Improving Text Classification using Local Latent Semantic Indexing

Tao Liu Zheng Chen* Benyu Zhang* Wei-ying Ma* Gongyi Wu
Nankai University, China ** Microsoft Research Asia* *Nankai University, China*
liut@office.nankai.edu.cn **{zhengc,byzhang,wyma}@microsoft.com* *wgy@nankai.edu.cn*

Abstract

Latent Semantic Indexing (LSI) has been shown to be extremely useful in information retrieval, but it is not an optimal representation for text classification. It always drops the text classification performance when being applied to the whole training set (global LSI) because this completely unsupervised method ignores class discrimination while only concentrating on representation. Some local LSI methods have been proposed to improve the classification by utilizing class discrimination information. However, their performance improvements over original term vectors are still very limited. In this paper, we propose a new local LSI method called "Local Relevancy Weighted LSI" to improve text classification by performing a separate Single Value Decomposition (SVD) on the transformed local region of each class. Experimental results show that our method is much better than global LSI and traditional local LSI methods on classification within a much smaller LSI dimension.

1. Introduction

Text classification is one of the core problems in Text Mining. Its task is to automatically assign predefined classes or categories to text documents [19]. There have been a lot of research done on text classification and SVM has been proved to be the best algorithm for text classification [6]. However, we found that the performance of a text classifier is strongly biased by the underlying feature representation. First, the inherent high dimension with tens of thousands of terms even for a moderate-sized text collection is prohibitively computational expensive for many learning algorithms [19]and easily raises the over fitting problem [11]. Secondly, polysemy (one word can have different meanings) and synonym (different words are used to describe the same concept) interfere with forming appropriate classification functions and make this task very difficult. [7,22].

There are typically two types of algorithms to represent the feature space used in classification. One type is the so-called "feature selection" algorithms, i.e. to select a subset of most representative features from the original feature space [8,20]. Another type is called "feature extraction", i.e. to transform the original feature space to a smaller feature space to reduce the dimension. Compared with feature selection, feature extraction can not only reduce the dimensions of the feature space greatly, but also succeed in solving the polysemy and synonym problem in a certain degree.

The most representative feature extraction algorithm is the Latent Semantic Indexing (LSI) which is an automatic method that transforms the original textual data to a smaller semantic space by taking advantage of some of the implicit higher-order structure in associations of words with text objects [1,2]. The transformation is computed by applying truncated singular value decomposition (SVD) to the term-by-document matrix. After SVD, terms which are used in similar contexts will be merged together. Thus, documents using different terminology to talk about the same concept can be positioned near each other in the new space [17].

Although LSI was originally proposed as an information retrieval method, it has also been widely used in text classification as well. For example, [19] used LSI to cut off noise during training process, [22] performed SVD on an expanded term-by-document matrix that includes both the training examples and background knowledge to improve text classification, [18] performed SVD on the term-by-document matrix whose terms include both single-word terms and phrase terms.

While LSI is applied to text classification, there are two common methods. The first one is called "Global LSI", which performs SVD directly on the entire training document collection to generate the new feature space. This method is completely unsupervised, that is, it pays no attention to the class label of the existing training data. It has no help to improve the discrimination power of document classes, so it always yields no better, sometimes even worse performance than original term vector on classification [15]. The other one is called "Local LSI", which performs a separate SVD on the local region of each topic [4,5,12,17]. Compared with global LSI, this method utilizes the class information effectively, so it improves the performance of global LSI greatly. However, due to the same weighting problem, the improvements over original term vector are still very limited.

It is noticed that in local LSI, all documents in the local region are equally considered in the SVD computation. But intuitively, different documents should play different roles to the final feature space and it is expected that more relevant documents to the topic can contributes more to the local semantic space than those non-relevant ones. So based on this idea, we propose a new local LSI method - “Local Relevancy Weighted LSI (LRW-LSI)”, which selects documents to the local region in a smooth way. In other words, LRW-LSI gives different weight to each document in the local region according to its relevance before performing SVD so that the local semantic space can be extracted more accurately. Experimental results shown later prove this idea and it is found LRW-LSI is much better than global LSI and ordinary local LSI methods on classification performance within a much smaller LSI dimension.

The rest of this paper is organized as the followings. In Section 2, we give a brief introduction to global LSI and different local LSI methods. In Section 3, we propose our new Local Relevancy Weighted LSI method. Then, several experiments are done to evaluate different LSI methods for text classification and their results are discussed in Section 4. Section 5 concludes this paper.

