Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering

Bin Tang[*], Michael Shepherd, Evangelos Milios, Malcolm I. Heywood
{btang, shepherd, eem, mheywood}@cs.dal.ca
Faculty of Computer Science, Dalhousie University, Halifax, Canada, B3H 1W5

Abstract

A great challenge of text mining arises from the increasingly large text datasets and the high dimensionality associated with natural language. In this research, a systematic study is conducted of six Dimension Reduction Techniques (DRT) in the context of the text clustering problem using three standard benchmark datasets. The methods considered include three feature transformation techiques, Independent Component Analysis (ICA), Latent Semantic Indexing (LSI), Random Projection (RP), and three feature selection techniques based on Document Frequency (DF), mean TfIdf (TI) and Term Frequency Variance (TfV). Experiments with the k-means clustering algorithm show that ICA and LSI are clearly superior to RP on all three datasets. Furthermore,it is shown that TI and TfV outperform DF for text clustering. Finally, experiments where a selection technique is followed by a transformation technique show that this combination can help substantially reduce the computational cost associated with the best transformation methods (ICA and LSI) while preserving clustering performance.

Keywords: dimension reduction techniques, ICA, LSI, term frequency variance, mean TfIdf

1 Introduction

Document clustering is the fundamental enabling tool for efficient document organization, summarization, navigation and retrieval for very large datasets. The most critical problem for text clustering is the high dimensionality of the natural language text. The focus of this research is to investigate the relative effectiveness of various dimension reduction techniques for text clustering.

There are two major types of DRTs, feature transformation and feature selection [17]. 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, Principal Components Analysis (PCA), Factor Analysis, Projection Pursuit, Latent Semantic Indexing (LSI), Independent Component Analysis (ICA), and Random Projection (RP) [8] . Feature selection methods only select a subset of "meaningful or useful" dimensions (specific for the application) from the original set of dimensions. For text applications, some feature selection methods for text applications include, Document Frequency (DF), mean TFIDF (TI), Term Frequency Variance (TfV).

Although many research projects are actively engaged in furthering DRTs as a whole, so far, there is a lack of experimental work comparing them in a systematic manner especially for text clustering task. In our previous work [18] , we compared four of the abovementioned methods (including ICA, LSI, RP, DF) on five benchmark datasets. Considering both the effectiveness and robustness of all the methods, 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. Though providing superior performance, the computation cost of ICA is much higher compared to DF. In [18] , we pointed out the need to find proper feature selection methods to pre-screen dimensions before the ICA computation to reduce the computational cost of ICA without sacrificing performance.

In this work, we investigate the relative effectiveness and robustness of six dimension reduction techniques when used for text clustering using three benchmark datasets. The used DRTs are Document Frequency (DF), mean TFIDF (TI), Term Frequency Variance (TfV), Latent Semantic Indexing (LSI), Random Projection (RP) and Independent Component Analysis (ICA). We also demonstrate the effectiveness of combining TI or TfV with ICA as a computationally cheaper alternative to the default ICA with full dimensions.

This paper is organized as follows. Section 2 provides more details for the DRTs used in this research. Section 3 describes our experimental procedure, evaluation methods and dataset issues. Section 4 presents our experimental results and appropriate discussion notes. Finally, conclusions are drawn and future research di-

[*] corresponding author

rections identified in Section 5.

2 Dimension Reduction Techniques for Text Clustering

In the rest of the discussion, we will use the following notations. A document collection is represented by its term-document matrix $X$of m by n, with m terms and n documents.

2.1 Feature Selection Methods Feature Selection methods sort terms on the basis of a numerical measure computed from the document collection to be clustered, and select a subset of the terms by thresholding that measure. In this section, we will describe the mathematic details of three feature selection methods, including Document Frequency (DF) in Section 2.1.1, Mean TFIDF (TI) in Section 2.1.2 and Term Frequency Variance (TfV) in Section 2.1.3.

2.1.1 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. DF can be formally defined as follows. For a document collection X of m terms by 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 X from m to k ($k < m$), we choose to use the k dimensions (terms) with the top k DF values. It is obvious that the DF takes $O(mn)$ to evaluate. In spite of its simplicity, it has been demonstrated to be as effective as more advanced techniques in text categorization [19].

