Sprinkled Latent Semantic Indexing for Text Classification with Background Knowledge

Haiqin Yang and Irwin King

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
{hqyang,king}@cse.cuhk.edu.hk

Abstract. In text classification, one key problem is its inherent dichotomy of polysemy and synonym; the other problem is the insufficient usage of abundant useful, but unlabeled text documents. Targeting on solving these problems, we incorporate a sprinkling Latent Semantic Indexing (LSI) with background knowledge for text classification. The motivation comes from: 1) LSI is a popular technique for information retrieval and it also succeeds in text classification solving the problem of polysemy and synonym; 2) By fusing the sprinkling terms and unlabeled terms, our method not only considers the class relationship, but also explores the unlabeled information. Finally, experimental results on text documents demonstrate our proposed method benefits for improving the classification performance.

1 Introduction

Text classification (or categorization) is one of key problems in text mining. It aims to automatically assign unlabeled documents to one or more predefined classes based on their content. A number of statistical and machine learning techniques have been developed to solve the problem of text classification, e. g., regression model, k-nearest neighbor, decision tree, Naïve Bayes, Support Vector Machines, etc. [4,9]. There are several difficulties for this task. First, the existence of polysemy (a word contains multiple meanings) and synonym (different words express the same concepts) makes it hard to form appropriate classification models [3,11,8]. Second, documents usually representing by "bag of words" are inherently in very high dimension feature space. It increases the chance of overfitting [9]. Third, due to the difficult and tedious nature of labeling, training samples are sometimes extremely limited, this makes it difficult to make decisions with high confidence [11].

In order to solve the problems of polysemy and synonym, researchers usually adopt the Latent Semantic Indexing (LSI) technique [3] due to its empirically effective at overcoming these problems in text classification. LSI also succeeds in a wide variety of learning tasks, such as search and retrieval [3], classification [11,8] and information filtering [5,6], etc. It is a vector space approach for modeling documents, and many papers have claimed that this technique brings out the latent semantics in a collection of documents [3]. Typically, LSI is applied in an unsupervised paradigm. It is based on the well known mathematical technique called Singular Value Decomposition (SVD) and maps the original term-document matrix into a low dimensional space. Recently, researchers have incorporated class information into LSI for text classification. For example, [10] has presented a technique called Supervised LSI which is based on iteratively identifying discriminative eigenvectors from class-specific LSI representations. [1] has introduced a very simple method, called "sprinkling" to integrate class information into LSI. The basic idea of [1] is to construct an augmented term-document matrix by encoding class labels as artificial terms and appending to training documents. LSI is performed on the augmented term-document matrix, where class-specific word associations being strengthened. These pull documents and words belonging to the same class closer to each other in a indirect way. [2] extends the idea of [1] by adaptively changing the length of sprinkling terms based on results from confusion matrices.

One problem is that the above supervised LSI techniques do not take into account useful background knowledge from unlabeled text documents. This reduces a chance to improve the performance of LSI in text categorization. After literature review, we found that [11] has incorporated background knowledge into LSI to aid classifying text categories. However, the LSI-based method in [11] does not consider the class label information, which is the most important information, in the text classification.

Hence, we aim at the above mentioned problems and utilize the advantages of LSI to improve the performance of classifying text documents. More specifically, we incorporate the sprinkling terms, which can capture relationships between classes, into original LSI and expand the term-by-document matrix by using the background knowledge from unlabeled text documents. We then validate the proposed method through detailed experiments on three text datasets from the CMU text mining group. The results show that our method can truly improve classification accuracy.

The rest of this paper is organized as followings: In Section 2, we give a brief introduction to the LSI-based techniques. In Section 3, we detail the procedure of the proposed new sprinkled LSI with background knowledge method. In Section 4, we present the experimental setup and results. Finally, the paper is concluded in Section 5.