2. Related Works

The most straightforward method of applying LSI for text classification is the global LSI method, which performs SVD directly on the entire training set and then testing documents are transformed by simply projecting them onto the left singular matrix produced in the original decomposition for evaluation. [15,18,19,22]. Global LSI is completely unsupervised, and it makes no use of class label, word order, syntactic relations or other existing information. It aims at deriving an optimal representation of the original data in a lower dimensional space in the mean squared error sense but this representation does no help the optimal discrimination of classes. This is especially true when there are some original terms that are particularly good at discriminating a category/class. That discrimination power may be lost in the new semantic space. Hence, global LSI performs not as good as expected and even drops the classification performance. Furthermore, Wiener, Pedersen and Weigend [17] also found that global LSI performed increasingly worse as topic frequency decreased due to the fact that infrequent topics are usually represented by infrequent terms, and infrequent terms may be projected out of LSI representations as noise.

To integrate the class information, Hull [4] first proposed the concept of local LSI, which performed SVD on a local region of a topic so that the most important local structure, which is crucial in separating relevant documents from non-relevant documents, can be captured. A drawback of Hull’s solution is that the local region is defined by only relevant/positive documents which contain no discriminative information, which makes the improvement of classification performance very limited. Therefore, it is very necessary to add some of other documents to balance the local region. [5,12,17] did this work and they extended the local region by introducing some non-relevant documents which are most difficult to be distinguished from the relevant documents. The introduction of the most nearby non-relevant documents provides the most valuable discrimination information and is found to be more effective than using relevant documents alone.

Compared to global LSI whose cost of SVD computing is incurred only once, local LSI has an increased cost because a separate SVD has to be computed for each class. However, the SVD is only applied to the local region, which means that the matrix is far smaller than the one used in global LSI so the computation can be extremely fast.

In the following sub-sections, a brief introduction on local LSI methods is given. All local LSI methods are similar in the generation processes of local region. That is, each document in the training set is first assigned with a relevancy score related to a topic, and those documents whose scores are higher than a predefined threshold value are picked to generate the local region.

2.1. Relevant Documents Selecting Method (RDS)

RDS defines the local region for a topic as the relevant documents only [4]. It is the simplest method but the local region contains no discrimination information, so it is very limited to improve the classification performance.

On the other hand, the frequency of topic occurrence varies greatly from topic to topic. For example, in Reuters-21578 data, the biggest topic “earn” has roughly 30% of the documents while only five training documents belong to topic “platinum”. So it is very difficult to select a different optimal LSI dimension for different topic. Hence, RDS is not a recommended local LSI method.

2.2. Query-specific Screening Method (QS)

QS is the most widely used local LSI method which defines the local region of a topic as the n most similar documents, where similarity is measured using the inner product score to the query vector which can be generated by Rocchio-expansion or the m most predictive terms of the topic.

2.2.1. Query by Rocchio-expansion. Rocchio-expansion is the simplest method to combine relevant documents and non-relevant documents [10]. It has been widely used to define the query of a topic as the mean of relevant documents in the training set [3,12,13].

2.2.2. Query by predictive terms. Wiener et al. [17] used another method to define the query as the 100 most predictive terms of the topic where the predictive score of a term was measured by Relevancy Score (RS).

Many other measures can be used to rank terms for a topic, such as Information Gain (IG), x^2 statistic (CHI) and Mutual Information (MI) [8,20]. While there is no big difference in selecting a small set of terms between RS, IG and CHI [17], in this paper, we only consider two representative measures: CHI and MI. CHI is defined as equation (1) which measures the association between the term and the topic. MI is defined as equation (2) which only measures how important a term to a topic by the presence of a term occurs in the documents.

$$\chi^2(t,c) = \frac{N \times (p(t,c) \times p(\bar{t},\bar{c}) - p(t,\bar{c}) \times p(\bar{t},c))^2}{p(t) \times p(\bar{t}) \times p(c) \times p(\bar{c})} \quad (1)$$

$$MI(t,c) = \log \frac{p(t,c)}{p(t) \times p(c)} = \log p(t \mid c) - \log p(t) \quad (2)$$

2.3. SVM Screening Method (SS)

Both RDS and QS actually can be viewed as a classification process to generate the local region. So similar with them, for each class, a SVM classifier can be trained using training documents and then be used to classify each training document to get its confidence value. Finally, the n most confident documents are picked as the local region of that class. This method is called the SVM Screening method.

