Dimensionality Reduction for Text Using LLE

Chuan HE
School of Information Engineering, Beijing University of Posts and Telecommunications.
Beijing, China
hechuanbupt@gmail.com

Zhe DONG
School of Information Engineering, Beijing University of Posts and Telecommunications.
Beijing, China
jimmybupt@gmail.com

Ruifan LI
School of Information Engineering, Beijing University of Posts and Telecommunications.
Beijing, China
ruifan.li@gmail.com

Yixin ZHONG
School of Information Engineering, Beijing University of Posts and Telecommunications.
Beijing, China
zyx@bupt.edu.cn

Abstract:

Dimensionality reduction is a necessary preprocessing step in many fields of information processing such as information retrieval, pattern recognition and data compression. Its goal is to discover the representative or the discriminative information residing in raw data. Locally linear embedding (LLE), one of effective manifold learning algorithms, addresses this problem by computing low-dimensional, neighborhood preserving embeddings of high-dimensional data. The embedding is derived from the symmetries for locally linear reconstructions. And the computation of this embedding is related to an eigen-problem in the i mplement. Since LLE was proposed, it has been being applied to deal with image data only because it originated from facial recognition. However, the problem of curse of dimensionality is very prevalent. Therefore, we here try to apply this algorithm for text processing. In this paper, we introduce the LLE briefly and analyze its advantage and latent disadvantages, and the relationship between LSI and LLE in the graph embedding framework is then discussed from a theoretic view. Finally, the experimental results are show with the datasets of Reuters21578 and TDT2.

1. Introduction

1.1 Dimensionality Reduction

The problem of dimensionality reduction comes out in many practical and theoretic fields such as pattern recognition, text processing, gene sequence analysis, data visualization and data compression. On one side, the data in many tasks is modeled to be multidimensional or high-dimensional. In the image processing, the grey level of every pixel is usually regarded as one dimension, so a facial image with 256×256 pixels can be expressed in a vector with 65536 dimensions. The case in document categorization and clustering is similar. On the other hand, high-dimensional data is difficult to deal with because of the limited performance of many algorithms and the existing capacity for computing and storing in the hardware. For example, 100 evenly-spaced sample points suffice to sample a unit interval with no more than 0.01 distances between points; an equivalent sampling of a 10-dimensional unit hypercube with a lattice with a spacing of 0.01 between adjacent points would require 10^{20} sample points, which is intractable in practice. Because of the fact described above, dimensionality reduction is urgently on demand.

In the existing dimensionality reduction method, there are two classes [3 and 8] ----feature extraction and feature selection. Feature selection means to find the representative features or attributions in the original feature space, giving up the useless features. In statistics, chi-square test is an effective tool to select the features with helpful information. Feature extraction does not choose or operate in the original feature space, but map the original high-dimensional space into a new low-dimensional space, in which every feature may possess its own meaning physically. One of feature extraction methods is principal component analysis (PCA, also called KL Transformation) [3, 8 and 13] which is used and studied extensively. As a linear, global and unsupervised method, PCA computes the covariance matrix of the input data, chooses the eigen-vector corresponding to the large eigen-values as the projection coordinates, and projects the original data to the new low-dimensional data. It is rather easy to be understood and the implement is also very easy. The computation is cost on the eigen-problem of the covariance matrix with the size of $D \times D$ (D is the dimensionality of the original data), which is the cube of D in general. More useful and efficient algorithms for the eigen-problem are discussed in [10]. Besides PCA, manifold learning methods recently becomes increasingly attractive to researches. Locally linear embedding (LLE) [14, 16, 18 and 19], as a popular one, addresses this problem by computing low-dimensional, neighborhood preserving embeddings of high-dimensional data. The embeddings are derived from the symmetries for locally linear reconstructions,

978-1-4244-2780-2/08/$25.00 ©2008 IEEE

and the computation of the embedding is related to an eigen-problem in the implement. In Section 2 and 3, LLE and the related are show in detail.