2.1.2 Mean TFIDF (TI) In information retrieval (IR), we value a term with high term frequency but low document frequency as a good indexing term. In IR, we generate a vector representation for each document d_j, where the weight for each term t in document d_j is its $tfidf$ value, defined as:

$$tfidf_j = tf_j \log \frac{|T_r|}{DF_t}$$

where

$$tf_j = \begin{cases} 1 + \log t_j & \text{if } t_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

and T_r is the total number of documents in collection X, DF_t is the document frequency of term t, t_j is the frequency of term t in document d_j. In this work, we propose to use the mean value of $tfidf$ over all the documents (hereafter referred to as TI) for each term as a measure of the quality of the term. The higher the TI value, the better the term to be ranked.

2.1.3 Term Frequency Variance (TfV) The TfV method for ranking term quality was demonstrated to successfully reduce the dimension to only 15% of the original dimension [6, 13]. The basic idea is to rank the quality of a term based on the variance of its term frequency. This is similar in spirit to the intuition of TI method. The term frequency of term t in document d_j, tf_j, is defined the same way as in Section 2.1.2. The quality of term t is calculated by

$$\sum_j^n tf_j^2 - \frac{1}{n}\left[\sum_j^n tf_j\right]^2$$

where n is the total number of documents.

2.2 Feature Transformation Methods Feature transformation methods perform a transformation of the vector space representation of the document collection into a lower dimensional subspace, where the new dimensions can be viewed as linear combinations of the original dimensions. In this section, we will introduce some mathematic details of the three feature transformation methods, i.e., Latent Semantic Indexing (LSI) in Section 2.2.1, Random Projection (RP) in Section 2.2.2 and Independent Component Analysis (ICA) in Section 2.2.3.

2.2.1 Latent Semantic Indexing (LSI) LSI, as one of the standard dimension reduction techniques in information retrieval, has enjoyed long-lasting attention [2, 5, 7, 10, 15, 16]. By detecting the high-order semantic structure (term-document relationship), it aims to address the ambiguity problem of natural language, i.e., the use of synonymous, and polysemous words, therefore, a potentially excellent tool for automatic indexing and retrieval.

LSI 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). During the SVD process, the newly generated dimensions are orderd by their "importance". Using the full rank SVD, the term-document matrix X is decomposed as $X = USV^T$, where S is the diagonal matrix containing singular values of X. U and V are orthogonal matrices containing left and right singular values of X, often referred to as term projection matrix and document projection matrix respectively. Using truncated SVD, the best rank-k approximation (in least-squares sense) of X is $X_k \cong U_k S_k V_k^T$, in which X is projected from m dimensional space to k dimensional space ($m > k$). In the new k-dimension, each original document d can be re-represented as $\tilde{d} =$

$U_k S_k d^T$. The truncated SVD not only captures the most important associations between terms and documents, but also effectively removes noise and redundancy and word ambiguity within the dataset [5]. One major drawback of LSI is its high computational cost. For a data matrix, X, of dimension $m \times n$, the time complexity to compute LSI using the most commonly used *svd* packages is in the order of $O(m^2n)$ [15]. For a sparse matrix, the computation can be reduced to the order of $O(cmn)$, where c is the average number of terms in each document [16].

2.2.2 Random Projection (*RP*) As a computationally cheaper alternative to LSI for dimension reduction with bounded distance distortion error, the method of Random Projection (RP) has recently received attention from the machine learning and information retrieval communities [1, 4, 9, 12, 15]. Unlike LSI, the new dimensions in RP are generated randomly (random linear combinations of original terms) with no ordering of "importance". The new dimensions are only approximately orthogonal. However, researchers don't seem to agree on the effectiveness and computational efficiency of RP as a good alternative for LSI-like techniques [4, 9, 12, 15]. So far, the effectiveness of RP is still not clear, especially in the context of text clustering.