3. Local Relevancy Weighted LSI

As introduced above, in local LSI, each document in the training set is first assigned with a relevancy score related to a topic, and then the documents whose scores are larger than a predefined threshold value are selected to generate the local region. Then SVD is performed on the local region to produce a local semantic space. This process can be simply described as the jump curve in Figure 1. That is 0/1 weighting method is used to generate the local region where documents whose scores are larger than the predefined threshold value are weighted with 1 and others are weighted with 0.

The 0/1 weighting method is a simple but crude way to generate local region. It assumes that the selected documents are equally important in the SVD computation. However, it is obvious that each document plays a different role in the local semantic space and the more relevant documents should contribute more to the local semantic space, and vice versa. Furthermore, in real application, the size of local region is very difficult to tune. When the size is too big, non-relevant documents may be much more than relevant documents so that SVD may pay more attention to the semantic structure of non-relevant documents and the local semantic space may be very biased. On the other hand, when the size is too small, the local region may contain no enough discriminative information so that the local semantic space may also be limited. Based on these problems, an intuitive idea which can be described as the smooth curve in Figure 1 comes up. That is to introduce documents into the local region in a smooth way. In other words, the relevancy value of each document to a class is further used to weight the document so that more relevant documents can be introduced with higher weights and then they will do more contribution to SVD computation. Hence, the better local semantic space which results in better classification performance can be extracted to separate positive documents from negative documents.

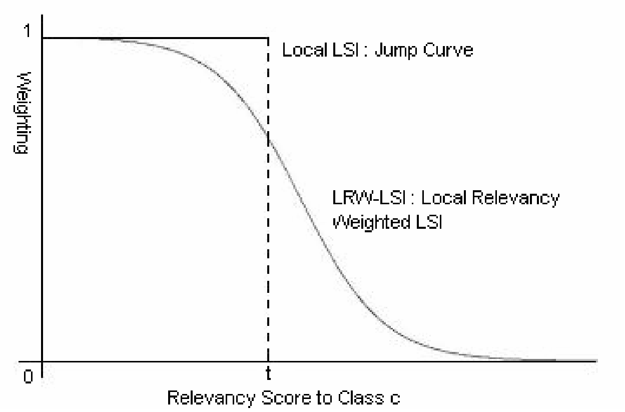

Figure 1. Local LSI and Local Relevancy Weighted LSI

This new method is named "Local Relevancy Weighted LSI (LRW-LSI)".

For each class, assume an initial classifier IC has been trained using training documents in term vector representation which can be can Rocchio classifier, Term Query classifier or SVM classifier as introduced in last Section. Then the training process of LRW-LSI contains the following six steps. At the first step, the initial classifier IC of topic c is used to assign initial relevancy score (rs) to each training document. Then at step two, each training document is weighted according to equation (3). The weighting function is a Sigmoid function which has two parameters a and b to shape the curve. At step three, the top n documents are selected to generate the local term-by-document matrix of the topic c . Then at step four, a truncated SVD is performed to generate the local semantic space. At step five, all other weighted training documents are folded into the new space. Then at the final step, all training documents in local LSI vector are used to train a real classifier RC of topic c .

$$\vec{d}_i{}' = \vec{d}_i * f(rs_i), \text{ where } f(rs_i) = \frac{1}{1 + e^{-a(rs_i + b)}} \quad (3)$$

In testing process, when a testing document comes in, it is first classified by the initial classifier IC to get its initial relevancy score. Then it is weighted according to the equation (3) and then folded into the local semantic space to get its local LSI vector. Then the local LSI vector is finally used to be classified by the classifier RC to decide whether it belongs to topic c or not.

Note that the parameters a and b of the Sigmoid function define how smoothly to introduce the documents. It is obvious that when the parameters a and b is suitable, for example, when a equals 0 or a and b are large enough, LRW-LSI is actually the same as local LSI, so local LSI is just a special case of LRW-LSI.

4. Experiments

In this Section, we first evaluate four Local LSI methods including QS-Roc, QS-CHI, QS-MI and QS-SS, and compare them with Term Vector and Global LSI. Then we evaluate Local Relevancy Weighted LSI method. SVM light[1] is chosen as the classification algorithm, SVDPAKC/sis[2] is used to perform SVD and F-Measure is used to evaluate the classification results. Two common data sets are used, including Reuters-21578 and Industry Sector.

Before performing classification, a standard stop-word list is used to remove common stop words and stemming technology is used to convert variations of the same words into its base form. Then those terms that appear in less than 3 documents are removed. Finally tf*idf (with "ltc" option) is used to assign the weight of each term in each document.

4.1. Data Sets

Text classification performance varies greatly on different dataset, so we choose two text collections, including Reuters-21578[3] and Industry Sector[4].