1.2 Text Dimensionality Reduction

As we mentioned above, documents usually have high-dimensionality. Generally speaking, when handling texts or documents, we adopt the "bag of words" model [2]. It assumes simply that a text can be represented as a collection of out-of-order words or phrases, ignoring the grammar and semantics of the text. In the model, at first a word list needs to be set up according to training corpus or a ready dictionary directly. The frequencies of every word in a certain text then form a text vector to represent the text, where one word is regarded as one dimension. That is to say, the number of words in the word list is the dimensionality of every text.

In many practical applications like document retrieval, filtering and routing, the preprocessing for dimensionality reduction is simple and fast for the huge amount of data, including natural language processing techniques and Document-Frequencies filtering. The former mainly refers to stemming or lemmatizing, while the latter means to filer the words occurring in every text or nearly every text and the words in only a few texts. These methods are feasible and effective in many situations.

In the research of text and language computing, the methods of feature selection mean to find the representative features in the original feature space without mapping or embedding them into another space. One of advantages of feature selection is interpretability. The features after feature selection still have the semantic meanings to some degree, but the dimensions after feature extraction cannot be well interpreted by researchers so far. So the useful feature selection methods should not be ignored: document frequencies, chi-square, information gain, mutual information and so on. The details can be found in three surveys [9, 17, and 22]. In the framework of text processing, one of feature extraction ways is latent semantic analysis (LSA or latent semantic indexing, LSI) [7]. The goal of LSA is to find a low-rank approximation to the term-document matrix. It is implemented by the singular value decomposition (SVD), which can be proved to be equivalent to PCA mathematically. Another algorithm applied to text dimensionality reduction is locality preserving projection (LPP, or LPI in the context of text processing) [4, 11 and 12]. LPI provides a linear graph embedding to the dimensionality curse by using the graph Laplacian. LPI is local, unsupervised, and it is similar to LLE. Therefore, the feasibility and effectiveness of LPI motivates us the application of LLE in text dimensionality reduction.

1.3 Arrangement

In this part, we give a concise background on text dimensionality reduction and its typical methods. In next section, the motivation and main idea of LLE algorithm will be illustrated, and some computation problem will also be show. A comparison of LLE and graph embedding is also discussed in Section 3, especially how to construct the graph and weights. There are the experiments on two datasets ----Reuters21578 and TDT2 in Section 4. At the last part of this paper, we discuss and conclude the trial of text dimensionality reduction using LLE.

2. LLE Algorithm

2.1 Motivation and Main Idea

All of manifold learning algorithm has the assumption that the data are sampled in a low-dimensional nonlinear manifold which is embedded in a high-dimensional space, and there is a mapping or embedding with only a few parameters to reveal the relationship between the two spaces. A manifold has a locally Euclidean property, while the global structure are rather complex. Therefore, the embedding of LLE needs to preserve the local configuration of nearest neighbors. In another words, the nearby points in the high-dimensional space remain nearby and similar co-located with respect to another in the low-dimensional space.

How does LLE model the embedding? It is the locally linear reconstruction. At first, each point is represented or reconstructed by its neighbors. It is necessary to note that the neighbors of each point are determined by the neighboring graph. By minimize the reconstruction error, the algorithm can compute the reconstruction weights of every points. It is the reconstruction weights that store the neighboring information of all the data point in the local manifold. Leaving the original data aside, the next step only makes use of these reconstruction weights. In order to preserve the neighboring information, the thought of symmetry is adopted here to calculate the low-dimensional data after the mapping. The reconstruction errors are minimized with respect to the new data points while the reconstruction weights here are fixed.

2.2 LLE Algorithm

In the formalization, the input data is denoted by X_i (i=1,…,N), xi is in the D-dimensional space. The algorithm has three steps. At first, the neighborhood graph is constructed. There are two strategies of construction the graph mainly:

- After computing the distance between the point X_i and any other points, K nearest neighbors of X_i are chose to reconstruct it. Here K is a parameter to be adjusted.
- After computing the distance between the point X_i and any other points, all the neighbors of X_i within a ball or hyper ball of fixed radius L are chose to reconstruct it. Here L is also a parameter.

