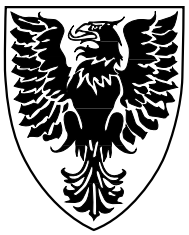

A Comparative Study of Dimension Reduction Techniques for Document Clustering

Bin Tang
Xiao Luo
Malcolm I. Heywood
Michael Shepherd

Technical Report CS-2004-14

December 6, 2004

Faculty of Computer Science
6050 University Ave., Halifax, Nova Scotia, B3H 1W5, Canada

A Comparative Study of Dimension Reduction Techniques for Document Clustering

Bin Tang, Xiao Luo, Malcolm I.Heywood, Michael Shepherd

Faculty of Computer Science,
Dalhousie University, 6050 University Avenue,
Halifax, Nova Scotia, Canada, B3H 1W5
{btang, shepherd, mheywood, luo}@cs.dal.ca

Abstract

Dimension reduction techniques (DRT) are applicable to a wide range of information systems. Application context naturally has a significant impact on the appropriateness of the DRTs. In this research, a systematic study is conducted of four DRTs for the text clustering problem using five benchmark datasets. Of the four methods -- Independent Component Analysis (ICA), Latent Semantic Indexing (LSI), Document Frequency (DF) and Random Projection (RP) -- ICA and LSI are clearly superior when the k-means clustering algorithm is applied, irrespective of the datasets. Random projection consistently returns the worst results, where this appears to be due to the noise distribution characterizing the document clustering task.

Keywords-dimension reduction techniques, text clustering, Independent Component Analysis, Latent Semantic Indexing, Random Projection, Document Frequency.

1 Introduction

The wide spread availability of Internet technology and hardware capacity has led to an exponential increase of the amount of documents available electronically across all the application fields. The huge amount of text information provides a natural

requirement for efficient document organization, summarization, navigation and retrieval. Document clustering is the fundamental and enabling tool for all these applications. In general, document clustering can be defined as, given a document collection, *S*, of *n* documents in a high dimensional space, to find a partition of *S* such that the documents within each cluster are similar to each other. This is something of a "wish"; reality might require classification where the overhead is the need for labeled training data.

The more general clustering problem has long been of interest to many research communities, including fuzzy logic [5, 12, 33], statistical learning [8], information theory [13, 40], neural networks [9, 19, 31, 39] and database [20, 43] communities. In general, common wisdom agrees that no single clustering algorithm can properly manage all the complexities inherent in all datasets equally well. These complexities include different density distributions and density levels, different shapes, sub-cluster structures and heterogeneous subspaces for different clusters.

One specific characteristic that makes document clustering particularly difficult among the general clustering problems is the high dimensionality of the problem. This adds an extra very difficult characteristic to the standard clustering problem. In most common text applications, documents are represented by a vector in an *m*-dimensional term space, where *m* is the number of different terms occurring in the dataset of documents. It is not uncommon to find thousands or tens of thousands of different words for even a relatively small sized text data collection of a few thousand documents. Moreover, only a subset of the different terms will appear in any one document, resulting in documents being described by a sparse but multidimensional feature vector.

The high dimensionality of natural text is often referred to as the "curse of dimensionality". In the context of clustering, the commonly used distance measures

between data points begin to lose discriminative power as the number of dimensions increases for the given dataset. It has been shown that, in a high dimensional space, data points almost always have equal distance to each other for various data distributions and distance functions [4].

To solve the high dimensionality problem, various dimension reduction techniques have been proposed [17, 37]. There are two major types of dimension reduction techniques, feature transformation and feature selection [37]. In feature transformation, the original high dimensional space is projected onto a lower dimensional space, in which each dimension in the lower dimensional space is some linear or non-linear combination of the original high dimensional space. Widely used examples include, Principle Component analysis (PCA), Factor Analysis, Projection Pursuit, Latent Semantic Indexing (LSI), Independent Component Analysis (ICA), and Random Projection (RP) [17]. In the case of feature transformation, the lower dimensional space is often believed to represent the underlying latent structure of the dataset. Such a transformation either has to guarantee a good degree of distance preservation among data points or generate statistically more independent components of the original dataset. However, it has been argued that such methods are less effective when a huge number of irrelevant dimensions are present in the dataset compared to the number of meaningful dimensions [37]. Feature selection, as the name implies, only has to select a subset of "meaningful or useful" dimensions (specific for the application) from the original set of dimensions. A current trend in feature selection is therefore to select relevant subspaces appropriate for each cluster separately [2, 38].

Not all the dimension reduction techniques have been used widely in text applications such as text categorization, clustering and information retrieval tasks (IR). In

this work, we are only interested in investigating the relative effectiveness and robustness of a few dimension reduction techniques when used for text clustering. They are Document Frequency (DF), Latent Semantic Indexing (LSI), Random Projection (RP) and Independent Component Analysis (ICA). More detailed reviews of the four DRTs will be introduced in the next section.

Although many research projects are actively engaged in furthering DRTs as a whole, so far, there is no experimental work comparing them in a systematic manner. As data miners, we feel strongly that a systematic comparative study on these four techniques be conducted in the context of text clustering, using benchmark datasets of differing characteristics.

This paper is organized as follows. Section 2 provides more details for the four DRTs used in this research. Section 3 describes the general experimental procedure and evaluation methods that we use in this work. Section 4 describes the characteristics of the datasets used and the pre-processing procedure followed. Section 5 presents our experimental results and appropriate discussion notes. Finally, conclusions are drawn and future research directions identified in Section 6.

2 Some math details of the DRTs discussed in this paper

Document Frequency (DF)

Document Frequency (DF) may itself be used as the basis for feature selection. That is, only those dimensions with high DF values appear in the feature vector. In spite of its simplicity, it has been demonstrated to be as effective as more advanced techniques in text categorization [42]. We are curious to know how effective DF will be as a

dimension reduction technique for text clustering, and what preprocessing methodology is the most appropriate when using DF as a dimension reduction technique in text clustering.

DF can be formally defined as follows. For a document collection in matrix notation, $A_{m \times n}$, with *m* terms and *n* documents, the *DF* value of term 't', DF_t, is defined as the number of documents in which *t* occurs at least once among the *n* documents. To reduce the dimensionality of *A* from *m* to *k (k<m)*, we choose to use the *k* dimensions with the top *k DF* values. It is obvious that the DF takes *O(mn)* to evaluate.

Latent Semantic Indexing (LSI)

LSI, as one of the standard dimension reduction techniques in information retrieval, has enjoyed long-lasting attention [3, 11, 14, 15, 16, 23, 24, 36]. It was initially designed to be an effective automatic indexing and retrieval tool. By detecting the high-order semantic structure (term-document relationship), it addresses the ambiguity problem of natural language, i.e., the use of synonymous, near- synonymous, and polysemous words. It uses Singular Value Decomposition (SVD) to embed the original high dimensional space into a lower dimensional space with minimal distance distortion, in which the dimensions in this space are orthogonal (statistically uncorrelated). Using truncated SVD, not only can we capture the most important association between terms and documents, but also we can effectively remove noise and redundancy and word ambiguity within the dataset [3]. A probabilistic variant of LSI, pLSI, has been proposed recently [23] which defines a generative model for directly minimizing word perplexity based on the principle of maximum likelihood.