Reuters-21578 (Reuters) is the most widely used text collection for text classification. There are total 21578 documents and 135 categories in this corpus. The frequency of topic occurrence varies greatly from topic to topic. For example, the most frequent topic "earn" appears in more than 30% of the documents while "platinum" is a topic mentioned in only five training documents. In our experiments, we only chose the most frequent 25 topics and used "Lewis" split which results in 6314 training examples and 2451 testing examples.

Industry Sector (IS) is a collection of web pages belonging to companies from various economic sectors. There are 105 topics and total 9652 web pages in this dataset. Compared to Reuters, the topics are averagely distributed in documents. Also, a subset of the 14 categories whose size are bigger than 130 is selected for the experiments. Then a random split of 70% training examples and 30% testing examples is produced which results in 2030 training examples and 911 testing examples.

4.2. Classification Algorithm

Support Vector Machine (SVM) is chosen in our experiment which is very popular and proved to be one of the best classification algorithms for text classification [18,21]. SVM is originally introduced by Vapnic in 1995 for solving two-class pattern recognition problem [16]. It is based on the Structural Risk Minimization principle from the computational learning theory. The idea is to find a hypothesis h to separate positive examples from negative examples with maximum margin. In linear SVM, we use $w * x - b = 0$ to represent the hyper-plane, where the normal vector w and constant b are defined by the distance from the hyper-plane to the nearest positive and negative examples.

4.3. Performance Measures

For evaluating the effectiveness of classification, we use the standard recall, precision and F1 measure. Recall is defined to be the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of the system's assignments. The F1 measure, initially introduced by van Rijsbergen [9], combines recall (r) and precision (p) with an equal weight in the following form:

$$F1 = \frac{2rp}{r+p} \tag{4}$$

There are two ways to measure the average F1 of a binary classifier over multi categories, namely, the macro-averaging and micro-averaging. The former way is to first compute F1 for each category and then average them, while the later way is to first computer precision and recall for all categories and use them to calculate F1. It is clear that macro-averaging gives an equal weight to the performance on every category, regardless how rare or how common a category is, but micro-averaging favors the performance on common categories. Both of them are calculated in our experiments.

[1] http://svmlight.joachims.org/

[2] http://www.netlib.org/svdpack/

[3] http://www.daviddlewis.com/resources/testcollections

[4] http://www-2.cs.cmu.edu/afs/cs.cmu.edu

4.4. Experimental Results and Discussion

4.4.1. Global LSI and Local LSI. The first experiment we conduct is to compare four local LSI methods with original term vector and global LSI, including QS-Roc, QS-CHI, QS-MI and SS where QS-Roc means the query is defined as Rocchio-expansion, QS-CHI and QS-MI means the query is defined as the top 100 predictive terms selected by CHI and MI measure. For Reuters-21578, local region is defined as five times the number of positive documents but no less than 350 and no more than 1500 which is similar with [17]. For Industry Sector, local region is defined as five times the number of positive documents.

Table 1. Classification results on Reuters

Method	Dimension	Micro-F1	Macro-F1
Term	>5000	94.02	81.50
Global LSI	250	92.64	73.26
QS-Roc	200	92.90	84.12
QS-CHI	200	93.78	84.79
QS-MI	200	93.99	85.60
SS	200	91.69	83.49

The classification results of Reuters-21578 using different methods are shown in Table 1. As can be seen from this table, term vector is the best representation for SVM classifier. Compared to term vector, global LSI degrades the performance markedly especially on Macro-averaging F1 (relatively 10.1% reduction). There is no remarkable difference between Local LSI methods whose micro-averaging F1 is slightly dropped while macro-averaging F1 is slightly improved.

While micro-averaging favors performance on common topic but macro-averaging gives an equal weight to all topics, we separate the 25 topics of Reuters-21578 averagely into three levels including high, medium and low to conduct a further analysis as in [17]. Table 2 shows the results. As can be seen, global LSI performs increasingly worse as topic frequency decreases which is due to the fact that the signal of infrequent topics is so weak that it is usually treated as noise and projected out of the new semantic space. Contrary to the global LSI, local LSI focuses on the local structure and performs increasingly better as topic frequency decreased. For example, compared to term vector for low frequent topics, both F1 are dropped nearly 29% by global LSI but improved nearly 13% by local LSI.

Similar results can be found on Industry Sector data as shown in Table 3. The best performance is produced by term vector. While all topics have similar frequency which can be viewed as medium frequency, both F1 measures were greatly dropped by global LSI and slightly dropped by local LSI.