Similar to LSI, RP projects the columns of term-document matrix X from the original high dimensional space (with m dimensions) onto a lower k-dimensional space using a randomly generated projection matrix R_k of shape $k \times m$, where the columns of R are unit length vectors following a Gaussian distribution. Under the new k dimension space, X is approximated as $X_k \cong R_k X$.

2.2.3 Independent component analysis (*ICA*) A recent method of feature transformation called Independent Component Analysis (*ICA*) has gained widespread attention in signal processing [11]. 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, 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 [11]. In most of applications of ICA, PCA is used as a preprocessing step, in which the newly generated dimensions are ordered by their importance. Based on the PCA transformed data matrix, ICA further transform the data into independent components. Therefore, using PCA as a preprocessing step, ICA can be used as a dimension reduction technique. Until very recently, there were only a few experimental works in which ICA is applied to text data [3, 14].

ICA assumes each observed data item (a document) x to have been generated by a mixing process of statistically independent components (latent variables s_i). Formally, for the term-document matrix $X_{m \times n}$, the noise-free mixing model can be written as $X_{m \times n} = A_{m \times k} S_{k \times n}$, where A is referred to as the mixing matrix and $S_{k \times n}$ is the matrix of independent components. The inverse of A, A^{-1}, is referred as the unmixing matrix, W. The independent components can be expressed as $S_{k \times n} = W_{k \times m} X_{m \times n}$. Here, W is functionally similar to the projection matrix R in RP that project X from the m dimensional space to a lower k dimensional space.

In this research, we used the most commonly used FastICA implementation [11]. FastICA is known to be robust and efficient in detecting the underlying independent components in the data for a wide range of underlying distributions [8]. The mathematical details of FastICA can be found in [11].

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

3 Evaluation

In this section, we present the evaluation methods and experimental setups in Section 3.1, followed by the description of the datasets used in Section 3.2, and ended with the description of the preprocessing procedure in Section 3.3.

3.1 Evaluation Methods and Experimental Setup The judgment of the relative effectiveness of the DRTs for text clustering is based on the final clustering results after different DRTs are applied. The final ranking of DRTs depends on both the absolute clustering results and the robustness of the DRT. Here, good robustness implies that when using a certain DRT, reasonably good clustering results remain relatively stable across a relatively wide range of reduced dimensions.

The quality of text clustering is measured by micro-average of *classification accuracy* (hereafter referred to as *CA*) over all the clusters, a similar measure to *Purity* as introduced in [20]. To avoid the bias from the training set, *CA* is only computed based on the test data in the following fashion. The clustering process is only based on the training set. After clustering, each cluster

i is assigned a class label T_i based on the majority vote from its members' classes using only training data. Then, assign each point in test set to its closest cluster. The CA_i for 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. The total CA is micro-averaged over all the clusters. The comparison between two methods is usually based on student t-test.

Since k-means or its variants are the most commonly used clustering algorithms used in text clustering, we choose to use k-means with our modification to do text clustering. A well-known problem for k-means is that poor choices of initialization often lead to poor convergence to sub optimal solutions. To ameliorate the negative impact of poor initialization, we devised a simple procedure, $InitKMeans$, to pre-select "good" seeds for k-means clustering. It has been proved very effective in our previous work [18]. Our experiments for all the DRTs follow the same general procedure. A sketch of our procedure is as follows, details of our experimental procedure including $InitKMeans$ can be found elsewhere [18].

```
1.  Each dataset is split randomly into training and
    testing set of ratio 3:1 proportionally to their
    category distribution.
2.  For each DRT, run a series of reduced dimensions
    For each desired dimension $k$,
        Apply DRT only to the training data,
        producing proper projection matrix $PR$
        (in feature transformation), or, subset of
        selected dimensions $SD$ (feature selection);
        Apply $PR/SD$ to both training and test set;
        Clustering on the reduced training set;
        Assign $T_i$ to each cluster in reduced
        training set;
        Compute $CA$ using reduced test set;
    End For
```

3.2 Dataset Characteristics In our experiments, we used a variety of datasets from different genres, which include WWW-pages (WebKB[1]), newswire stories (Reuters-21578[2]), and technical reports (CSTR[3]). These 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.