There are a few drawbacks associated with LSI. The first is its high computational cost. For a data matrix, $A_{m \times n}$, the time complexity can be estimated in the order of

$O(m^2n)+ O(m^3)$ [21]. Even for a sparse matrix of the same size, *m x n*, with computational more efficient methods, the time complexity is still at the order of *O(cmn)*, where *c* is the average non-zero values over all the data vectors [36]. In our research, we use the *svds* function in Matlab™, the SVD for sparse matrices. Another factor essential to the success of LSI is the choice of "k", the dimension to reduce to, as noticed by Deerwester et al [11]. In practice, the value of k is often determined *at hoc*. Though some theoretical works attempt to provide formal frameworks to determine the value of "k", these theoretically optimal k values seem to be either too high or not always in agreement with the IR performance [15, 16]. In this research, besides the effectiveness of LSI, we are also strongly interested in finding some general rules to determine the "close to optimal" k value for LSI in document clustering applications.

LSI uses the well-known Singular Value Decomposition (SVD) algorithm to reduce the dimensionality. The process of SVD takes the given document collection $A_{m\,x\,n}$ and decomposes it into three matrices in the form of: $A = U_r \bullet S_r \bullet V_r^T$, where *U* and *V* are orthogonal matrices that contain the left and right singular vectors of *A* respectively, *S* is the diagonal matrix that contains the singular values of A and the subscript *r* denotes the rank of *A*. *U* and *V* are often referred to as term vectors and document vectors respectively. Since the singular values are often sorted in descending order, truncated SVD can be used to project the original data onto a lower, *k*-dimensional space, which is the best rank-k approximation of *A* in the least-square sense. *A* is approximated as $\widetilde{A}_k \cong \widetilde{U}_k \bullet \widetilde{S}_k \bullet \widetilde{V}_k^T$. Under the new *k*-dimension, each original document *d* can be re-represented as $\widetilde{d} = U_k S_k d^T$, where $U_k S_k$ is often referred to as the term projection matrix.

The work of [24] argues that the term projection matrix should be normalized

before being used to project the document onto the lower dimension, which improves the performance on information retrieval tasks. In this work, we will first test whether normalization of the term projection matrix is beneficial for text clustering. Then we will choose the proper form of the projection matrix to be used in LSI.

Random projection (RP)

Recently, the method of Random Projection (RP) was developed to provide a low (computational) cost alternative to LSI for dimension reduction. Naturally, researchers in the text mining and information retrieval communities have become strongly interested in RP. RP has proven to a reasonably good alternative to LSI in preserving the mutual distances among documents [6]. Some researchers believe that RP is a good alternative DRT for classifiers similar to kNN and other clustering methods [10, 28, 29]. However, other researchers are not that convinced of the effectiveness, computational or otherwise, of RP as an alternative for LSI-like techniques [18,34]. So far, the effectiveness of RP is still unclear.

RP was initially proposed as a computationally cheaper alternative to LSI [22, 29]. Similar to LSI, RP projects the original high dimensional space onto a lower dimensional space using a randomly generated projection matrix, $\widetilde{A}_{[k\times n]} = R_{[k\times m]} \bullet A_{[m\times n]}$, where the columns of R follow Gaussian distribution with unit length. Unlike the newly generated space in LSI in which all dimensions are orthogonal, the dimensions in the newly generated k-dimensional space of RP are approximately orthogonal. It can be proven that the distance distortion error introduced by RP under the new dimensional space is well-bounded [1, 29]. In this work, we follow an approximation scheme to generate the normalized projection matrix as used in [1]. For any column of projection matrix, R, the element r_{ij} is generated from the following probability distribution,

$$r_{ij} = \sqrt{3} \times \begin{cases} +1 & with \quad probability \quad 1/6 \\ 0 & .. \qquad .. \qquad 2/3. \\ -1 & .. \qquad .. \qquad 1/6 \end{cases}$$

It is easy to see that the time complexity to generate R is *O(mk)*.

A theoretical concern for the appropriate usage of RP arises from the ambiguity and "noise level" inherent in natural language. It has long been suspected that the original high dimensional term space is not appropriate for random sampling due to the ambiguity of the meaning of the terms in natural languages [11]. Unlike other feature transformation methods that generate new features based on some statistical property, the new features in RP are generated randomly (random linear combinations of original terms). This randomness may add further disruption to the ambiguity found in natural language and diminish the effectiveness of RP as a DRT for text clustering. In general, we really want to understand how effective RP is for text clustering.

Independent component analysis (ICA)

A recent method of feature transformation called Independent Component Analysis (ICA) has gained widespread attention in signal processing [25]. It is a general-purpose statistical technique, which tries to linearly transform the original data into components that are maximally independent from each other in a statistical sense. Unlike LSI or PCA, the independent components are not necessarily orthogonal to each other, but are statistically independent. This is a stronger condition than statistical uncorrelateness, as used in PCA or LSI [25]. ICA can be used as a dimension reduction technique. It can also be used to estimate the latent variables of a given dataset. So far, it has enjoyed good success in many different areas, such as signal processing, telecommunication, and economics [27].

Until very recently, there are only a few experimental works in which ICA is

applied to text applications. It has been used as an indexing tool instead of LSI [30]. ICA has been compared favorably to LSI in producing representations better aligned with the grouping structure of the given text [32]. An extension of standard ICA to streaming data has been used successfully for identifying topics in a dynamical textual environment, i.e., chat room conversation streams [7]. So far, the applications of ICA to text applications are still atypical. As a DRT with great potential, more research is needed to demonstrate its effectiveness for text clustering, especially when compared to other well-known methods.

ICA is defined under a generative model, i.e., it assumes each observed data (a document *x*) being generated by a mixing process of statistically independent components (latent variables s_i). Formally, using vector-matrix notation, the noise-free mixing model can be written as $X_{m\times n} = A_{m\times k} S_{k\times n}$, where A is often referred to as the mixing matrix and the inverse of *A* is often referred as the unmixing matrix, *W*. The independent components can be expressed as $S_{k\times n} = W_{k\times m} X_{k\times n}$. Here, the statistical independence is equivalent to nongaussianity. The problem of ICA is to use *X* to simultaneously estimate both the mixing matrix, *A*, and the independent components, *S*. The objective function of ICA (contrast function) measures the nongaussianity of components, which should be maximized during the ICA process.