Table 2. Classification results on Reuters over three topic frequency ranges

Method	Micro-F1			Macro-F1		
	High	Med	Low	High	Med	Low
Term	95.3	88.3	71.3	85.3	87.8	71.1
Global LSI	94.6	83.1	50.7	83.8	81.7	50.7
Local LSI	94.6	89.6	80.4	85.1	89.3	80.1

Table 3. Classification results on Industry Sector

Method	Dimension	Micro-F1	Macro-F1
Term	>5000	80.91	80.61
Global LSI	250	63.95	63.99
QS-Roc	200	75.08	76.03
QS-CHI	200	76.55	77.81
QS-MI	200	75.56	76.07
SS	200	75.81	77.11

4.4.2. Local Relevancy Weighted LSI. For local relevancy weighted LSI, we use SVM classifier as the initial classifier IC to generate each document's initial relevancy score. And the parameters a and b of Sigmoid function are initially set with 5.0 and 0.2.

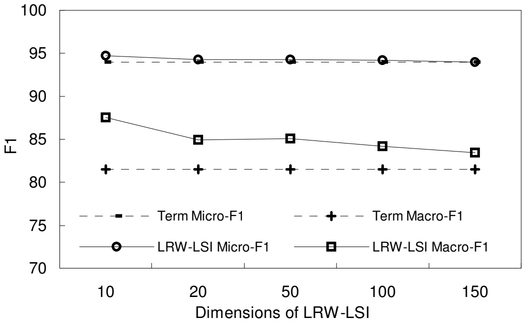

Figure 2. LRW-LSI results on Reuters

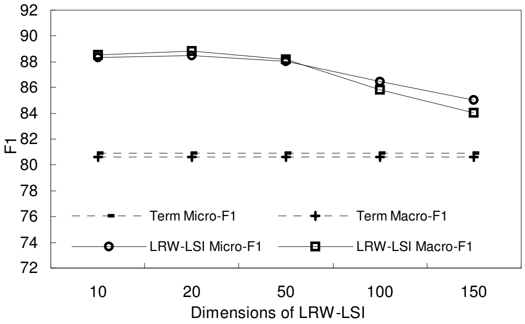

Figure 3. LRW-LSI results on Industry Sector

Figure 2 and Figure 3 displays the classification results on Reuters-21578 and Industry Sector. The dotted lines of term vector are displayed only as the reference points in terms of performance comparison. From these figures, the following observations can be made:

First, compared to term vector, LRW-LSI improves the both F1 performances greatly on both data. For example, using 10 dimensions on Reuters-21578, the micro-averaging F1 is improved by 0.8% and the macro-averaging F1 is improved by 7.4%; using 20 dimensions on Industry Sector, the micro-averaging F1 is improved by 9.4% and the macro-averaging F1 is improved by 10.2%.

Second, the optimal dimension of LRW-LSI is much smaller than that of global LSI and local LSI method. For example, on Reuters-21578, the optimal dimension of LRW-LSI is only 10 while that of the global LSI is 250 and local LSI is 200. Such small dimension also means that the computation is much faster than the computation of the global LSI and local LSI method. Table 4 shows the run time of different LSI methods on a PC with Pentium III 500MHz and 256M memory. The runtime includes both training procedure and testing procedure. As can be seen, term vector is the fastest and it needs only hundred seconds. Global LSI needs much more time than term vector due to the costly SVD computation on entire training set. Although SVD computation on local region is very fast, the overall computation on all topics is extremely high, so local LSI is not expected to be used in practice. Similar with local LSI, LRW-LSI has to perform a separate SVD on local region of each topic, but such a low LSI dimension makes LRW-LSI be extremely rapid. It needs only less than 3 times of runtime of term vector, so it can be widely used in practice.

Table 4. Run Time of Different Methods (Seconds).

Method	Reuters-21578	Industry Sector
Term	372	291
Global LSI	4702	1905
Local LSI	6508	5210
LRW-LSI	1080	752

Third, with the LSI dimension increases, the performances decrease slowly. But even in a relatively high dimension, the performances are still equal or above the performances of term vector. Using 150 dimensions, for example, on Reuters-21578 the micro-averaging F1 is equal and the macro-averaging F1 is still improved by 2.4%; on Industry Sector, the micro-averaging F1 is still improved by 5.1% and the macro-averaging F1 is still improved by 4.2%.