Reuters-2, a subset of Reuters-21578 dataset, is a collection of documents each document with a single topic label. The version of Reuters2 that we used eliminates categories with less than 4 documents, leaving only 50 categories. WebKB4 is a subset of WebKB dataset, which is limited to the four most common categories: student, faculty, course, and project. The CSTR dataset contains 505 abstracts of technical reports, divided into four research areas: AI, Robotics and Vision, Systems, and Theory.

3.3 Preprocessing The pre-processing of the datasets follows the standard procedures, including removal of the tags and non-textual data, stop word removal[4], and stemming[5]. Then we further remove the words with low document frequency. For example, for the Reuter2 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 in Section 2.1.2.

4 Experimental Results

For each given dataset, we applied six DRTs for a complete comparative study. First, we concentrate on comparing the feature selection methods. The results are described in detail in Section 4.1. The comparison results of feature transformation methods are mainly extracted from our previous work [18], which will be summarized in Section 4.2. Based on the results from both DRT method groups, we choose to use TI and TfV as thresholding methods to pre-select subset of dimensions to be further processed by ICA. We focus on comparing the results of ICA with TI/TfV thresholding at different threshold levels against the default version of ICA without TI/TfV thresholding. Here, the threshold levels are defined as the top $x\%$ of selected dimensions using TI or TfV. In this set of experiments, we use TI (or TfV) to pre-select the top $x\%$ of dimensions and pass on the dataset with reduced dimensions to the ICA computation. The results are described in detail in Section 4.2. For completeness, we compile all the comparison results in one figure 1-3 for each dataset. In each figure, there are four sub-figures, describing the results of feature transformation methods, results of feature selection methods, results of ICA with TI thresholds, and results of ICA with TfV thresholds respectively.

The comparison of any two methods is based on Student paired t-test comparing the performance of the

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

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

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

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

[5] http://www.tartarus.org/∼martin/PorterStemmer/

Datasets	Dataset size $\lvert terms \rvert \times \lvert docs \rvert$	#classes	Class Size range	Type
Reuters 2	7315 x 8771	50	[4, 3807]	News
WebKB4	9870 x 4199	4	[504, 1641]	University Web pages
CSTR	2335 x 505	4	[76, 191]	Technical Reports

Table 1: Summary of the datasets

two methods over a dimension range. The dimension range, denoted as $[k1, k2]$, is usually hand-picked, such that, within such a range, the two methods cannot be clearly differentiated visually, and beyond this range, the performance of the two comparing methods are too poor to be of interest.

4.1 Comparing Feature Selection Methods We performed mutual comparison among DF, TI and TfV for all the three datasets using paired Student t-test. The p values are reported in In Table 2. For the paired Student t-test, the null hypothesis, H_0, assumes $\mu_{X-Y} = 0$. Here X represents the methods listed in rows in Table 2, while Y represent methods listed in columns in Table 2. The alternative hypothesis, H_a, assumes $\mu_{X-Y} > 0$. For Reuters2, the comparisons are performed over the the dimension range of $[70, 1095]$. Based on the p values of the paired t-test, the null hypothesis, $\mu_{DF-TI} = 0$ is weakly rejected, and $\mu_{DF-TfV} = 0$ is strongly rejected and $\mu_{TI-TfV} = 0$ holds. Therefore, for Reuters2, we can say that DF systematically performs worse than TI and TfV, and there is no statistical difference between TI and TfV. For WebKB4, the comparisons are performed over the dimension range of $[80, 1980]$. The resulting p values indicate that there is no significant different among DF, TI and TfV, even though TI and TfV provide better CA results than DF. For CSTR, the comparisons are performed over the range of dimensions $[115, 989]$. The resulting p values indicate that there is no significant difference between DF and TfV and between TI and TfV, while DF is worse than TI with slight significance.

Considering all the comparison results, TI and TfV are better feature selection methods than DF for text applications. Therefore, we choose to use TI and TfV as pre-screening methods for ICA in subsequent experiments.

