L. Lee, Huei Chuang, and K. Seamons. 1997. Document ranking and the vector-space model. *IEEE Software*, 14(2):67–75.

Wei Li, Bob Lee, Franl Krausz, and Kenan Sahin. 1991. Text classification by a neural network. In *Proceedings of the Twenty-Third Annual Summer Computer Simulation Conference*, pages 313–318.

M. L. Littman, Fan Jiang, and Greg A. Keim. 1998. Learning a language-independent representation for terms from a partially aligned corpus. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 314–322.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, April.

D. Merkl. 1998. Text classification with self-organizing maps: Some lessons learned. *Neurocomputing*, 21(1-3):61–77.

B. Moghaddam and A. Pentland. 1997. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):696–710.

B. Moghaddam, W. Wahid, and A. Pentland. 1998. Beyond eigenfaces: Probabilistic matching for face recognition. In *The 3rd IEEE Int'l Conference on Automatic Face & Gesture Recognition*, Nara, Japan, April.

Kamal Nigam, Andrew Kachites MaCcallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3):103–134, May.

V. V. Raghavan and S. K. M. Wong. 1986. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–87.

Gerard Salton. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

Gerard Salton. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–524.

Hinrich Schütze and Craig Silverstein. 1997. Projections for efficient document clustering. In *Proceedings of SIGIR'97*, pages 74–81.

L. Sirovich and M. Kirby. 1987. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524.

Borge Svingen. 1997. Using genetic programming for document classification. In John R. Koza, editor, *Late Breaking Papers at the 1997 Genetic Programming Conference*, pages 240–245, Stanford University, CA, USA, 13–16 July. Stanford Bookstore.

M. Turk and A. Pentland. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

C. J. van Rijsbergen. 1979. *Information retrieval*. Butterworths.