Forth, while local LSI has similar performance with term vector and LRW-LSI is much better than term vector, LRW-LSI is much better than local LSI not only in performance but also in much smaller LSI dimension. It means that LRW-LSI is more effective in separating relevant documents and nearby non-relevant documents. In order to find out the reason, we project the documents into coordinate plane using the first and the second local LSI factors as in [4,5].

Figure 6 shows the local LSI projection and Figure 7 shows the LRW-LSI projection for "interest" topic of Reuters-21578 where the red asterisks represent relevant documents and blue circles represent non-relevant documents. As can be seen from these figures, relevant documents and non-relevant documents are separated by Local LSI only to a certain degree, that is, there are still a big chunk of documents mixed together. Comparatively, LRW-LSI is much more successful in class separation, that is, non-relevant documents are almost clustered around the origin while relevant documents are widely distributed.

In the previous experiment, we set the parameters of Sigmoid function by experience. So in order to learn the influence of the parameters setting, we conduct two testing experiments using different tuning strategies. The first one is to fix the parameter a to 5.0 and change the parameter b from 0~1.0 and the second one is to fix the parameter b to 0.0 and change the parameter a from 0.0 to 6.0. In both tests, 10 dimensions are used for classification.

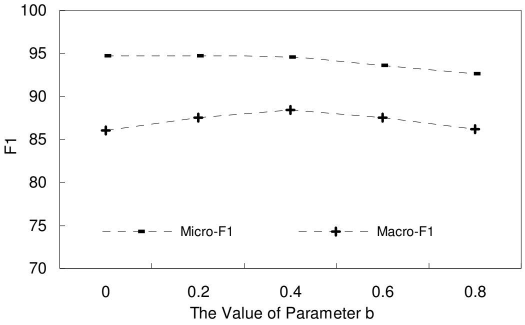

Figure 4 Parameter b Tuning on Reuters

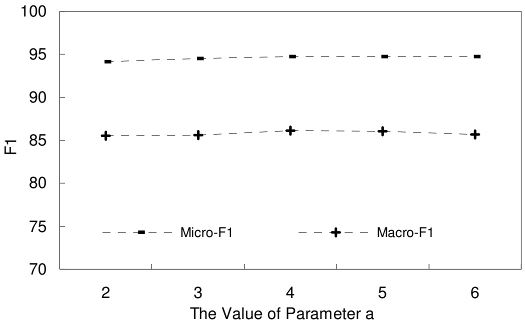

Figure 5 Parameter a tuning on Reuters

Figure 4 and Figure 5 show the testing results on Reuters-21578. We don't display the results on Industry Sector because its results are similar with Reuters-21578. As can be seen from these figures, LRW-LSI is more easily influenced by parameter b than by parameter a. But the performance influences of both parameters are very small, so in general, LRW-LSI is insensitive to both parameters.

