Incorporating Latent Semantic Indexing into Spectral Graph Transducer for Text Classification

Xinyu Dai[1], Baoming Tian[1], Junsheng Zhou[2], Jiajun Chen[1]
[1]State Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, 210093, China
[2]Department of Computer Science, Nanjing Normal University, 210097, China

Abstract

Spectral Graph Transducer(SGT) is one of the superior graph-based transductive learning methods for classification. As for the Spectral Graph Transducer algorithm, a good graph representation for data to be processed is very important. In this paper, we try to incorporate Latent Semantic Indexing(LSI) into SGT for text classification. Firstly, we exploit LSI to represent documents as vectors in a latent semantic space since we propose that the documents and their semantic relationships can be reflected more pertinently in this latent semantic space. Then, a graph needed by SGT is constructed. In the graph, a node corresponds to a vector from LSI. Finally, we apply the graph to Spectral Graph Transducer for text classification. The experiments gave us excellent results on both English and Chinese text classification datasets and demonstrated the validation of our assumption.

Introduction

Over the recent years, text classification has attracted more and more attention due to its wide applicability. Many supervised classifiers(Fabrizio 2002), such as Naïve Bayes, K-Nearest Neighbor(KNN) and Support Vector Machine(SVM) from machine learning community have been applied to text classification.

However, these supervised learning approaches are not effective enough when a large number of labeled training examples are not available. So, some semi-supervised learning algorithms have been applied to text classification using a small set of labeled data and many unlabeled data. These approaches use Expectation Maximization to estimate posteriori parameters for Naïve Bayes Classifier (Nigam *et al.* 2000), use transductive inference for SVM (Vapnik 1998)(Joachims 1999), and use the co-training algorithm (Blum & Mitchell 1998)(Nigam & Ghani 2000).

Recently, due to the fine properties of spectral graph theory(Chung 1997), some graph-based semi-supervised learning methods (Blum & Chawala 2001)(Zhu, Ghahramani, & Lafferty 2003)(Blum *et al.* 2004) have been proposed and applied to text classification. Data should be firstly represented as a graph in the graph-based methods. Then these methods can exploit the structure information in the graph to find optimal cut as classification. The graph-based methods start with a graph where nodes are the labeled and unlabeled data, and weighted edges reflect the similarities between nodes. For the text classification task, all documents are represented as a graph in which a node represents a document, and a weighted edge means the similarity between two documents. The good text classification result has been obtained by Spectral Graph Transducer, a superior graph-based semi-supervised learning algorithm(Joachims 2003).

Although the good results have been obtained by Spectral Graph Transducer, the text representation used by Joachims is also based on "bag of words", where a document is represented as a set of words appearing in it. In such a word-based feature space, the similarity cannot reflect the semantic relationship between documents since the latent associations between words are ignored.

In any graph-based methods, the graph construction is the first and also a very important step (Zhu 2005) where a good graph should be constructed in a given feature space to represent the data with domain knowledge, and further to represent their distributions and relationships.

Latent semantic indexing (LSI) (Deerwester *et al.* 1990) is a technique widely used in Information Retrieval community(Deerwester *et al.* 1990)(Kumar & Srinivas 2006). It is an automatic method that represents the documents in a new reduced semantic space. LSI may find the latent semantic structure between the words and the documents in a document collection. LSI is especially useful in combating polysemy(one word can have different meanings) and synonymy(different words are used to describe the same concept), which can make classification task more difficult(Zelikovitz & Hirsh 2003). Through using LSI, documents that do not share any words can still be close to each other if their words are semantically related.

In this paper, we propose a text classification approach by introducing LSI into SGT for text classification. In our proposed method, all documents are firstly represented as vectors using Vector Space Model. Then we use LSI to refine the vectors in a semantic space and construct a graph in which the nodes correspond to the vectors from LSI. This graph can better reflect the documents and their semantic relationships. Finally, we applied this graph to Spectral Graph Transducer for text classification. The results of our experiments demonstrate that by incorporating LSI into Spectral Graph Transducer, we can improve the performance of text classification significantly.