Software packages have been developed for implementing the basic ICA algorithms, e.g., JADE_{TD} and FastICA [27, 35]. In addition, FastICA is known to be robust and efficient for a wide range of underlying distributions [17]. In this research, we used fixed-point ICA, the FastICA implementation [27]. The following introduction is mainly based on FastICA.

For the one component ICA, FastICA tries to find a vector *w* whose projection of

$w^T x$ maximizes nongaussianity. Nongaussianity is measured by the approximation of negentropy $J(w^T x)$. $J(y)$ is defined as $J(y) \approx c[E\{G(y)\}-E\{G(\upsilon)\}]^2$, where c is an constant, υ is a Gaussian variable of zero mean and unit variance, y is also assumed to be of zero mean and unit variance, and G is any nonquadratic function. The following choices of G are often appropiate: $G_1(u)=(1/a_1)\log\cosh(a_1 u)$, $G_2(u)=-\exp(-u^2/2)$, where $1 \le a_1 \le 2$ is some suitable constant (in the following, the derivative of G is denoted as g. The basic procedure of FastICA of one unit is as follows:

a. Randomly choose an initial weight vector w.
b. Let $w^+ = E\{x g(w^T x)\} - E\{g'(w^T x)\} w$
c. Let $w = w^+ / \|w^+\|$
d. Go back to b until converge

To estimate multiple components, one can either use a deflation scheme, or estimate all components simultaneously in symmetric manner and use orthogonal decorrelation $W=(WW^T)^{-1/2}W$, where W is the matrix $(w_1, \ldots w_n)^T$ of the weight vectors. Further details can be found elsewhere [26]. It is easy to see that run time of FastICA is determined by its convergence speed. It is argued that this algorithm converges in at least quadratic time [26].

In practical applications of FastICA, often, there are two pre-processing steps. The first is centering, i.e., making x zero-mean variables. The second is whitening, which means that we linearly transform the observed vector x to x^{new}, such that its components are uncorrelated and their variance equal unity. Whitening is done through eigenvalue decomposition as used in PCA. In practice, the most time consuming part of FastICA is the whitening, which can be computed with *svds* in Matlab™.

3 Evaluation methods and general experimental procedure

Evaluation methods

To judge the relative effectiveness of the DRTs, we first apply them to text clustering tasks on different datasets. Based on the quality of their clustering results, we rank them accordingly. There are two perspectives to the ranking, the absolute clustering results and the robustness of the method. Here, good robustness implies that when using a certain DRT, reasonably good clustering results should be found across a relatively wide range of dimensions (reduced), i.e., the clustering results should degrade gracefully if non-optimal reduced dimensions are used. Obviously, robustness is a highly desirable property of a DRT for text clustering and other data mining tasks.

There are many ways to measure the quality of text clustering. We believe that we should use the class labels of the data as relevant references to judge the quality of clustering results. Hence, we choose to use *Purity*, which measures the percentage of the data points in the cluster that belong to primarily one class [44]. It is defined as the weighted sum of individual cluster purities: $Purity = \sum_{i=1}^{C} \frac{n_i}{n} P(S_i)$. Here, $P(S_i)$ is the purity for a particular cluster of size n_i, C is the total number of clusters and n is the total number of points in the dataset. Since we divide the whole dataset into training and testing set, we modify the calculation of Purity as follows. Each cluster i is assigned a class label, T_i, based on a majority vote by its members using only the training set. Then, $P(S_i)$ of cluster i is defined as the proportion of points assigned as members of cluster i in the test set whose class labels agree with T_i. Obviously, the higher the Purity value, the purer the cluster in terms of the class labels of its members, the better the clustering results. It is easy to establish that Purity is the clustering-version of the micro-average of

classification accuracy when the classification accuracy is micro-averaged over all the clusters instead of the classes. Hereafter, we refer to the cluster quality measure as *classification accuracy (CA)* instead of *Purity*.

To judge the relative robustness of DRTs, we combine a heuristic observation and student t test. We first plot the CA curves of the DRTs against the dimensionalities. Based on the CA values, it is visually possible to clearly establish the relative effectiveness of the DRTs based on these curves. For situations when more than one curve shares very similar CA values over "an interesting range of dimensions"(defined later), such that we cannot visually resolve performance levels, we will perform a paired student t test. For each dataset, the relative ranks of the DRTs are determined by the combination of visual observation and paired student t test on the CA curves of the DRTs.

To ensure that the results are representative and systematic, many precautions have to be taken in the process of comparison. First, the choice of datasets has to be made in such way that a broad genre of text collections are covered in our test. The second issue concerns the usage of the clustering algorithm. For the choice of clustering algorithm, we use k-means, since k-means or its variants are the most commonly used clustering algorithms used in text clustering [41]. It is a well-known problem that the clustering results of k-means are not always optimal and stable due to poor choices of initialization. In our implementation, a simple procedure, InitKMeans (defined later), is introduced to ameliorate the negative effect of poor initialization. The third issue concerns the proper usage of LSI and ICA. Though LSI is a standard procedure, different forms of the projection matrix used in LSI may have differing impacts on the quality of the transformation [24]. Such a concern may also be extended to the use of ICA, since so

little work has been done on ICA for text clustering. Therefore, in our experiments, proper forms of LSI and ICA are determined experimentally.

pseudocodes for experiment procedures

Our experiments follow a general procedure, briefly listed in Procedure DRT_Text_Clustering. Our initialization procedure for k_means is briefly listed in Procedure InitKMeans.

PROCEDURE DRT_Text_Clustering(test-dimensions, DRT, $AllData_{m \times n}$, # clusters)
BEGIN
1. Randomly split the AllData into training sets (data1) and test sets (Data1) of ratio 3:1 proportionally to their category distribution if possible.
2. Normalize data1 & Data1 such that each document has unit length.
3. Dimension reduction and clustering
   FOR each experimental dimension k,
      Apply DRT(data1,k) to either select a subset of dimensions to use or to generate the transformation matrix A at dimension k;
      Generate the dimension reduced version of data1, $data_k$;
      Renormalize $data_k$ to unit length for each document;
      InitKMeans ($data_k$, # clusters) returns seeds for k_means;
      Apply k_means using the seeds on $data_k$ to generate centers_k;
   END FOR
4. Calculate final results
   FOR each experimental dimension k,
      Reduce dimensions for data1 and Data1 to k and generate training and testing

data, $data_k$ and $Data_k$;

Normalize both $data_k$ and $Data_k$ to unit length for each document;

Label clusters with class label using majority vote of the members using only $data_k$;

Calculate classification accuracy for $Data_k$;

END FOR

END

PROCEDURE InitKMeans(data, n) // n is the number of seeds to return

BEGIN

1. Randomly select α% of data, D, proportionally to their category distribution.
2. Calculate the mutual distances between points in D.
3. For each point, sort its distance to all other points in ascending order.
4. For each point, define its neighborhood size, γ, as, its average distance to its first β closest neighbors. β is a small number, e.g., 6, 10.
5. Return the data points whose γs are ranked among the first n smallest.