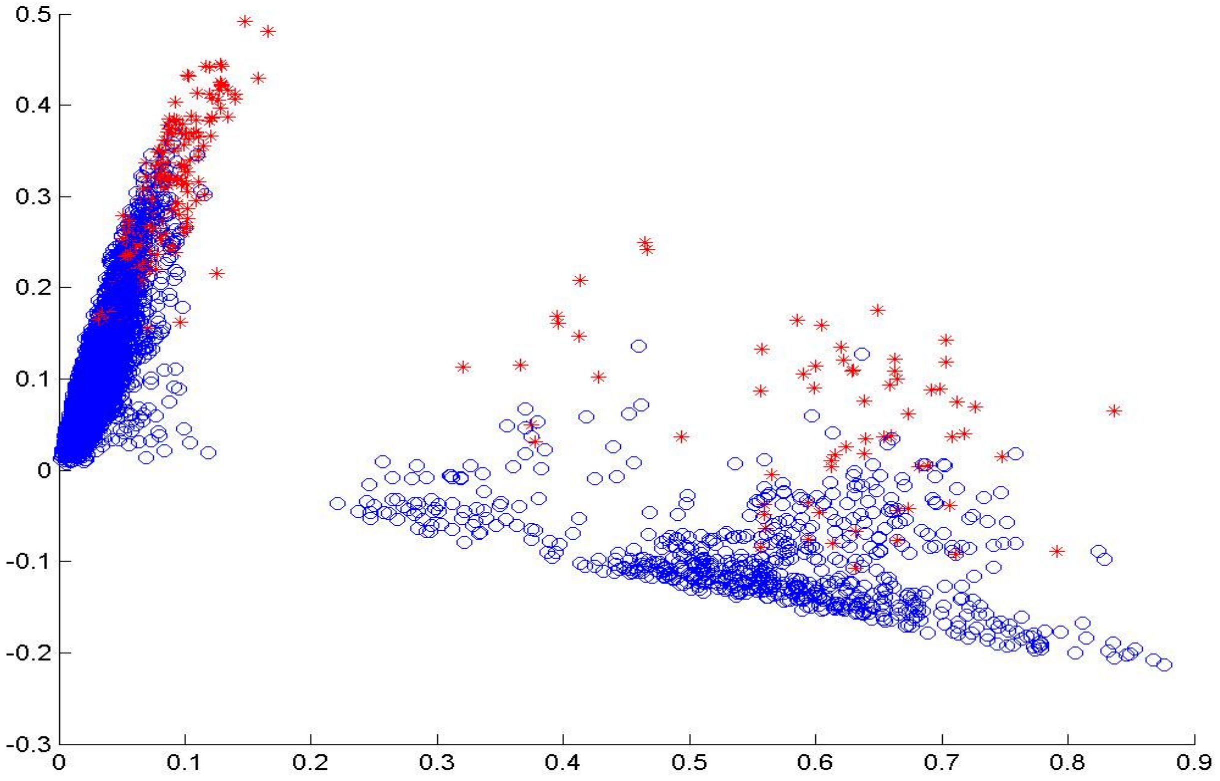

Figure 6. Local LSI projection for interest topic of Reuters

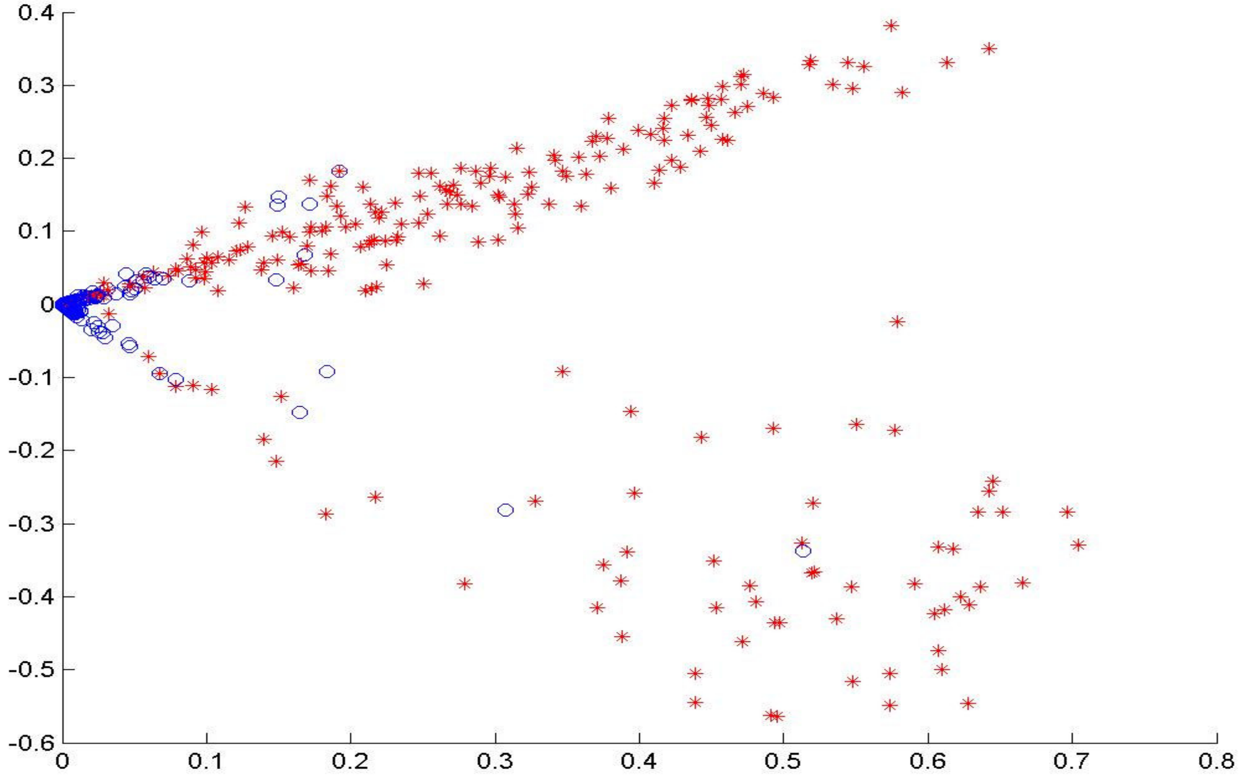

Figure 7. LRW-LSI projection for interest topic of Reuters

5. Conclusion

In this paper, we propose a new Local Relevancy Weighted LSI (LRW-LSI) method to help improve the text classification performance. This method is developed from Local LSI, but different from Local LSI in that the documents in the local region are introduced using a smooth descending curve so that more relevant documents to the topic are assigned higher weights. Therefore, the local SVD can concentrate on modeling the semantic information that is actually most important for the classification task. The experimental results verify this idea and show that LRW-LSI is quite effective. It can improve the classification performance greatly using a much smaller dimension compared to the global LSI and local LSI methods.