The rest of this paper is organized as follows. We first briefly describe the Spectral Graph Transducer. Then LSI is introduced to represent the documents as refined vectors. Further, we propose to incorporate LSI into Spectral Graph Transducer method for the text classification task. After that, we describe our experiments and report the results. Finally, the conclusion and future work are given.

Spectral Graph Transducer

Spectral Graph Transducer (SGT) is a graph-based transductive learning method and is firstly introduced in (Joachims 2003). In this section, we'll briefly present the main ideas of this algorithm for a complete description of our method.

As opposed to inductive learning, transductive learning is defined by Vapnik(Vapnik 1998) to tackle the problem of learning from small training samples. In (Vapnik 1998), the transductive SVM is proposed. It tries to find a maximum margin as a separating hyperplane for both labeled and unlabeled data, and at the same time assigns labels to all the unlabeled samples. In the transductive setting, many graph-based methods, such as s-t mincut(Blum & Chawala 2001), Gaussian Field(Zhu, Ghahramani, & Lafferty 2003), Spectral Graph Transducer(Joachims 2003) and randomized mincut(Blum *et al.* 2004), were designed to address semi-supervised learning. Graph-based transductive learning methods can exploit the structure and distribution information within a graph to make exact classification.

In the s-t mincut algorithm (Blum & Chawala 2001), classification was transferred as a bi-partition task, that is, to find a cut which divides a graph into two sub-graphs. The partition objective is to minimize the cut-value, the sum of the edge-weights across the cut. The cut value is calculated as following

$$cut(G^+, G^-) = \sum\nolimits_{y_i y_j = -1} A_{ij}$$

For the s-t mincut algorithm, the graph G is firstly constructed, where the nodes represent labeled and unlabeled examples, and the edge weight A_{ij} denotes similarities between neighboring examples. All A_{ij} can be represented as an adjacent matrix A of the graph. The partitioning process assigns labels to unlabeled examples by cutting G into two subgraphs G^- and G^+, and tags all examples(nodes) in $G^-(G^+)$ with $y_i = -1$ and $y_i = +1$.

While the s-t mincut algorithm seems to be a good solution, it can easily lead to degenerated cuts, thereby producing a biased partition(Joachims 2003). To overcome this problem, Joachims proposed to use the ratio-cut (Hagen & Kahng 1992), where the goal of the graph partition became

$$\begin{aligned} &\min\nolimits_{\vec{y}} \tfrac{cut(G^+,G^-)}{|\{i:y_i=+1\}||\{i:y_i=-1\}|} \\ &s.t.\ y_i = 1,\ if\ x_i\ is\ positive \\ &\quad\ y_i = -1,\ if\ x_i\ is\ negative \\ &\quad\ \vec{y}\ \in \{-1,+1\}^n \end{aligned}$$

where x_i is a sample in the training data or testing data. The denominator is the product of the number of positive and the number of negative in both training and testing data, which can help to get more balanced cut. The prediction vector $\vec{y}$ is searched for minimum the ratio-cut value. And in the transductive setting, there is a constraint that nodes in training examples must lie in $G^-(G^+)$ according as their known positive(negative) labels.

The ratio-cut solution is known to be NP-Hard (Shi & Malik 2000). Fortunately, spectral methods can efficiently give a good approximation to the solution. Based on the ratio-cut and spectral solution, Joachims (Joachims 2003) exploited the ratio of the positive and negative samples in the training data, and generalized the constraint ratio-cut problems via spectral methods.

SGT algorithm avoids the degenerated cut in graph-based transductive learning. It has been used for speech recognition, digital recognition and text classification by Joachims, and appears to be one of the superior transductive learning algorithms(Joachims 2003).

From the above introduction to SGT, we can see that the construction of a suitable graph G is one of the key points to this method. The classification result generated by SGT depends on the graph represented as the adjacent matrix A. A matrix A which represents the real relationships among the data can help SGT generate the optimal classification. However, few studies focused on how to construct a good graph for Spectral Graph Transducer. This paper focuses on how to construct a better graph for the documents in which each weighted edge (A_{ij}) can reflect the semantic similarity between documents, so that the graph as an adjacent matrix A can help SGT generate better classification result.