4.2 Results of Feature Transformation Methods and Thresholded ICA In the following, we will describe the results by the order of dataset. For each dataset, we will remark on the comparison results for feature transformation methods based on our previous work [18] for completeness. We will focus on the comparisons between the performance of default ICA and ICA preceded by TI/TfV thresholding. The comparison results are reported based on the p values in separate Tables 3,4,5.

Reuters2 Results Based on the results of our previous work [18], comparing ICA, LSI and RP, we observed that both ICA and LSI achieve superior results with low dimensionalities ([30,93]) comparing to RP. Within the dimension range of [30,93], ICA not only shows a superior performance over LSI in terms of classification accuracy but also demonstrates better stability than LSI.

The results of comparing the plain ICA (with no pre-selection of dimensions) with that of ICA with pre-selection of dimensions by TI/TfV are reported in Table 3. The null hypothesis, H_0, assumes $\mu_{X-Y} = 0$. Here, X refers to plain ICA, while Y represents ICA with different TI/TfV thresholding levels. The alternative hypothesis, H_a, assumes $\mu_{X-Y} > 0$. Another alternative hypothesis, H_b, assumes $\mu_{X-Y} < 0$. [6] The comparisons are performed over that dimension range of $[10, 153]$. In Table 3, the p values clearly indicate that the plain ICA performs significantly better than ICA with TI-thresholding levels of 5-15%. But there are no significant differences between the plain ICA and ICA with TI-thresholding levels of 20-25% . Similarly, the plain ICA performs significantly better than ICA with TfV-thresholding levels of 5-20%. Interestingly, p value indicates that the ICA with TfV-thresholding level of 25% performs significantly better than the basic ICA.

WebKB4 Results Based on our previous work, we observe that the best performance of ICA is slightly worse than that of LSI [18]. But ICA shows much stable performance over longer range of dimensions than LSI. Both LSI and ICA are better than RP.

In Table 4, we reported the results of combining ICA with TI/TfV thresholding. The comparisons between the plain ICA and those ICAs with TI/TfV thresholding are performed over the range of $[7, 90]$. The p values indicate clearly that the plain ICA is significantly better than ICAs with TI-thresholding levels of 5% and 20%. But there is no significant

[6] We used the same hypothesis tests for Table 4, 5,therefore, not stated explicitly later.

Reuters2				WebKB4				CSTR			
	DF	TI	TfV		DF	TI	TfV		DF	TI	TfV
DF	N/A	0.07	0.01	DF	N/A	0.16	0.16	DF	N/A	0.04	0.13
TI	0.93	N/A	0.32	TI	0.84	N/A	N/A	TI	0.96	N/A	0.31
TfV	0.99	0.68	N/A	TfV	0.84	N/A	N/A	TfV	0.87	0.69	N/A

Table 2: P Values of Student Paired t-test for Comparing Feature Selection Methods

.