Another work we have done is a comparative study on global LSI and several local LSI methods for text classification. It is found that local space is more suitable for LSI than the global space. Global LSI can optimize representation of the whole original data in a low dimensional space but gives no help to optimizing the discrimination of the topic, so it always drops the classification performance. Local LSI captures the important local structure which is crucial in separating relevant documents from nearby non-relevant documents, so it succeeds in keeping or improving slightly the classification performance in a low dimension.

Acknowledgments

The authors would like to thank Weiguo Fan and Wensi Xi from Virginia Polytechnic Institute and State University for fruitful discussions.

References

[1] Berry, M. W., Dumais, S. T., & O'Brien, G. W, Using Linear Algebra for Intelligent Information Retrieval, *SIAM Review*, 37:573-595, 1995

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

[3] Hearst, M., Pedersen, J., Pirolli, P. & Schutze, H, Xerox Site Report: Four TREC-4 Tracks, *In D. Harman (Ed.) TREC-4*, pp. 97-119, 1996

[4] Hull, D, Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing, *Proc. of SIGIR'94*, pp. 282-289, 1994

[5] Hull, D.A, *Information Retrieval using Statistical Classifcation*. PhD thesis, Stanford University, 1995

[6] Joachims, T, Text Categorization with Support Vector Meachines: Learning with Many Relevant Features, *Proc. of ECML'98*, pp. 137-142, 1998

[7] Lewis, D.D. & Ringuette, M, A Comparison of Two Learning Algorithms for Text Categorization, *Proc. of SDAIR'94*, pp. 81-93, 1995

[8] Liu, T., Liu, S., Chen, Z., & Ma, W, An Evaluation on Feature Selection for Text Clustering, *Proc. of ICML'03*, pp. 488~495, 2003

[9] Rijsbergen, C.J, *Information Retrieval*. Butterworths, 2nd Edition, 1979

[10] Rocchio, J.J, Relevance Feedback in Information Retrieval, *In Gerard Salton, (Ed.)*, The Smart retrieval systeml experiments in automatic document processing, pp. 313~323, 1971

[11] Sebastiani, F, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34(1):1-47, 2002

[12] Schutze, H., Hull, D.A. & Pedersen, J.O, A Comparison of Classifiers and Document Representations for the Routing Problem, *Proc. of SIGIR'95*, pp. 229-237, 1995

[13] Schutze, H., Pedersen, J.O. & Hearst, M. A, Xerox TREC3 Report: Combining Exact and Fuzzy Predictors, *In D. Harman (Ed.) TREC-3*, 1995

[14] Schutze, H. & Silverstein, C, Projections for Efficient Document Clustering, *Proc. of SIGIR'97*, pp. 74~81, 1997

[15] Torkkola, K, Linear Discriminant Analysis in Document Classification, 2002

[16] Vapnic, V, *The Nature of Statistical Learning Theory*, Springer, 1995

[17] Wiener, E., Pedersen, J.O. & Weigend, A.S, A Neural Network Approach to Topic Spotting, *Proc. Of SDAIR'95*, pp. 317-332, 1995

[18] Wu, H. & Gunopulos, D, Evaluating the Utility of Statistical Phrases and Latent Semantic Indexing for Text Classification, *Proc. of ICDM'02*, pp. 713-716, 2002

[19] Yang, Y, Noise Reduction in a Statistical Approach to Text Categorization, *Proc. of SIGIR'95*, pp. 256-263, 1995

[20] Yang, Y. & Pedersen, J. O, A comparative study on feature selection in text categorization, *Proc. of ICML'97*, pp. 412-420, 1997

[21] Yang, Y. & Liu, X, A Re-examination of Text Categorization Methods,. *Proc. of SIGIR'99*, pp. 42-49, 1999

[22] Zelikovitz, S. & Hirsh, H, Using LSI for Text Classification in the Presence of Background Text. *Proc. of CIKM01*, pp. 113-118, 2001