Text Representation and Latent Semantic Indexing

As for text representation as a graph, each node corresponds to a document, and each edge means the relationship between two documents. Usually, as a node in the text graph, a document is represented as a vector by Vector Space Model.

Vector Space Model

In VSM, each document d_j is expressed by a weight vector $\vec{d_j} = (w_{1j}, w_{2j}, ...w_{tj})^T$, where t is the dimension of the word-based space, and w_{ij} is the weight or importance of the word i in the representation of the document $\vec{d_j}$. Usually, the weight w_{ij} is given by word frequency(tf) and inverse document frequency(idf). All of n documents is then represented by a term-document matrix $X = (\vec{d_1}, ..., \vec{d_i}, ..., \vec{d_n})$ with t rows and n columns. The similarity between two documents can be computed by cosine measure

$$sim_{vsm}(\vec{d_i}, \vec{d_j}) = \frac{\sum\limits_{z=1}^{t} w_{zi} w_{zj}}{\sqrt{\sum\limits_{z=1}^{t} w_{zi}^2}\sqrt{\sum\limits_{z=1}^{t} w_{zj}^2}}$$

where it implies that two documents are considered similar only when they contain the same words. But to the nature of text, this is not true. Many documents that are related to each other semantically might not share many common words, and sometimes documents not related to each other might share some common words. This ascribes to the subtleties of natural language, where the same concept can be represented by many different words (synonymy), and words can have various meanings (polysemy). In VSM of word-based feature space, the latent relationships between words are ignored, thus relationship between documents represented by VSM may not be accurate. Fortunately, Latent Semantic Indexing technique provides a good solution to capturing the semantic relationships between documents.

Latent Semantic Indexing(LSI)

Latent Semantic Indexing (Deerwester *et al.* 1990), which is firstly used in Information Retrieval, is supposed to be able to map the documents into a "semantic space" implicitly, so that it can capture the semantic relationships between documents pertinently.

LSI starts with the term-document matrix. Singular value decomposition (SVD) can be used to analyze the term-document matrix to derive the latent semantic structure in it. The vectors representing documents are then projected into a new, low-dimensional subspace obtained by truncated SVD. Using SVD, the term-document matrix X can be decomposed into the product of three other matrices $X = T_o S_o D_o^T$, where T_0 and D_0 are the matrices of left and right singular vectors and S_0 is the diagonal matrix of singular values. The diagonal elements of S_0 are ordered by magnitude, and the first k largest values can be selected as a means of developing a "latent semantic" representation of the original matrix X. Through truncated SVD, approximation to X can be created $X \approx \widehat{X} = T_k S_k D_k^T$, where T_k is the $t * k$ word-concept matrix, S_k is the $k * k$ concept-concept matrix, and D_k is the $k * n$ concept-document matrix.

Using SVD, latent semantic structure in the documents can be detected, which can be viewed as semantic knowledge in this specific domain(Zelikovitz & Hirsh 2003). Then the original documents are represented as the refined vectors in the k reduced dimensional latent semantic space. Further, the refined graph for all documents can be constructed with nodes(the refined vectors) and the weighted edges(similarities between refined vectors).

Incorporate LSI into SGT

Spectral Graph Transducer is one of the superior methods for text classification. However, different graph representation of text data may influence the result generated by SGT. We have proposed that we can use LSI to construct a refined graph of text data for SGT. In this section, we present how to incorporate LSI into Spectral Graph Transducer method for text classification. Firstly, all documents are represented as vectors using Vector Space Model. Then we use LSI to refine the vectors in a semantic space and we construct a graph in which the nodes correspond to the vectors from LSI. We think this graph can better reflect the documents and their real semantic relationships. Finally, we applied Spectral Graph Transducer to this graph to perform text classification.