END

All our experiments are conducted under Matlab 6.5.1 environment. The k_means, svds procedures are taken directly from Matlab toolboxes. The matlab code for ICA is from [27]. We implement rest of the codes with matlab.

4 Characteristics of the data sets

The Datasets

In our experiments, we used a variety of datasets, which include WWW-pages

(WebKB), newswire stories (Reuters-21578, 20Newsgroup), and technical reports (CSTR). Most of the datasets are widely used in the research of information retrieval and text mining. The number of classes ranges from 4 to 50 and the number of documents ranges between 4 and 3807 per class. Table 1 summarizes the characteristics of the datasets.

20Newsgroups [45] The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents evenly partitioned across 20 different newsgroups. The subset we used consists of four newsgroups, i.e., soc.religion.christian, sci.crypt, sci.med, and sci.space, referred to as 20NG-4. This subset has been used to prove the distance preservation property of RP in [6]. Therefore, we choose to use it to test the "real effectiveness" of RP when applied to real text clustering.

Reuters-21578 [46] Reuters-21578 is a standard multi-class, multi-labeled benchmark. It contains 12,902 newswire stories that have been classified into 118 categories. Two subsets of it are used. Reuter-2 is a collection of documents each document with a single topic label. The version of Reuter-2 that we used eliminates categories with less than 4 documents, leaving only 50 categories. The categories of Reuter-2 are very imbalanced. To partially remove the effect of the imbalance within Reuter-2, we derive a second subset from Reuters-2, referred to as Reuters-10, consisting only of the ten most frequent categories.

WebKB [45] The WebKB data set contains 8,282 WWW-pages collected from computer science departments of various universities in January 1997 by the CMU text learning group. We only used a subset, commonly referred to as the WebKB4, which is limited to the four most abundant categories: student, faculty, course, and project.

CSTR [47] CSTR is a collection of the abstracts of technical reports published

in the Department of Computer Science at the University of Rochester between 1990 and 2004. The dataset contains 505 abstracts, divided into four research areas: AI, Robotics and Vision, Systems, and Theory.

Table 1 Summarization of the datasets

Datasets	Dataset size (|terms| × |docs|)	#classes	Class Size range	Type
20NG-4	7694 × 4009	4	[997,1012]	News
Reuters-2	7315 × 8771	50	[4, 3807]	News
Reuters-10	6649 × 7720	10	[107, 3807]	News
WebKB4	9870 × 4199	4	[504, 1641]	University web pages
CSTR	2335 × 505	4	[76, 191]	Technical Reports

Preprocessing

The pre-processing of the datasets follows the most practiced procedures, including, removal of the tags and non-textual data, stop word removal [48], and stemming [49]. Then we further remove the words with low document frequency. For example, for the Reuter-2 dataset we only selected words that occurred in at least 4 documents.

The word weighting scheme we used is the "*ltc*" variant of the *tfidf* function, defined as follows:

$$tfidf(w_{k,}d_j) = tf(w_k, d_j) \cdot \log \frac{|T_r|}{\#T_r(w_k)}$$

$$tf(w_k, d_j) = \begin{cases} 1 + \log \#(w_k, d_j) & if\ \#(w_k, d_j) > 0 \\ 0 & otherwise \end{cases}$$

T_r – Total number of documents in collection D

$\#T_r(w_k)$ - Number of documents in D in which term w_k occurs at least once

$\#(w_k, d_j)$ - Frequency of term w_k in document d_j

5 Experiments results and discussions

5.1 Choosing proper form of LSI and ICA

In the first set of experiments, the objective is to determine the proper form of the LSI and ICA projection matrix, i.e., whether it is appropriate to normalize the projection matrix used in both methods. To do so, each method is applied to all the available datasets twice, once with the projection matrix normalized and once without. To compare their relative effectiveness, visual inspection is used, with paired student t tests if necessary. The results for LSI/LSI_Norm and ICA/ICA_Norm are plotted in Figures 1 and 2, respectively.

From Figures 1 and 2, for datasets of Reuter-2, Reuter-10 and WebKB4, it is obvious that normalization of the projection matrix for LSI/ICA is not necessary, since the non-normalized version of LSI/ICA consistently performs better than the normalized version. For 20NG-4 and CSTR, we conduct paired student t tests. We are only interested in comparing their performance on the most "interesting" dimension range. By "interesting" dimension range, we refer to the dimension range within which the methods produced the best clustering results. Hereafter, we will use [a, b] to denote the "interesting" dimension range under investigation. The results of the t-tests are listed in Table 2. Based on the p values at α=0.05 level, we observe that for most of the cases, there is no significant difference between using the normalized or non-normalized projection matrix for ICA and LSI for the CSTR and 20NG-4 datasets. The only exception is the ICA vs ICA_Norm case for 20NG-4, where the p values suggest that ICA_Norm is slightly better than ICA. Based on the majority of cases, we conclude that normalization of the projection matrix for both LSI and ICA is not necessary. In the following comparison of the four DRTs, the non-normalized version of LSI and ICA will

be used.

Figure 1. Comparison of LSI vs LSI_normalized. In all the plots, the x-axis denotes dimensionality, and y-axis is the classification accuracy. '+' represents 'normalized' version of LSI, '.' represents 'non-normalized' version of LSI.

Figure 2. Comparison of ICA vs ICA_normalized. In all the plots, the x-axis denotes dimensionality, and y-axis is the classification accuracy. '+' represents 'normalized' version of ICA, '.' represents 'non-normalized' version of ICA.