Note that the distance here is calculated in the metric of Euclidean space. Repeat this for each point to find out its neighbors. We can understand the step in the form of a matrix S of neighboring relationship. S is $N\times N$ and unnecessarily symmetric. Its element S_{ij} means the

connectivity among points, which only can be either 1 or 0. If X_j is one of the neighbors of X_i, $S_{ij} = 1$; vice verse.

The next step is to minimize the reconstruction errors with respect to the reconstruction weights for each neighbor for each point. The errors are measured by the cost function as:

$$W = \arg\min_{W} E(W) = \arg\min_{W} \sum_{i} \| \vec{X}_i - \sum_{j} W_{ij} \vec{X}_j \|^2 \qquad (1)$$

X_i is the raw point, X_j is the neighbors of X_i, $\sum_j W_{ij} \vec{X}_j$ is the reconstruction point, and so the error is expressed by $\vec{X}_i - \sum_j W_{ij} \vec{X}_j$. There is a constraint to keep the translation invariance of the embedding: $D_{ii} = \sum_j W_{ji}$. With a Lagrange multiplier, the optimization problem has a solution with close form.

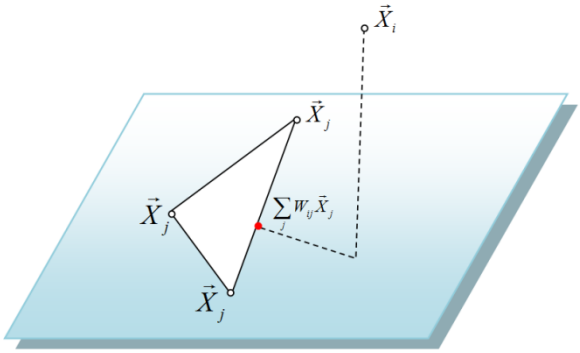

Figure 1. The linear reconstruction in the original space

In the last step, the goal is still to minimize the reconstruction errors. But the reconstruction weights are known, and the minimization is conducted in a new low-dimensional space. Formally, the problem is described as follow:

$$Y = \arg\min_{Y} F(Y) = \arg\min_{Y} \sum_{i} \| \vec{Y}_i - \sum_{j} W_{ij} \vec{Y}_j \|^2 \qquad (2)$$

Similar to the second part, Y_i is the original point, Y_j is the neighbors of Y_i, $\sum_j W_{ij} \vec{Y}_j$ is the reconstruction point, and $\vec{Y}_i - \sum_j W_{ij} \vec{Y}_j$ is the error. However, all the variables are in the low-dimensional space. The information in the original space is all kept in the reconstruction weights W_{ij}, which connect the two spaces. The optimization will not be performed only with (2) , because the solution space is too large. Without affecting the cost function in (2), some constraints necessarily make the problem well-posed:

$$\sum_i \vec{Y}_i = 0 \qquad (3)$$

$$\frac{1}{N} \sum_i \vec{Y}_i \cdot \vec{Y}_i^T = I \qquad (4)$$

3. The related about LLE

3.1 Some drawbacks

One of the drawbacks is the sampling assumption which requires that the sampled manifold is smooth and the sample data is sufficient to describe the local Euclidean property. Put another way, every point and its neighbors lie on a locally linear pitch so that locally linear reconstruction makes sense. In fact, many high-dimensional available data does not agree with this, especial the complex irregular data.

The second is that LLE intends to map the data points closer sometimes. The main motivation is to preserve the neighborhood of the nearby points, but it has no idea about how to handle the faraway points. At the same time, the neighborhood on the manifold is determined ironically by Euclidean metric. Under some circumstances, two points with a long geodesic distance may become "neighbors" of each other just as a result of a short Euclidean distance. And then the consequence is that they become real neighbors in the new space. In Figure 2, the example points are point A, B and C. For A and B, the mapping is naturally verified. For B and C, they look close intuitively while they are far away from each other when the roll is unfolded. LLE often fails in this case.