2 Related Works

2.1 Latent Semantic Indexing

Latent Semantic Indexing [3] is based on the assumption that there is an underlying semantic structure in textual data, and that the relationship between terms and documents can be re-described in this semantic structure form. Textual documents are usually represented as vectors in a vector space. Each position in a vector corresponds to a term. If the term does not appear in the document, the value for the corresponding position equal to 0 and it is positive otherwise.

Hence, a corpus is deemed into a large term-by-document ($t \times d$) matrix X, where, x_{ij} corresponds to the presence or absence of a term (in row i) in a document (in column j), $X_{\cdot j}$ represents the vector of a document j, $X_{i \cdot}$ represents the vector of a word i. The value of x_{ij} can have other expressions, e.g., term frequency (tf), term frequency inverse document frequency (tf-idf), etc.

This matrix, X, is typically very sparse, as most documents consist of only a small percentage of the total number of terms occurred in the full corpus of documents. Due to the nature of text documents, where a word can have ambiguous meanings and each concept can be represented by many different words, LSI reduces the large space to a small space hoping to capture the true relationships between documents by Singular Value Decomposition (SVD), i.e., $X = TSD^{\top}$, where $^{\top}$ is the transpose of a matrix, T and D are orthogonal matrices and S is the diagonal matrix of singular values and the diagonal elements of S are ordered by magnitude. To reduce the dimension, the smallest k values in S can simplified be set to zero. The columns of T and D that correspond to the values of S that were set to zero are deleted. The new product of these simplified three matrices consists of a matrix $\tilde{X}$ that is an approximation of the term-by-document matrix, $\tilde{X} = T_k S_k D_k^{\top}$.

These factors can be deemed as combining meanings of different terms and documents; and documents can be reexpressed using these factors. When LSI is used for retrieval, a query, q, is represented in the same new small space that the document collection is represented in, forming a pseudo-document. This is done by multiplying the transpose of the term vector of the query with matrices T_k and S_k^{-1} [3], $X_q = q^{\top} T_k S_k^{-1}$.

Once the query is represented in this way, the distance between the query and documents can be computed using the cosine metric (or Euclidean distance), which measures similarity between documents. LSI returns the distance between the query and all documents in the collection. Those documents that have higher cosine value (or smaller Euclidean distance) than a given threshold can be returned as relevant to the query. For text classification, k-nearest neighbor (kNN) or Support Vector Machine (SVM) are then applied on the reduced features to get the decision rule or boundary.

2.2 Sprinkling

In [1], a sprinkling LSI was developed by augmenting artificial terms based on class labels in the term-by-document matrix. As the usual manner of LSI, SVD is then performed on the augmented term-by-document matrix. Noisy dimensions corresponding to small singular values are deleted and a low rank approximated matrix is constructed. When in the test phase, in order to make the training document representations compatible with test documents, the sprinkled dimensions are removed. Standard classification methods, e.g., kNN and SVM are then applied in the new represented training features. The inherent reason for this method is that the sprinkled term can add contribution to combine the class information of text documents in the classification procedure.

Further, in [2], an adaptive sprinkling LSI is proposed to adjust the number of sprinkled terms based on the complexity of the class decision boundary and the classifier's classification ability. The number of sprinkled terms for each class is then determined by the confusion matrices: when it is hard to distinguish two classes, it induces more errors in the confusion matrix. In this case, more sprinkling terms are introduced for those classes.

3 Our Approach

In this section, we detail our proposed method on how to incorporate the class information and unlabeled information from background knowledge into the LSI technique. Fig. 1 gives a demonstration.

Assume that there are two classes documents, each consisting of two documents and two other unlabeled documents. We then extend the term-by-document matrix, which is represented by term frequency, by adding sprinkling terms, where we set 1 when the documents are from the same class, zero when they are not in the same class or no label information is given. We then perform the SVD method and obtain an approximation of the term-by-document matrix, X_k, which is reconstructed by selecting the two largest values in S, as shown in Fig. 1. It is easy to see that the approximate matrix has reconstructed the value distribution of data.