Table 2. LSI/LSI_Norm and ICA/ICA_Norm Comparison for CSTR and 20NG-4

CSTR					20NG-4				
Dims	LSI	LSI_N	ICA	ICA_N	Dims	LSI	LSI_N	ICA	ICA_N
5	0.827	0.819	0.827	0.843	4	0.960	0.964	0.669	0.750
9	0.811	0.835	0.843	0.874	7	0.967	0.957	0.825	0.882
13	0.835	0.819	0.835	0.827	10	0.959	0.968	0.868	0.895
17	0.858	0.819	0.835	0.858	20	0.953	0.955	0.935	0.949
21	0.835	0.827	0.850	0.874	30	0.926	0.950	0.941	0.957
23	0.843	0.835	0.835	0.811	40	0.927	0.934	0.947	0.945
33	0.764	0.795	0.843	0.858	50	0.904	0.922	0.935	0.947
43	0.787	0.835	0.772	0.780	60	0.914	0.918	0.930	0.937
53	0.764	0.669	0.780	0.748	70	0.889	0.917	0.934	0.938
63	0.740	0.701	0.803	0.732	77	0.900	0.917	0.926	0.933
69	0.764	0.685	0.693	0.654	97	0.878	0.915	0.916	0.917
115	0.646	0.591	0.488	0.472	117	0.874	0.901	0.897	0.907
161	0.591	0.528	0.583	0.496	137	0.861	0.882	0.876	0.894
207	0.457	0.417	0.520	0.528	157	0.850	0.892	0.872	0.890
253	0.378	0.370	0.409	0.465	177	0.856	0.873	0.871	0.889
299	0.370	0.433	0.425	0.457	197	0.851	0.880	0.855	0.893
					217	0.840	0.880	0.851	0.873
					231	0.835	0.879	0.849	0.874
					385	0.859	0.874	0.821	0.839
					539	0.836	0.821	0.801	0.818
					693	0.810	0.819	0.785	0.788
					847	0.788	0.792	0.767	0.765
H_0: $\mu_{CA\ LSI[5,23]} = = \mu_{CA\ LSI\ Norm[5,23]}$					H_0: $\mu_{CA\ LSI[4,40]} = = \mu_{CA\ LSI\ Norm[4,40]}$				
p=0.16, H_0 hold, LSI = LSI_Norm					p=0.12, H_0 hold, LSI = LSI_Norm				
H_0: $\mu_{CA\ ICA[9,33]} = = \mu_{CA\ ICA\ Norm[9,33]}$					H_0: $\mu_{CA\ ICA[30,70]} = = \mu_{CA\ ICA\ Norm[30,70]}$				
p=0.14, H_0 hold, ICA = ICA_Norm					p=0.04, reject H0, ICA < ICA_Norm				

5.2 Comparisons of the four DRTs

For each dataset, we summarize the performances of the four DRTs in one Figure. The DRT comparisons are conducted by the combination of visual inspection and paired student t tests. To detect the "good range of reduced dimensions", we also plot the LSI performance against its singular values. Since ICA uses PCA as a preprocessing stage to "whiten" the raw data and determine the number of components (dimensions) to reduce

to, we are also interested in the correlation between ICA performance and eigenvalues used in the whitening step. This correlation may suggest how to determine the "good range of dimensions to reduced to" by ICA.

For each dataset, the classification accuracies of all the DRTs for the test data are reported in a separate table, including some detailed results of the paired student t tests. In Tables and Figures 3 through 7, the dimensions are ordered as follows: for DF, the dimensions are ordered according to the DF values, for LSI the dimensions are ordered based on the singular values (which indicate the importance of the dimensions), similarly, for ICA, the number of dimensions are determined by the PCA preprocessing step, in which the principle components are ordered based on their eigenvalues indicating their relative importance, and for RP, there is no ordering for the newly generated dimensions.

Results of Reuter-2.

From Figure 3 and Table 3, we observe the following. Within the whole range of dimensions being investigated, RP is inferior to DF. The performance of DF peaks around dimension of 657 with classification accuracy (CA) of 0.85 and then flattens and settles around 0.8 with increasing dimensionality. ICA and LSI achieve their best results with lower dimensionalities ([30,93]) that match with the best performance of DF. To compare the performance of ICA and LSI within their best common dimension range [30,93], the null hypothesis of the paired student t test assumes the means of CAs for ICA and LSI for the range [30,93] are equal. The result of the t-test rejects the null hypothesis, indicating superior performance of ICA over that of LSI. We also notice that ICA is more robust than LSI in that ICA maintains very good performance over a much larger range of dimensions than LSI.

The correlation between singular values and LSI performance (or eigenvalues and

ICA performance) is not clear. Thus, it is not possible to pinpoint the optimal dimensionality as a threshold as is done in many signal-processing applications [50]. We observe that, both the singular and eigen values decrease very rapidly within the first few to few tens of dimensions, after which there is general reduction. Hereafter, we refer to the part of singular/eigen value curve that transits from very rapid reduction to slow reduction as the transition zone. This transition zone seems to correspond to the best performance of LSI/ICA. Such an observation seems to be compatible with previous research results [3]. In all cases, it appears that over the transition zone, the CA curve of ICA reaches its peak and keeps at a constant level over a wider range of dimensions than that of LSI, indicating less feature sensitivity of ICA. Considering both the absolute best performance and robustness, for Reuter-2 dataset, we rank the DRTs in the order of ICA > LSI > DF> RP, where ">" denotes better.

Table 3. DRT Comparison for Reuter-2

Dims	ICA	LSI	DF	RP
10	0.840	0.774	0.695	0.438
20	0.859	0.828	0.678	0.496
30	0.854	0.852	0.671	0.534
40	0.859	0.852	0.697	0.551
50	0.857	0.858	0.720	0.581
60	0.857	0.859	0.749	0.615
70	0.859	0.853	0.794	0.605
73	0.859	0.855	0.794	0.626
93	0.859	0.848	0.800	0.670
113	0.854	0.830	0.804	0.680
133	0.849	0.820	0.808	0.711
153	0.854	0.823	0.805	0.729
173	0.857	0.818	0.802	0.732
193	0.848	0.803	0.827	0.741
219	0.852	0.791	0.821	0.779
365	0.826	0.777	0.831	0.798
511	0.786	0.757	0.843	0.809
657	0.739	0.739	0.853	0.805
H_0: $\mu_{CA\_ICA[30,93]} = = \mu_{CA\_LSI[30,93]}$, p=0.034, ICA > LSI				

Figure 3. DRT performance summary for Reuter-2.

a. parallel comparison of four DRTs, x-axis: dimensionality, same for the rest of the plots. y-axis: CAs for DRTs

b. comparisons between DF and RP with extended dimensionality

c. correlation of classification accuracy and normalized singular value for LSI, '+' denotes the CA curve and '.' denotes the normalized singular values

d. correlation of classification accuracy of ICA and the normalized eigenvalues of its PCA step, '+' denotes the CA curve and '.' denotes the normalized eigenvalues.

Results of Reuter-10

We have the following observations based on Figure 4 and Table 4. RP is inferior to DF for the whole dimension range investigated. DF reaches its peak performance at a dimension of 726 with a CA of 0.90, and then settles around 0.90 as the dimensionality increases. ICA provides good results within the range of [8,126], with the best results at dimension 20 among all the DRTs. LSI also provides reasonably good results in the range

of [8,126], but is inferior compared to ICA in terms of the best results and robustness based on paired t-tests. Similar to Reuter-2, the bst results of LSI and ICA seem to coincide with the transition zone of singular/eigen value curves. The relative ranking of the DRTs for Reuter-10 is in the order of ICA >LSI >DF >RP.

Figure 4. DRT performance summary for Reuter-10.

a. parallel comparison of four DRTs, x-axis: dimensionality, same for the rest of the plots. y-axis: CAs for DRTs
b. comparisons between DF and RP with extended dimensionality
c. correlation of classification accuracy and normalized singular value for LSI, '+' denotes the CA curve and '.' denotes the normalized singular values
d. correlation of classification accuracy of ICA and the normalized eigenvalues of its PCA step, '+' denotes the CA curve and '.' denotes the normalized eigenvalues.