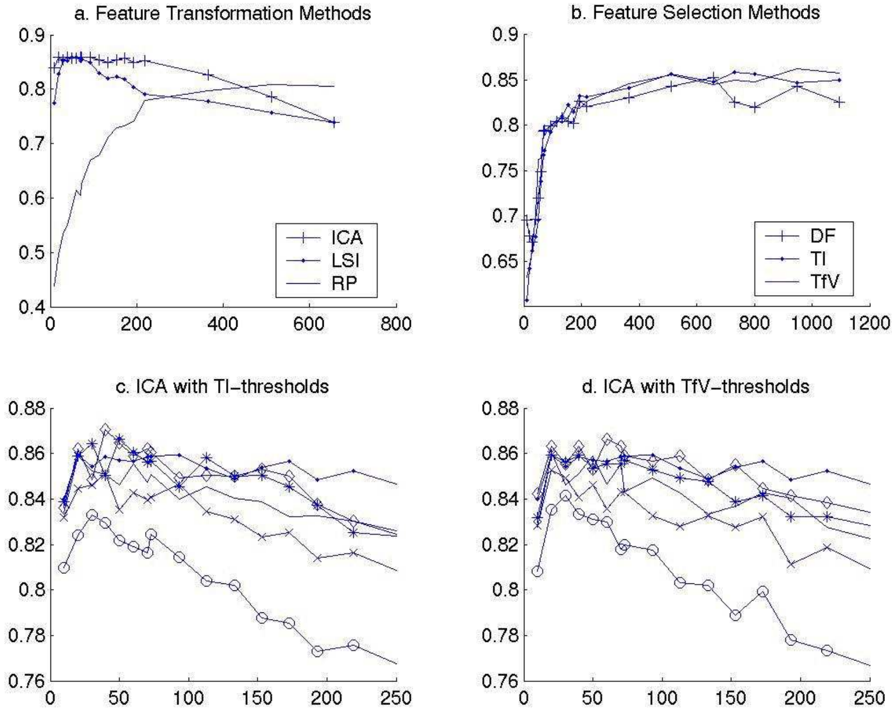

Figure 1: Comparison results of Reuters2. In all the sub-figures, the x-axis denotes the dimensionality, and the y-axis represents CA. (a) results of feature transformation method. '+' denotes ICA, '.' denotes LSI, '-' denotes RP. (b) results of feature selection methods. '+' denotes DF, '.' denotes TI, '-' denotes TfV. (c) results of ICA with different level of TI thresholding. 'o' denote thresholding level 5%, 'x' 10%, '-' 15%, '*' 20%,'⋄' 25%, and with '.' for plain ICA with full dimensions. (d) results of ICA with different levels of TfV thresholding, 'o' denotes thresholding level 5%, 'x' 10%, '-' 15%, '*' 20%, '⋄' 25%, and with '.' for basic ICA

		ICA with TI thresholding					ICA with TfV thresholding				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
H_a	p-value	0.00	0.00	0.00	0.44	0.63	0.00	0.00	0.00	0.01	0.96
H_b	p-value	1.00	1.00	1.00	0.56	0.37	1.00	1.00	1.00	0.99	0.04

Table 3: P-values of the results of ICA combined with TI/TfV thresholding (Reuters2)

difference between the plain ICA and ICAs with TI-thresholding level of 10%, 15% and 25%. For TfV thresholding, the plain ICA is better than ICA with TfV thresholding level of 5 % with significance and better than 10% with slight significance. But there is no significant difference between the plain ICA and ICAs with TfV-thresholding level of 15-25%.

CSTR Results From our previous work, we observed no significant difference between ICA and LSI for the dimension range of $[5, 33]$. ICA and LSI are better than RP [18].

The results of combining ICA with TI/TfV thresholding are reported in Table 5. We compared the performance of the plain ICA with those of ICAs with TI/TfV thresholdings over the dimension range of $[5, 43]$. Based on the p values, we conclude that the plain ICA is significantly better than ICAs with TI thresholding levels of 5-15% , and there is no significant difference between the plain ICA and ICAs with TI thresholding levels of 20-25%. For TfV thresholding, the plain ICA is better than ICAs with TfV thresholding levels of 5-15%, and there is no significant difference between the plain ICA and ICAs with TfV thresholding levels of 20-25% .

5 Conclusion and Future Work

In this research, we compared the performance of six DRT methods when applied to text clustering problem using three benchmark datasets of distinct genres. Based on all the results, we have observed the following. For feature transformation methods, we can rank ICA > LSI > RP considering classification accuracy and stability. Both ICA and LSI reach their best performance with very low dimensionality, often less than 100 and occasionally lower than 10. ICA and LSI maintain their best performances over a wide range dimensions. ICA appears more stable than LSI. For feature selection methods, DF is inferior comparing to TI and TfV. The best results of TI and TfV can match those of ICA and LSI but at much higher dimensions. The results of combining ICA with TI or TfV thresholding are most interesting. For most of the cases, it is safe to say that ICA with TI or TfV thresholding level 25% performs at least the same as the basic ICA if not better occasionally. This is interesting, since the bottleneck of computing ICA is its preprocessing PCA step (takes $O(m^2n)$ to compute, where m is the dimensionality, and n is the number of points). With our datasets, m and n are of the same magnitude, then the PCA step can be estimated as $O(m^3)$. With pre-screening the dimensions by TI or TfV methods, theoretically, we reduce the computational cost of PCA to 1/64 of the original cost without sacrificing performance.