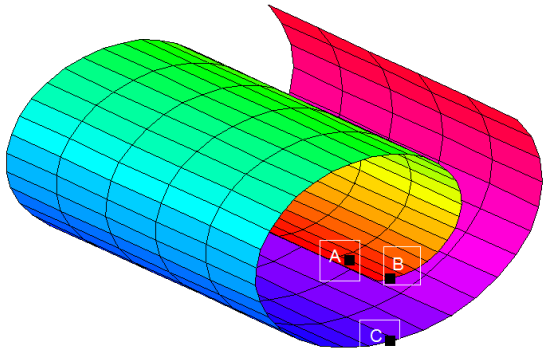

Figure 2-a. Three points A, B and C in the roll

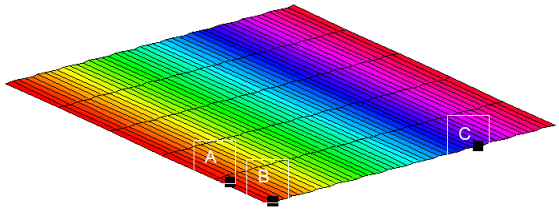

Figure 2-b. Three points A, B and C in the unfolded roll

Another detailed but not trivial problem lies on the constraint $\sum_j W_{ij} = 1$ in the second step. From the error expression, this may imply to equalize reconstruction points to the raw point X_i. However, this condition constraints the minimum solution in a linear simplex

whose vertices are the neighbors of X_i. If X_i is not in the simplex, the construction errors cannot be zero, say, the construction point is not the best choice to describe or displace the raw point X_i. So the construction weights cannot be calculated to be optimal subject to this constraint. If we relax this problem to unconstrained optimization, the construction point is different and even overlaps on the raw point. Figure 3 illustrates the difference. We suppose that in the case of two dimensions $X_1(0, 1)$, $X_2(0, 0)$, and $X_3(1, 0)$ is reconstructing $X_0(1, 1)$. Figure 3-a is the case with the condition: the construction point is the midpoint of X_1 and X_3, $W_{01}=W_{03}=0.5$, $W_{02}=0$; Figure 3-b is the case without the condition: the construction point and X_0 overlap, $W_{01}=W_{03}=1$, $W_{02}=0$. There is no construction error for X_0.

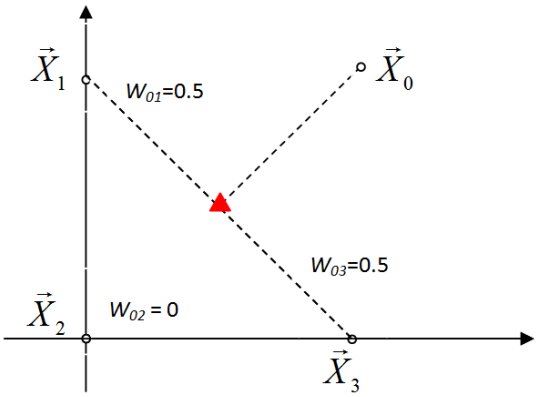

Figure 3-a. The reconstruction with one weights sum

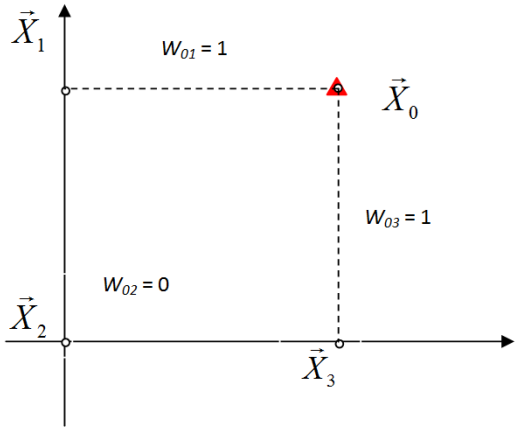

Figure 3-b. The reconstruction without one weights sum

3.2 LLE in Dimensionality Reduction

As we mentioned, a great number of transforming or mapping methods (feature extraction methods) have been studied for dimensionality reduction. Here, we can categorize them from three views, and then analyze and explain LLE from these views.