Hence, we summarize the procedure as follows:

1. Construct a term frequency matrix, $X \in \mathbb{R}^{f\times(N_L+N_U)}$, by using labeled and unlabeled data, where f is the number of words, N_L and N_U corresponds to the number of labeled and unlabeled documents, respectively;
2. Expand X to a sprinkled matrix, $X_e \in \mathbb{R}^{(f+d)\times(N_L+N_U)}$, by adding sprinkling terms with ones for the labeled data and terms with zeros for the unlabeled data, where d is the length of sprinkling terms;
3. Perform Singular Value Decomposition (SVD) on the expanded matrix, i.e., $X_e = TSD^\top$, where $T \in \mathbb{R}^{(f+d)\times(f+d)}$ is an orthogonal matrix, $S \in \mathbb{R}^{(f+d)\times(N_L+N_U)}$ is a diagonal matrix, and $D \in \mathbb{R}^{(N_L+N_U)\times(N_L+N_U)}$ is another orthogonal matrix;

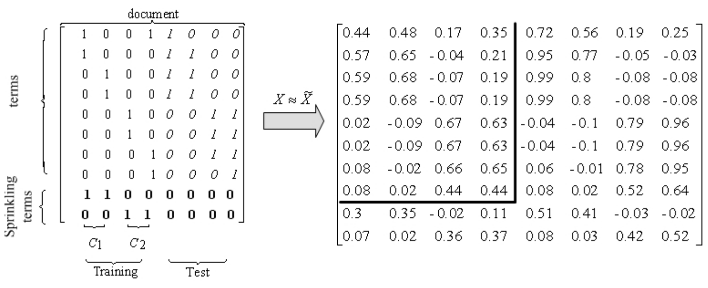


Fig. 1. An example of the term-by-document matrix transforming by Sprinkling LSI with background knowledge

4. Determine columns of T and D corresponding to the k largest eigen values in S and discard the additional sprinkling terms. Hence, finally, $T_k \in \mathbb{R}^{f \times k}$, $S_k \in \mathbb{R}^{k \times k}$, a diagonal matrix, and $D_k \in \mathbb{R}^{k \times N_L}$;
5. Transform training data to new features, D_k, and for a test data, $q \in \mathbb{R}^f$, the corresponding new feature is calculated by $\tilde{q} = q^\top T_k S_k^{-1} \in \mathbb{R}^k$. The length of new features becomes shorter since $k \ll f$;
6. Perform classification algorithms on transformed new features.

4 Experiments

4.1 Experimental Setup

In this section, we evaluate the standard LSI, sprinkled LSI (SLSI), LSI with background knowledge (LSI-bg) and our proposed sprinkled LSI with background knowledge (SLSI-bg) on three datasets: the WebKB dataset, the 20 newsgroups dataset and the industry sector (IS) dataset, from CMU text mining group [1].

Test is performed in the transductive mode: test data are considered as unlabeled data and used for the test. Accuracy [2] is used to evaluate the performance of text classification on kNN with two different metrics, the Euclidean distance (kNNE) and the cosine similarity (kNNC).

Before performing classification, text processing, skipping headers, skipping html, etc., is run by the rainbow package [2]. The option of whether or not using stoplist is detailed in the following data sets description. Further, we remove the words that only occur once and choose the top 1000 with highest mutual information gain [2].