Table 4. DRT Comparison for Reuter-10

Dims	ICA	LSI	DF	RP
3	0.794	0.734	0.667	0.490
5	0.857	0.777	0.816	0.495
8	0.908	0.814	0.784	0.523
10	0.911	0.843	0.781	0.488
20	0.926	0.906	0.754	0.551
30	0.919	0.893	0.761	0.547
40	0.890	0.906	0.782	0.606
50	0.883	0.906	0.789	0.623
60	0.881	0.889	0.821	0.645
66	0.886	0.889	0.825	0.662
86	0.881	0.876	0.845	0.706
106	0.874	0.848	0.860	0.734
126	0.857	0.829	0.857	0.747
146	0.866	0.816	0.873	0.765
166	0.868	0.829	0.877	0.793
186	0.852	0.813	0.884	0.797
198	0.858	0.799	0.882	0.811
330	0.809	0.733	0.895	0.854
462	0.769	0.715	0.896	0.878
594	0.741	0.608	0.901	0.847
726	0.703	0.629	0.905	0.885
H_0: $\mu_{CA\_ICA[8,126]} = = \mu_{CA\_LSI[8,126]}$, p=0.046, ICA >LSI				

Results for WebKB4

For WebKB4, Table 5 and Figure 5 show that RP is again inferior to DF for the range of dimensionality investigated. DF peaks at dimension 495 with a CA of 0.757 and then flattens out and settles with a CA around 0.70. LSI provides the best results at dimension 7 with CA of 0.81. ICA also provides reasonably good results in the range of [7,60]. Comparing the performance of LSI and ICA for the range [7,60], the null hypothesis assuming the means of ICA and LSI being equal is only weakly rejected with a p value of 0.051. Though only weakly inferior to LSI for the range of [7,60], ICA seems more stable with little variance. Again, we observe a coincidence between good performances of LSI/ICA and the transition zones of singular/eigen value curves. For

WebKB4, we can rank the DRTs in the order of LSI > ICA > DF > RP.

Figure 5. DRT performance summary for WebKB4

a. parallel comparison of four DRTs, x-axis: dimensionality, same for the rest of the plots. y-axis: CAs for DRTs

b. comparisons between DF and RP with extended dimensionality

c. correlation of classification accuracy and normalized singular value for LSI, '+' denotes the CA curve and '.' denotes the normalized singular values

d. correlation of classification accuracy of ICA and the normalized eigenvalues of its PCA step, '+' denotes the CA curve and '.' denotes the normalized eigenvalues.

Table 5. DRT Comparison for WebKB4

Dims	ICA	LSI	DF	RP
4	0.687	0.748	0.468	0.333
7	0.741	0.810	0.493	0.360
10	0.755	0.800	0.539	0.376
20	0.744	0.770	0.677	0.408
30	0.749	0.766	0.643	0.415
40	0.754	0.748	0.696	0.404
50	0.739	0.747	0.647	0.440
60	0.741	0.730	0.663	0.464
70	0.721	0.728	0.674	0.480
80	0.720	0.736	0.730	0.476
90	0.711	0.714	0.726	0.511
99	0.724	0.717	0.723	0.515
297	0.643	0.668	0.720	0.623
495	0.634	0.655	0.757	0.684
H_0: $\mu_{CA\ ICA[7,60]} = = \mu_{CA\ LSI[7,60]}$, p=0.051, ICA< LSI				

Table 6. DRT Comparison for CSTR

Dims	ICA	LSI	DF	RP
5	0.827	0.827	0.487	0.417
9	0.843	0.811	0.548	0.386
13	0.835	0.835	0.548	0.378
17	0.835	0.858	0.654	0.409
21	0.850	0.835	0.654	0.409
23	0.835	0.843	0.646	0.433
33	0.843	0.764	0.677	0.480
43	0.772	0.787	0.606	0.465
53	0.780	0.764	0.685	0.441
63	0.803	0.740	0.677	0.512
69	0.693	0.764	0.795	0.457
115	0.488	0.646	0.764	0.575
161	0.583	0.591	0.748	0.685
207	0.520	0.457	0.819	0.677
253	0.409	0.378	0.850	0.717
299	0.425	0.370	0.819	0.764
H_0: $\mu_{CA\ ICA[5,33]} = = \mu_{CA\ LSI[5,33]}$,p=0.165, ICA = LSI				

Results of CSTR

As shown in Table 6 and Figure 6, RP is inferior compared to DF. DF performance peaks at a dimension of 253 and then settles with a CA around 0.82. Both

ICA and LSI provide equally good results over a range of [5, 33] as indicated by the p value of paired t test in Table 6. The best results of LSI and ICA are better than that of DF, but are achieved with very low dimensionalities. Again, the good performances of LSI/ICA coincide with the transition zones of singular/eigen value curves in Figure 6. For CSTR, we rank the DRTs in the order of ICA = LSI >DF >RP.

Figure 6. DRT performance summary for CSTR.
a. parallel comparison of four DRTs, x-axis: dimensionality, same for the rest of the plots. y-axis: CAs for DRTs
b. comparisons between DF and RP with extended dimensionality
c. correlation of classification accuracy and normalized singular value for LSI, '+' denotes the CA curve and '.' denotes the normalized singular values
d. correlation of classification accuracy of ICA and the normalized eigenvalues of its PCA step, '+' denotes the CA curve and '.' denotes the normalized eigenvalues.

Results of 20NG-4

Based on Figure 7 and Table 7, RP is inferior to DF for the whole range of dimensionality investigated, only catching up with DF after dimension 1500. For this dataset, DF achieves the best result among all the DRTs at dimension 40, and then rapidly drops off and settles with CA around 0.79 with full dimension. Visually, ICA and LSI show indistinguishable performance in the range [20, 97] with the mean CA for both above 0.91. The paired t-test comparing the means of the CA for ICA and LSI clearly identifies the superiority of ICA over LSI in Table 7. Again, the best performances of LSI and ICA overlap closely with the transition zones of singular/eigen value curves. In summary, for 20NG-4, we can rank the four DRTs in the order of ICA > LSI >DF >RP considering both their best performances and robustness.