Linear vs. Nonlinear. Linear approaches always try to find some projection vectors by matrix computation, such as PCA, independent component analysis (ICA), and linear discriminative analysis (LDA). Nonlinear approaches mainly include kernelized linear ones and manifold learning methods. Kernelized approaches imply there is an intermediate step to map the data to nonlinear space by certain kernel functions, such as kernel PCA. Manifold learning assume that the high-dimensional data is sampled from nonlinear lower dimensional manifold, and learning algorithms are trying to find the embedded manifold structure. Locally linear embedding (LLE) and Isomap are typical manifold learning approaches.

Local vs. Global. Local approaches only consider the neighborhood information of a data point in feature space when dealing with one point. The neighbor might depend on the distances between two points. Literally, LLE only use the neighboring point for reconstruction. Global approaches believe all the data points are related to some degree, so all of them should be taken into account. From the graph view, a complete graph is constructed in the global approaches while a disconnected graph including several sub-graphs corresponds to the local approaches.

Supervised vs. Unsupervised. The goal of unsupervised approaches is to represent the raw high-dimensional data in low-dimensional spaces while supervised ones are to discriminate the data with different labels in low-dimensional spaces. In face recognition, Fisherface is supervised, and Eigenface is unsupervised.

LLE is nonlinear, local, and unsupervised. Local methods only calculate the neighboring distances and the neighboring reconstruction weights, without caring about all the data points for the input, so the fast computation is very attractive. Note that LLE is not linear essentially, though it looks like linear. The nonlinearity of LLE is up to the assumption that all the data are sampled in an underlying nonlinear manifold. Manifold structure is proved to be effective in many complex problems such like image classification, where linear methods or systems are too simple and naïve to describe the characteristics and discover the intrinsic structures when facing the image and text problems. As to how to use it in supervised framework, one can either extend the supervised version of LLE, or take LLE as the preprocessing and then classify or fit the new data in a new space with appropriate supervised algorithms.

3.3 LLE and Graph Embedding

According to [23], graph embedding provides dimensionality reduction with a generalized framework, and many algorithms are its specialized version, including LSI and LPI. In this framework, one first needs to derive a sparse graph by the neighboring relationship. Then a weight matrix W is constructed in various ways; most algorithms differ in this step. The last and key step is to compute the projected points by:

$$Y = \arg\max_{Y} \sum_{ij} \| \vec{Y}_i - \vec{Y}_j \|^2 W_{ij} \qquad (5)$$

Where $Y=(Y_1, Y_2, \ldots, Y_N)$, Y_i (a d-dimensional vector) is the projected point, W is the weight matrix, and D is a diagonal matrix whose entries are column sums of W, $D_{ii} = \sum_j W_{ji}$. Note that $L=D-W$ is the graph Laplacian,

which is described in [23].

The case of LSI and LPI in graph embedding is at large discussed in the [6, 7, 12]. Now, we give the theoretical relationship about LLE and graph embedding.

$$
\begin{aligned}
F(Y) &= \sum_i \| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \|^2 \\
&= \sum_i \| \sum_j W_{ij} (\vec{Y}_i - \vec{Y}_j) \|^2 \\
&= \sum_i (\sum_j W_{ij} (\vec{Y}_i - \vec{Y}_j))^T (\sum_j W_{ij} (\vec{Y}_i - \vec{Y}_j)) \\
&= \sum_i (\sum_j W_{ij} (\vec{Y}_i - \vec{Y}_j))^T (\sum_k W_{ik} (\vec{Y}_i - \vec{Y}_k)) \\
&= \sum_i \left\{ \begin{array}{l} \sum_j W_{ij} \sum_k W_{ik} \vec{Y}_i^T \vec{Y}_j - \sum_j W_{ij} \sum_k W_{ik} \vec{Y}_i^T \vec{Y}_k \\ -\sum_j W_{ij} \sum_k W_{ik} \vec{Y}_j^T \vec{Y}_i + \sum_j W_{ij} \sum_k W_{ik} \vec{Y}_j^T \vec{Y}_k \end{array} \right\} \\
&= \sum_i \sum_j \vec{Y}_i^T \vec{Y}_j - \sum_i \sum_j W_{ij} \vec{Y}_i^T \vec{Y}_j - \sum_i \sum_j W_{ji} \vec{Y}_i^T \vec{Y}_j \\
&\quad + \sum_k \sum_j W_{kj} \sum_i W_{ki} \vec{Y}_i^T \vec{Y}_j \\
&= \sum_{ij} (\delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}) \vec{Y}_i^T \vec{Y}_j
\end{aligned}
$$