4.2 Data Sets

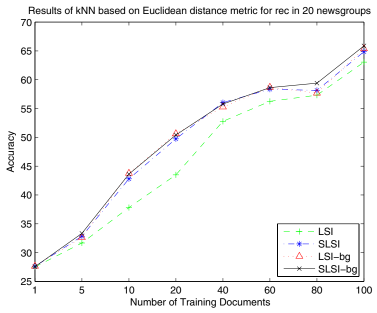
Three standard text documents data sets are tested in the experiments:

20 newsgroups dataset: This data set is a collection of 20, 000 UseNet news postings into 20 different newsgroups [7] including seven sub-trees: alt, comp, rec, sci, soc, talk and misc. We use rec sub-tree which consists of 4 classes in the experiment. 500 documents from each class are selected [2]. Words in the standard SMART stoplist is removed from the dataset. Finally, this dataset forms a 1000×2000 term-by-document matrix.

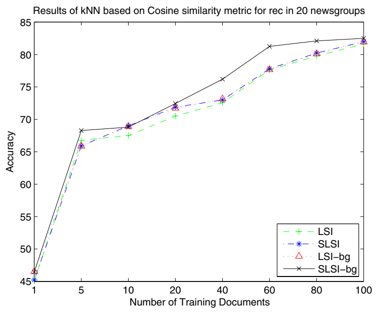
WebKB dataset: It is the 4 universities data set from the World Wide Knowledge Base (WebKb) project of the CMU text learning group [3]. There are 8, 282 pages were manually classified into the following seven categories: student, faculty, staff, department, course, project and other. Here, again, each class consists of 500 documents. Hence, the categories of staff and department do not use in the experiment. Stoplist option is not used in this dataset. The dataset forms a 1000×2500 term-by-document matrix.

[1] http://www.cs.cmu.edu/~TextLearning/datasets.html
[2] http://www.cs.umass.edu/~mccallum/bow/rainbow/
[3] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

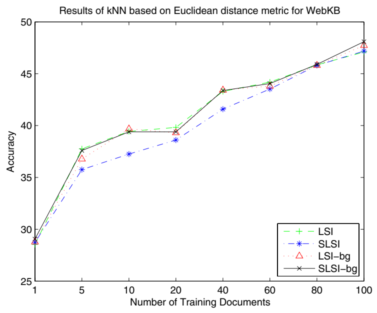
Industry sector dataset: This dataset consists of 7 industry sector information: basic materials, energy, finance, healthcare, technology, transportation and utilities. Here, again, we choose 500 documents for each class. Hence, only categories of basic materials, finance, technology and transportation are considered in the experiment. Stoplist option is used in this dataset. Finally, the dataset consists of a 1000×2000 term-document matrix.



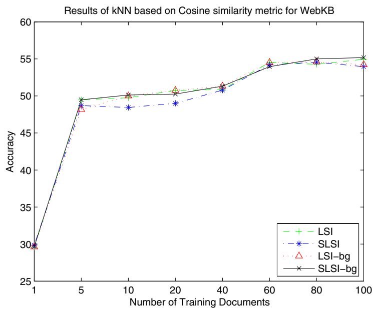
(a) kNNE for rec



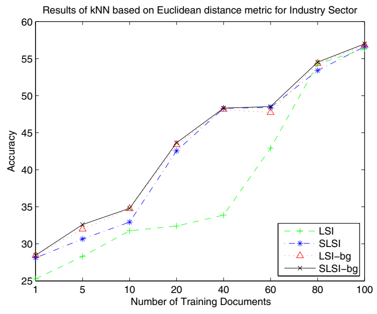
(b) kNNC for rec



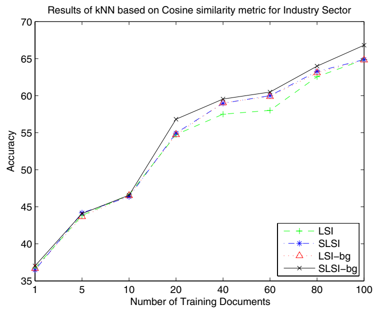
(c) kNNE for WebKB



(d) kNNC for WebKB



(e) kNNE for IS



(f) kNNC for IS

Fig. 2. Results of kNN based on two different distance metrics for rec, WebKB, and IS datasets

4.3 Experimental Results

In the experiments, we let the number of labeled data be 1, 5, 10, 20, 40, 60, 80 and 100 to test the effect of the number of training data. In classification using kNN method, features generated by LSI, SLSI, LSI-bg and SLSI-bg are used.