From our previous and current research, we identify the "ideal" dimension reduction technique for text clustering to be ICA. Though we have achieved moderate success in reducing the computational cost of ICA, we believe that further research should be focused on this issue. Different sampling techniques should be able to provide even more fruitful success in reducing the computational cost of ICA without sacrificing its performance.

References

[1] D. Achlioptas. Database-friendly random projections. In *Proceedings of PODS*, pages 274–281, 2001.

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

[3] E. Bingham, A. Kabán, and M. Girolami. Topic identification in dynamical text by complexity pursuit. *Neural Processing Letters*, 17(1):69–83, 2003.

[4] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proc. SIGKDD*, pages 245–250, 2001.

[5] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[6] I. S. Dhillon, J. Kogan, , and M. Nicholas. Feature selection and document clustering. In M.W. Berry, editor, *A Comprehensive Survey of Text mining.* Springer-Verlag, 2003.

[7] C. H. Ding. A probabilistic model for dimensionality reduction in information retrieval and filtering. In *Proc. of 1st SIAM Computational Information Retrieval Workshop*, 2000.

[8] I.K. Fodor. A survey of dimension reduction techniques. T technical report UCRL-ID-148494, LLNL, 2002.

[9] D. Fradkin and D. Madigan. Experiments with random projection for machine learning. In *Proc. SIGKDD*, pages 517–522, 2003.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. SIGIR*, pages 50–57, 1999.

[11] A. Hyvarinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000. FastICA package: http://www.cis.hut.fi/∼xaapo/.

[12] S. Kaski. Dimensionality reduction by random mapping. In *Proc. Int. Joint Conf. on Neural Networks*, volume 1, pages 413–418, 1998.

[13] J. Kogan, C. Nicholas, and V. Volkovich. Text mining with information-theoretical clustering. *Computing in Science and Engineering*, accepted May 2003.

[14] T. Kolenda, L. K. Hansen, and S. Sigurdsson. Independent components in text. In *Advances in Independent Component Analysis*, pages 229–250. Springer-Verlag, 2000.

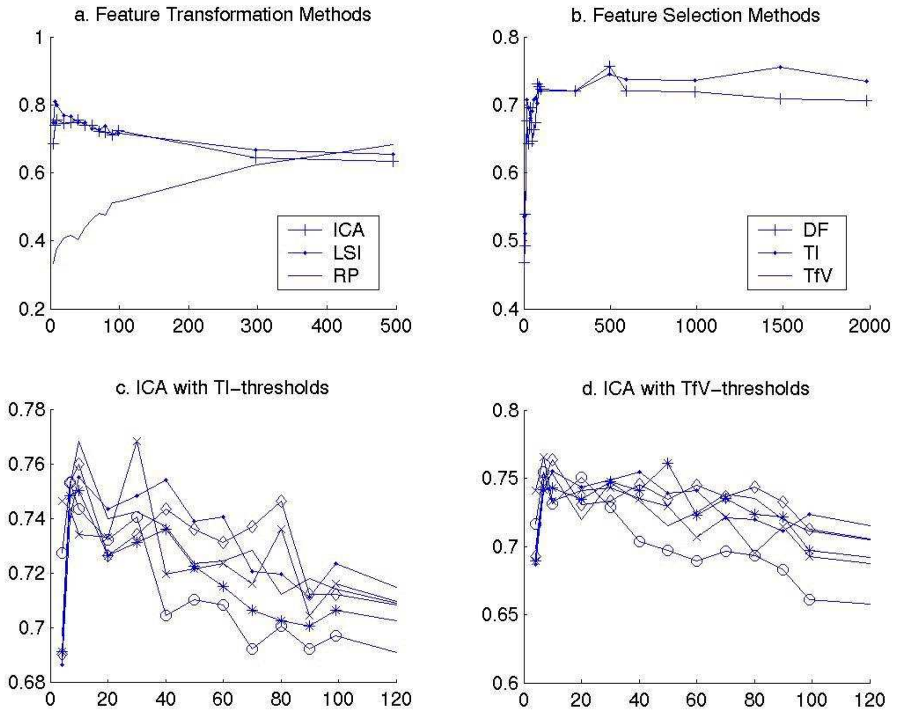

Figure 2: Comparison results of WebKB4. (a) results of feature transformation method. (b) results of feature selection methods. (c) results of ICA with different level of TI thresholding. (d) results of ICA with different levels of TfV thresholding.(The full legends are the same as in Figure 1, omitted here).