With the constraints of (3) (4), the optimization can be modified into:

$$
\begin{aligned}
Y &= \arg\min_Y F(Y) \\
&= \arg\min_Y \sum_{ij} (\delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}) \vec{Y}_i^T \vec{Y}_j \\
&= \arg\min_Y \sum_{ij} (\delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}) \| \vec{Y}_i - \vec{Y}_j \|^2
\end{aligned}
$$

Here, in LLE $\delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$ can be regarded as whole weights in the (2). In other word, the weights are expressed in $\delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$.

4. Experiments

In this section, experiments are performed on the datasets, Reuters21578 and TDT2 respectively, which are standard document collections for research. The text representation in original space and the ones after LLE and LSI algorithms are compared. Usually, the good mapping for text representation can keep the discriminative information, say, the labels of the data points preserved. So we take text categorization as our task here.

4.1 Evaluation on Reuters21578

4.1.1 Experimental Preparation

Before we vectorize the document in the collection, it needs to be preprocessed. Reuters21578 collection is claimed to include 21578 documents. The documents in two or more categories (with two or more topics) are removed, and the duplicated documents are also deleted from the collection. There are 65 topics in the collection, and the number of documents in different topics varies from 1 to 3713, 8293 in all. Considering the problem of the computation complexity, we then *randomly* choose 15 subsets from 8293 documents. Every subset contains N (from 143 to 748) documents and the number of topics in one subset ranges from 3 to 7. Notice that the number of documents with the same topic is not evenly distributed. In every subset, the words with low document frequency are removed, which leaves D (about 4000) words. So with term frequency indexing scheme, the document are represented by a vector of D dimensionality. Every subset can also expressed by a matrix with $D \times N$.

Table 1. The precision and dimensionality in 15 subsets of Reuters21578

Subset	#Topics	#Docs	Baseline	LSI		LLE	
			Precisions	Dims	Precisions	Dims	Precisions
1	3	410	70.73	405	70.73	10	**73.41**
2	3	237	71.78	235	72.21	11	**86.16**
3	4	255	51.06	254	51.48	34	**61.18**
4	3	599	66.44	598	66.44	38	**75.80**
5	4	175	55.52	174	56.11	12	**70.26**
6	4	223	51.58	222	51.58	10	**61.54**
7	7	212	46.21	211	46.21	28	**46.67**
8	3	652	68.56	651	**68.87**	65	68.72
9	4	248	81.90	244	82.30	41	**89.50**
10	5	143	52.38	142	52.38	6	**65.62**
11	5	385	65.99	383	66.50	11	**70.63**
12	4	166	60.04	165	60.04	32	**64.82**
13	3	783	72.28	779	**72.41**	39	67.71
14	5	249	42.97	244	**43.78**	27	43.35
15	5	554	68.23	247	**68.42**	45	66.09

4.1.2 Experimental design and results

Due to the supervised task, K-Nearest-Neighbor is used for classification. For every subset, three results are recorded: the first is the classification precision in the original text space; the second is the precision in the LSI-transformed space and the corresponding dimensionality; the last is the best precision in the LLE-transformed space and the corresponding dimensionality. Note that the parameter—the reduced dimensionality in LLE is tunable, and we here choose the one with the best precision for text categorization.

Table 1 show the specific result of every record in 15 subsets. As we can see in almost all the subsets, the precisions in the LSI-transformed space are slightly higher than or equal to the ones in the original space, and the best precisions in the LLE-transformed space are greatly higher than the ones in both the two spaces. From the view of the dimensionality, the best dimensionalities in LLE-transformed space are apparently lower than the ones in the other two spaces. In Figure 4, the average precisions are compared in the form of histograms.