The length of sprinkling term for each class is set to 4 as [1]. In the kNN, k is set to 1. Results are performed on 10 runs to get the average accuracies.

Fig. 2 presents the results of kNNE and kNNC on the rec sub-tree of 20 newsgroups, the WebKB, and the industry sector datasets, respectively. From the results, we have the following observations:

- As the number of labeled training samples increases, the accuracies for all four features, LSI, SLSI, LSI-bg, SLSI-bg increase correspondingly. SLSI-bg is the best overall for all three datasets.
- The performance usually has a large increase when the number of training sample increase from 1 to 5, especially using the kNNC method. For example, when using the kNNC method, it is over 41% improvement on the rec subtree and over 70% improvement on other two datasets.
- The performance of kNNC outperforms that of kNNE greatly for all three datasets. Especially, the improvement using the same feature is obviously significant, at least 20% improvement, for the rec subtree in 20 newsgroup dataset. Even for the WebKB dataset, although the improvement is not large when the number of labeled data is 1, the performance increases significantly when the number of labeled data is larger than 1.
- For the industry sector dataset, it is shown that LSI is rather worse in this dataset under the kNNE metric, especially when the number of training samples equals 20, 40, 60; while other three features can get relative better results. This means that other features can help for the classification procedure.

5 Conclusions

In this paper, we consider two problems: the existence of poylsemy and synonym, and the huge useful unlabeled text documents, in text classification. We propose a novel Latent Semantic Indexing expression which utilizes the class relationship and background knowledge embedding in text documents. We perform detailed experimental comparisons to test the effects of class relationship and background knowledge. Experimental results demonstrate our proposed method is promising in text classification.

There are still several works need to be considered. For example, why and how this combination will help increase the performance of text classification. Developing theoretical framework for analyzing the difference between our method and original LSI and providing theoretical guidance are significant works.

Acknowledgments

The work described in this paper was fully supported by the grants from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 4125/07E).

References

1. Chakraborti, S., Lothian, R., Wiratunga, N., Watt, S.: Sprinkling: Supervised latent semantic indexing. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 510–514. Springer, Heidelberg (2006)
2. Chakraborti, S., Mukras, R., Lothian, R., Wiratunga, N., Watt, S., Harper, D.: Supervised latent semantic indexing using adaptive sprinkling. In: Proceedings of IJCAI 2007, pp. 1582–1587 (2007)
3. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Chichester (2000)
5. Dumais, S.T.: Using lsi for information filtering: Trec-3 experiments (1995)
6. Gee, K.R.: Using latent semantic indexing to filter spam. In: Proceedings of the 2003 ACM symposium on Applied computing, pp. 460–464 (2003)
7. Lang, K.: NewsWeeder: learning to filter netnews. In: Proceedings of ICML 1995, pp. 331–339. Morgan Kaufmann publishers Inc., San Mateo (1995)
8. Liu, T., Chen, Z., Zhang, B., Ma, W.y., Wu, G.: Improving text classification using local latent semantic indexing. In: Proceedings of ICDM 2004, pp. 162–169 (2004)
9. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
10. Sun, J.-T., Chen, Z., Zeng, H.-J., Lu, Y.-C., Shi, C.-Y., Ma, W.-Y.: Supervised latent semantic indexing for document categorization. In: Perner, P. (ed.) ICDM 2004. LNCS, vol. 3275, pp. 535–538. Springer, Heidelberg (2004)
11. Zelikovitz, S., Hirsh, H.: Using LSI for text classification in the presence of background text. In: Paques, H., Liu, L., Grossman, D. (eds.) Proceedings of CIKM 2001, pp. 113–118 (2001)