		ICA with TI thresholding					ICA with TfV thresholding				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
H_a	p-value	0.00	0.10	0.24	0.00	0.56	0.00	0.06	0.22	0.47	0.80
H_b	p-value	0.99	0.90	0.76	1.00	0.44	1.00	0.94	0.78	0.53	0.2

Table 4: P-values of the results of ICA combined with TI/TfV thresholding (WebKB4)

		ICA with TI thresholding					ICA with TfV thresholding				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
H_a	p-value	0.00	0.03	0.00	0.08	0.80	0.00	0.01	0.05	0.06	0.93
H_b	p-value	1.00	0.97	1.00	0.92	0.20	1.00	1.00	0.95	0.94	0.07

Table 5: P-values of the results of ICA combined with TI/TfV thresholding (CSTR)

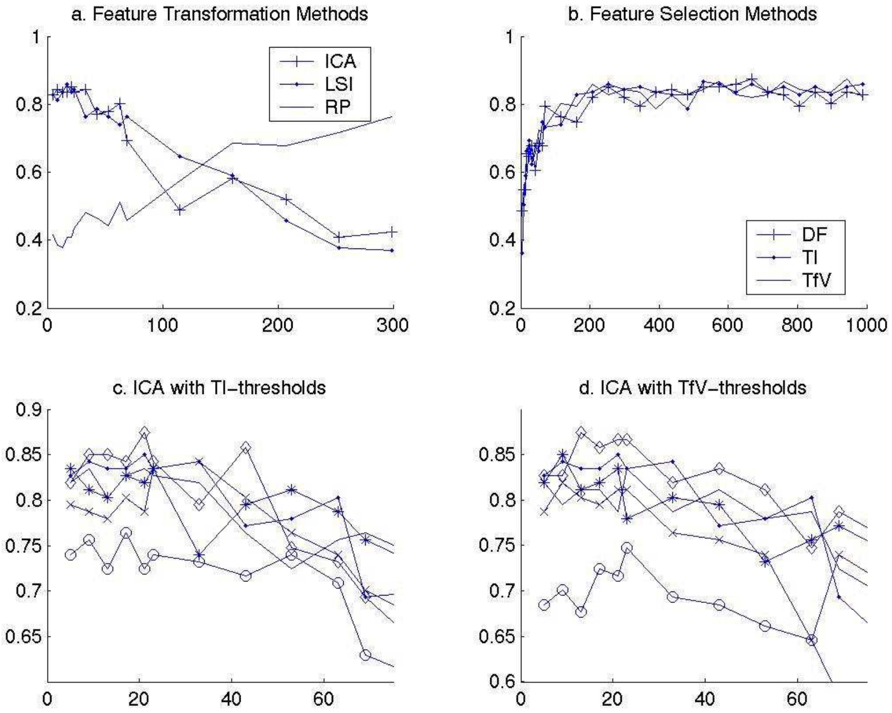

Figure 3: Comparison results of CSTR. a) results of feature transformation method. (b) results of feature selection methods. (c) results of ICA with different level of TI thresholding. (d) results of ICA with different levels of TfV thresholding.(The full legends are the same as in Figure 1, omitted here).

[15] J. Lin and D. Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *Proc. SDM'03 Conf., Text Mining Workshop*, 2003.

[16] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proc. ACM SIGPODS*, pages 159–168, 1998.

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

[18] B. Tang, X. Luo, M.I. Heywood, and M. Shepherd. A comparative study of dimension reduction techniques for document clustering. Technical Report CS-2004-14, Faculty of Computer Science, Dalhousie University, 2004. http://www.cs.dal.ca/research/techreports/2004/CS-2004-14.shtml.

[19] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. ICML*, pages 412–420, 1997.

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