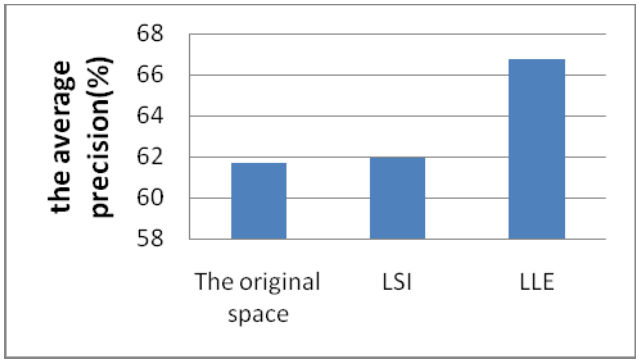

Figure 4. The average precision on 15 subsets of Reuters21578

Table 2. The precision and dimensionality in 20 subsets of TDT2

Subset	#Topics	#Docs	Baseline	LSI		LEE	
			Precisions	Dims	Precisions	Dims	Precisions
1	6	539	86.06	529	86.99	19	**91.28**
2	6	469	77.39	467	78.46	24	**91.05**
3	5	535	91.43	527	91.24	15	**95.34**
4	3	736	89.53	728	90.48	29	**95.11**
5	6	764	70.03	755	71.07	44	**86.51**
6	6	901	83.35	881	84.32	47	**93.90**
7	4	817	89.59	802	89.59	22	**92.04**
8	7	882	89.02	867	90.12	24	**94.45**
9	11	751	68.99	745	69.79	23	**84.30**
10	5	914	82.39	895	84.14	56	**92.57**
11	6	969	88.24	956	88.44	25	**95.98**
12	5	858	76.80	846	77.03	37	**83.56**
13	5	506	77.86	498	78.06	18	**90.51**
14	5	600	83.83	592	84.33	29	**89.17**
15	3	712	86.09	695	86.51	9	**98.17**
16	6	621	85.51	615	85.67	41	**90.18**
17	3	940	94.15	923	94.26	68	**98.83**
18	6	791	78.25	780	78.88	22	**90.14**
19	3	665	96.09	653	96.09	5	**99.85**
20	6	707	75.11	697	77.09	20	**94.49**

4.2 Evaluation on TDT2

4.2.1 Experimental Preparation

After the preprocessing similar to the one of Reuters21578, TDT2 dataset contains 9394 documents, which belong to 30 topics. According to the method used in Reuters21578, we sample *randomly* from this collection, and obtain 20 subsets, in which the number of documents ranges from 469 to 969 and the number of topics ranges from 3 to 11.

4.2.2 Experimental design and results

The experiments on TDT2 are designed in the same way of the one of Reuters21578. Table 2 and Figure 5 describe the results on the specific subsets and the whole collection respectively. The LLE algorithm can gain the best precision over the other two in all the subsets and on average. Besides, the precisions here are higher than the ones on the Reuters21578, the reason of which might be that TDT2 collection is well-separated to some degree.

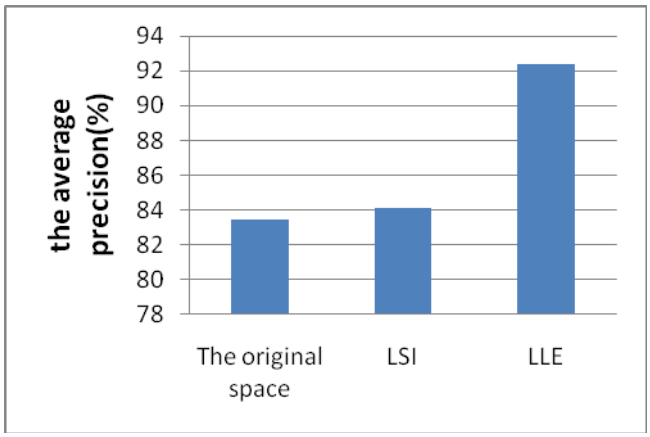

Figure 5. The average precision on15 subsets of TDT2

5. Discussion and Conclusion

In the first three section, the LLE and text dimensionality reduction are discussed in theory and then the experiments in Section 4 prove the effectiveness of text dimensionality reduction by the LLE algorithm. In terms of the classification precision, the LLE obviously outperforms the other two methods. Besides, there are several problems to be explored:

- The computation time of LLE is much less than LSI when K-Nearest-Neighbor is used, though no proof is given. In KNN and LLE, it is necessary to compute the distance between any two of data point. Therefore, when LLE+KNN is applied, the distances are computed for only once. However, LSI cannot save this part of time.