Table 7. DRT Comparison for 20NG-4

Dims	ICA	LSI	DF	RP
4	0.669	0.960	0.441	0.278
7	0.825	0.967	0.469	0.337
10	0.868	0.959	0.444	0.358
20	0.935	0.953	0.627	0.496
30	0.941	0.926	0.975	0.538
40	0.947	0.927	0.945	0.626
50	0.935	0.904	0.924	0.664
60	0.930	0.914	0.912	0.722
70	0.934	0.889	0.906	0.723
77	0.926	0.900	0.887	0.722
97	0.916	0.878	0.886	0.771
117	0.897	0.874	0.876	0.775
137	0.876	0.861	0.856	0.785
157	0.872	0.850	0.864	0.796
177	0.871	0.856	0.861	0.804
197	0.855	0.851	0.876	0.790
217	0.851	0.840	0.872	0.830
231	0.849	0.835	0.877	0.842
385	0.821	0.859	0.869	0.827
539	0.801	0.836	0.864	0.845
693	0.785	0.810	0.861	0.842
847	0.767	0.788	0.860	0.810
H_0: $\mu_{CA\_ICA[20,97]} = = \mu_{CA\_LSI[20,97]}$,p=0.008, ICA >LSI				

Figure 7. DRT performance summary for 20NG-4.

a. parallel comparison of four DRTs, x-axis: dimensionality, same for the rest of the plots. y-axis: CAs for DRTs

b. comparisons between DF and RP with extended dimensionality

c. correlation of classification accuracy and normalized singular value for LSI, '+' denotes the CA curve and '.' denotes the normalized singular values

d. correlation of classification accuracy of ICA and the normalized eigenvalues of its PCA step, '+' denotes the CA curve and '.' denotes the normalized eigenvalues.

5.3. Discussion

Taking the performances of the four DRTs over the 5 datasets together, we begin to see some general behavior patterns of the DRTs. In general, we can rank the DRTs in the following order: ICA > LSI > DF > RP.

In most instances, ICA and LSI can achieve the best or sub-optimal results with very low dimensionality, often less than 100 and occasionally lower than 10. ICA and LSI maintain their best performances over a range of 100 to 200 or even longer dimension ranges and then start to decrease as the dimensionality increases (this is more obvious from the tables than from the graphs). Presumably, the point at which performance starts to decrease is the point as which ICA and LSI have derived the maximum necessary features from those datasets. The stability of ICA/LSI suggests that during the process of dimension reduction, the discovered latent variables by both methods seem to have a clear structure in terms of noise content. This may explain why ICA/LSI show very good performance across a wide range of consecutive dimensions (the components/dimensions being used are "noise free") and a rapid performance drop after the number of dimensions increases above certain values (when we begin to include more "noisy/trivial components" into the clustering procedure).

ICA often shows additional stability when compared to LSI, which suggests that the ICA discovers more "non-trivial" or "noise free" latent variables than LSI does. This may be rooted in the fact that the latent variables derived from ICA are statistically more independent from each other than those derived from LSI. More "non-trivial" or "noise free" latent variables implies a better description of the text data. One of the well-known drawbacks of methods like LSI and PCA is the lack of interpretability, i.e., the latent variables derived from LSI /PCA are often very difficult for human users to understand. Therefore, ICA may provide a good alternative for the automatic generation of human-understandable latent variables from the text data, which is a very interesting future research direction.

We also observe a strong correlation between the good performance of ICA/LSI

and the transition zone of the corresponding eigen/singular value curve. In the future, we can use this as a heuristic rule to choose the dimensions that we should include for clustering when using ICA/LSI for dimension reduction. But, only from our experiments, it is still not clear how to decide how far the transition zone should extend or, in another words, the maximum number of dimensions to be used without degrading cluster quality. This is an open research question and worth pursuing.

The performance of DF often peaks at some middle range dimensions (much higher than that of ICA/LSI's best dimensions), and then settles down as the number of dimensions increases. It is also interesting to notice that the best performance of DF at relatively high dimensions often matches up with the best performances of ICA/LSI at much lower dimensions. Such behavior suggests that the full dimension representation is not needed for text clustering and that the majority of the dimensions are very noisy. It is not clear, yet, that we can use DF to pre-select some dimensions to be used for ICA/LSI instead of using the full set of dimensions, which is much more expensive for computing ICA/LSI. This is another research direction worth investigating.

Obviously, the performance of RP is a disappointment. Based on the discussion above, we can understand why RP fails. The goal of RP is to provide a projection for the data from the original high dimensional space onto a lower dimensional space while maintaining a good approximation of the mutual distances among data points, i.e., the distance distortion error is well bounded. All the problems with RP come from the fact that most of the dimensions in text data are very noisy, not reliable and meaningful, which makes the distance profile based on the full dimensional space very noisy and not meaningful. Therefore, it is not appropriate for RP to try to approximate and maintain such distance profiles without any noise reduction as in ICA/LSI. Even worse, when RP

projects the original high dimensional space onto a lower dimensional space, there is some chance that RP will create even noisier representations than the original dataset since RP linearly combines the original dimensions into new dimensions with a random procedure. This may explain why RP performs systematically worse than DF at relatively low dimensions, and only slowly catches up with DF when the dimensionality is increased to a significant level. Our experiments on RP seem to confirm the discoveries of other researchers [18, 34].

6 Conclusion and future work

In this research, we compared four well-known dimension reduction techniques, DF, RP, LSI and ICA, for the document clustering task. To judge their relative effectiveness and robustness, we applied all four of them to five benchmark datasets of different characteristics. Over all the datasets, we identified some general behaviors of these techniques. In general, we can rank the four DRTs in the order of ICA >LSI >DF >RP. ICA demonstrates good performance and superior stability compared to LSI. Both ICA and LSI can effectively reduce the dimensionality from a few thousands to the range of 100 to 200 or even less. The best performances of ICA/LSI seem correspond well with the transition zone of the eigen/singular value curve. The experiments with DF clearly indicate to us that most of the raw dimensions in the text data are very noisy and meaningless with respect to the document clustering task, which further explains the relatively poor performance of RP.

Since we have identified ICA and LSI as good candidates for dimension reduction for text clustering, future research will be focused mainly on different aspects of these

two methods. First, we would like to investigate the semantic meanings of the latent variables derived from ICA and LSI, and evaluate their quality difference using human judgment for their interpretability. Secondly, we want to investigate the possibility of using DF to pre-screen the raw dimensions as a pre-processing step for LSI/ICA to further reduce the computational cost of LSI/ICA. Finally, to further reduce the computational cost of ICA/LSI, we may want to investigate proper sampling techniques to select the "representative" documents to which ICA/LSI will be applied.

References

[1] D. Achlioptas, “Database-friendly random projections,” *Proc. PODS'01*, pp. 274-281, 2001.

[2] C.C. Aggarwal, "An Efficient Subspace Sampling Framework for High-Dimensional Data Reduction, Selectivity Estimation, and Nearest-Neighbor Search," *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1247-1262, 2004.

[3] M.W. Berry, S.T. Dumais, and G.W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Review,* vol. 37(4), pp. 573-595, 1995.

[4] K. Beyer, J. Goldstein., R. Ramakrishnan., & U. Shaft, "When is the Nearest Neighbour Meaningful?" *Proceedings of the 7th International Conference on Database Theory,* pp.217-235, 1999.

[5] J.C. Bezdek, *Pattern recognition with fuzzy objective function algorithms,* New York: Plenum, 1981.