- The reduced dimensionality of LLE is much less than LSI. The mathematical essence is SVD, which removes all the zero singular values (eigenvalues) and leaves all the nonzero singular values. In text dimensionality reduction, the rank of data matrix is close or just is the number of documents in subsets. The dimensionality after LSI will not reduce too much. In other side, the dimensionality in LLE is adjustable, which give us much space to obtain the best precision. From the experiments, the reduced dimensionality of LLE varies from 5 to 68, which means the instinct dimensionality of the text structure possibly.

Reference

[1] Belkin M., and Niyogi P., *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*, Neural Computation, Volume 15, 2003

[2] Bellman, R. E., *Adaptive control Preocesses*, Princetion University Press, Princeton, NJ. 1961.

[3] Bishop C., *Machine Learning and Pattern Recognition*, Cambridge University Press, 2006

[4] Cai D. and He X., *Orthogonal Locality Preserving Indexing*, Proc. of the 28th International ACM SIGIR, 2005

[5] Cai D., He X. and Han J., *Document Clustering Using Locality Preserving Indexing*, IEEE Transaction on Knowledge Discovery Engineering, Volume 17, 2005

[6] Cai D., He X., Hu Y., Han J. and Thomas H., *Learning a Spatially Smooth Subspace for Face Recognition*, Proc. 2007 IEEE International Conference on Computer Vision and Pattern Recognition, 2007

[7] Deerwester S., Dumais S. and Harshman R., *Indexing by Latent Semantic Analysis*, Journal of the American Society for Information Science, 1999

[8] Duda O. R. and Hart E. P., *Pattern Classification*, 2nd edition, Wiley-Interscience, 2000

[9] Forman G., *An Extensive Empirical Study of Feature Selection Metrics for Text Classification*, Journal of Machine Learning Research, Volume 3, 2003

[10] Golub H. G. and Charles F. Van Loan, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, 1996

[11] He X. and Niyogi P., *Locality Preserving Projections*, Advances in Neural Information Processing Systems (NIPS) 15, 2003

[12] He X., Cai D. Liu H. and Ma W., *Locality Preserving Indexing for Document Representation*, Proc. of 27th International ACM SIGIR, 2004

[13] Jolliffe I. T., *Principal Component Analysis*, 2nd edition, Springer, NY, 2002

[14] Kouropteva O., Okun O., Hadid A., Soriano M., Marcos S. and Pietikainen M., *Beyond Locally Linear Embedding Algorithm*, Technical Report, MVG, 2002

[15] Lewis D. D., Yang Y., Rose G. T. and Li F., *RCV1: A New Benchmark Collection for Text Categorization Research*, Journal of Machine Learning Research, Volume 5, 2004

[16] Ridder de D., Kouropteva O., Okun O., Pietikainen M. and Duin R. P., *Supervised Locally Linear Embedding*, Technical Report, MVG, 2003

[17] Rogati M. and Yang Y., *High-Performing Feature Selection for Text Classification*, Proc. of the International ACM CIKM, 2002

[18] Roweis T. S. and Saul K. L., *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science, Volume 290, 2000

[19] Saul K. L. and Roweis T. S., *Think Globally, Fit Locally: Unsupervised Learning of Low-dimensional Manifolds*, Journal of Machine Learning Research, Volume 4, 2003

[20] Scholkopf B., Smola, A. and Muller K. R., *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*, Neural Computation, Volume 10, Page 1299-1319, 1998

[21] Tenenbaum J. B. Vin de Silva, and Langford J. C., *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science, Volume 290, 2000

[22] Yang Y. and Pedersen J., *A Comparative Study on Feature Selection in Text Categorization*, Proc. of the 14th International Conference on Machine Learning, 1997

[23] Yan S., Xu D., Zhang B. and Zhang H., *Graph Embedding and Extension: a Framework for Dimensionality Reduction*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007