[6] E. Bingham and H. Mannila, “Random projection in dimensionality reduction: applications to image and text data," *Proc. SIGKDD*, pp. 245-250, 2001.

[7] E. Bingham, A. Kabán, and M. Girolami, "Topic identification in dynamical text by complexity pursuit", *Neural Processing Letters*, vol. 17(1), pp. 69-83, 2003.

[8] H.H. Bock, "Probabilistic aspects in clustering analysis," in O.Opitz (Ed.), *Conceptual and numerical analysis of data*, pp. 12-44, Berlin: Springer-verlag, 1989.

[9] G. A. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565-588, 1991.

[10] S. Dasgupta, "Experiments with random projection," *Proc. UAI,* 2000.

[11] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, & R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science,* vol. 41(6), pp. 391-407, 1990.

[12] R. N. Devé and R. Krishnapuram, "Robust Clustering Methods: A United View." *IEEE Transactions on Fuzzy Systems,* vol. 5, no. 2, pp. 270-293, 1997.

[13] I. S. Dhillion, S. Mallela, & S.S Modha, "Information-theoretic co-clustering," *Proc. SIGKDD*, pp. 89-98, 2003.

[14] C.H. Ding, "A Similarity-based Probability Model for Latent Semantic Indexing," *Proc. SIGIR*, pp. 58-65, 1999.

[15] C. H. Ding, "A Probabilistic Model for Dimensionality Reduction in Information Retrieval and Filtering," *Proc. of 1st SIAM Computational Information Retrieval Workshop*, 2000.

[16] M. Efron, "Amended Parallel Analysis for Optimal Dimensionality Reduction in Latent Semantic Indexing," SILS Technical Report TR-2002-03, http://ils.unc.edu/ils/research/reports/TR-2002-03.pdf.

[17] I.K. Fodor, "A survey of dimension reduction techniques," LLNL technical report, UCRL-ID-148494, 2002, http://www.llnl.gov/CASC/sapphire/pubs.html.

[18] D. Fradkin, & D. Madigan, “Experiments with Random Projection for Machine Learning*,” Proc. SIGKDD,* pp. 517-522, 2003.

[19] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Proceesing System* 7, G. Tesauro, D. S. Touretzky, and T. K. Leen Eds. Cambridge, MA: MITPress, pp. 625-632, 1995.

[20] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases,." *Proc. ACM SIGMOD,* pp. 73-84, 1998.

[21] G. H. Golub and C. F. van Loan, *Matrix Computations*, North Oxford Academic, Oxford, UK, 1983.

[22] R. Hecht-Nielsen, “Context vectors: general purpose approximate meaning representations self-organized from raw data,” Computational *Intelligence: Imitating Life,* J.M. Zurada, R.J. Marks II, and C.J. Robinson, eds, IEEE Press, Piscataway, NJ, pp. 43-56, 1994.

[23] T. Hofmann, "Probabilistic Latent Semantic Indexing," *Proc. SIGIR,* pp. 50-57, 1999.

[24] P. Husbands, H. Simon and C.Ding, "On the Use of Singular Value Decomposition for Text Retrieval," *Proc. SIAM Comp. Info. Retrieval Workshop*, 2000.

[25] A. Hyvärinen and E. Oja. “A fast fixed-point algorithm for independent component analysis,” Neural *Computation*, vol, 9, pp. 1483-1492, 1997.

[26] A. Hyvärinen, “Fast and Robust Fixed-Point Algorithms for Independent Component Analysis,” IEEE. *Trans. Neural Networks,* vol. 10 (3), pp. 626-634,

1999.

[27] A. Hyvärinen and E. Oja, ”Independent Component Analysis: Algorithms and Applications,” Neural *Networks*, vol. 13 (4-5), pp. 411-430, 2000. FastICA package: http://www.cis.hut.fi/~aapo/.

[28] P. Indyk and R. Motwani, “Approximate nearest neighbors: Towards removing the curse of dimensionality, “ Proc. *of 30th STOC*, pp. 604-613, 1998.

[29] S. Kaski, “Dimensionality reduction by random mapping,” *Proc. Int. Joint Conf. on Neural Networks,* vol. 1, pp. 413-418, 1998.

[30] Y. -H. Kim, and B. -T.Zhang, “Document Indexing Using Independent Topic Extraction, “ *Proceedings of the Third International Conference on Independent Component Analysis and Signal Separation*, pp. 557-562, 2001.

[31] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

[32] T. Kolenda, L. K. Hansen, S. Sigurdsson, *”*Indepedent Components in Text*,” Advances in Independent Component Analysis,* pp. 229-250, Springer-Verlag, 2000

[33] R. Krishnapuram and J. M. Keller, " A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.,* vol. 1, pp. 98-110, 1993.

[34] J. Lin and D. Gunopulos, “Dimensionality Reduction by Random Projection and Latent Semantic Indexing,” *Proc. SDM'03 Conf., Text Mining Workshop*, 2003.

[35] K.-R. Müller, P. Philips, and A. Ziehe, "JADETD : Combining higher-order statistics and temporal information for blind source separation (with noise*)*," *Proc. ICA'99*, pp. 87-92, 1999.

[36] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent Semantic

Indexing: A Probabilistic Analysis," *Proc. ACM SIGPODS*, pp. 159-168,1998.

[37] L. Parsons, E. Hague, H. Liu, "Subspace clustering for high dimensional data: a review", *ACM SIGKDD Explorations Newsletter*, *Special issue on learning from imbalanced datasets,* vol. 6 (1), pp. 90 - 105, 2004.

[38] L.K. Saul, and S.T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," *J. Machine Learning Research*, vol. 4, pp. 119-155, 2003.

[39] B. Tang, M.I. Heywood, and M. Shepherd, "The self-organization by lateral inhibition model: validation of clustering," *Proc. International Joint Conference on Neural Networks*, vol. 1, pp. 781-786, 2003.

[40] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," *Proc. of the 37th annual Allerton Conference on Communication, Control, and Computing,* pp. 368-377, 1999.

[41] W. Xu and Y. Gong, "Document clustering by concept factorization," *Proc. ACM SIGIR,* 2004.

[42] Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proc. ICML*, pp. 412-420, 1997.

[43] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proc. SIGMOD Conf*., pp. 103-114, 1996.

[44] Y Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," TR #01--40, Department of Computer Science, University of Minnesota, 2001, http://cs.umn.edu/karypis/publications.

[45] http://www2.cs.cum.edu/afs/cs/project/theo-11/www/wwkb

[46] http://www.cs.cmu.edu/TextLearning/datasets.html

[47] http://www.cs.rochester.edu/trs

[48] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

[49] http://www.tartarus.org/~martin/PorterStemmer/

[50] http://www.clecom.co.uk/science/autosignal/help/Signal_Threshold_Selection.htm