Considering that during the semantic structure extraction process, LSI does not deal with the labels of the given training samples at all. So, LSI can be used to analyze the term-document matrix combining the training data and testing data together, instead of the matrix only from the training data. This larger matrix may provide us more reliable latent semantic structure in all the given data (Zelikovitz & Marquez 2005).

Now we are ready to integrate LSI into SGT for text classification. Note that for SGT, we need to construct an adjacent matrix as a graph. And each element in the matrix means the similarity between two documents(represented as vectors). After normalization for each column of $\widehat{X}$, the similarity matrix between documents can be directly obtained by $\widehat{A} = \widehat{X}^T \widehat{X} = DS^2D^T$.

As SGT is a binary classifier, in order to use SGT to do multi-class text classification, we use one-against-all classifiers, i.e., one SGT classifier for each class.

With LSI, the whole algorithm description is as follows:

1. Preprocess the text classification datasets. For English, do word stemming and stop-word removal. And for Chinese, segmentation is needed.
2. Construct a $t * d$ term-document matrix X with t terms and d documents for all training and testing documents.
3. Carry out SVD on X ($[T_0, S_0, D_0] = SVD(X)$), select top k largest values in S_0 and get the corresponding matrices T_k S_k and D_k.
4. Construct a weighted undirected graph as adjacency matrix $\widehat{A}$.
5. Compute the Laplacian matrix L of $\widehat{A}$ and get the smallest 2 to $d+1$ eigenvalue and eigenvector of L.
6. With the constraint ratio calculated from the number of positive and negative examples in training data, exploit spectral method to get the prediction vector.
7. According to the prediction vector, get hard class assignment for each testing document.

The detailed description for SGT algorithm of step 5-7 can be referred to (Joachims 2003).

Experiments

In this section, in order to evaluate our method, we conduct experiments on two datasets Reuters21578(Lewis) and TanCorp-12 (Tan) and report the results.

Dataset

The Reuters21578 is the most commonly used dataset for English text classification. In this dataset, each document is labeled with at least one of the 135 possible categories. In our experiments, after ModApte Split(Fabrizio 2002) and omitting empty documents, we use a total number of 8986 documents from the 10 most frequent categories, where 2532 are used as the training data and 6454 are used as the testing data . After stemming and stop-word removal, 21726 words are kept, and we also use $tf * idf$ for word weighting.

The second dataset is TanCorp-12, which is a Chinese language text classification corpus. There are 14150 documents belong to 12 topics in this dataset. We randomly select 20 percent of documents in each topic as training data, and the remains as testing data. There are totally 72603 Chinese words as features. Words weighting are also calculated by $tf * idf$.

Performance Measure

We use $F1$ measure to evaluate the performance on these two dataset, $F1$ measure is defined as $F1 = 2pr/(p + r)$ where p is precision and r is recall. The precision is the percentage of predicted documents for the given topic that are correctly classified, and the recall is the percentage of total documents for the given topic that are correctly classified. On multi-class dataset, micro- and macro-average $F1$ are also important measures to evaluate overall performance across the entire dataset. In $microF1$, the performances of all topics are added and the overall recall and precision are computed. In $macroF1$, the performance measures are calculated separately for each topic, and the mean of the resulting performance values is taken. From the calculations, we can see that $microF1$ and $macroF1$ emphasize the performance of the system on common and rare categories respectively.

Comparison and Discussion

In the following, we report the experimental results of using LSI for SGT. As a baseline for comparison the results of SVM and SGT(without LSI included) are shown. For SVM we employed LibSVM (Chang & Lin 2001). We use the binary approach provided by LibSVM toolkit to perform one-against-all multi-class classification. When running libsvm, we use RBF kernel, and in order to get optimal parameters and better performance, five fold cross-validation and grid search are used. We run SGT^{light} from the website http://sgt.joachims.org/. Parameters for SGT^{light} were set to default values, that is, $k = 50$ and $d = 100$. And for LSI, we use SVDLIBC package which can be downloaded from http://tedlab.mit.edu/ dr/SVDLIBC/ to obtain latent semantic structure. Considering the efficiency, we transfer the term-document matrix into st(sparse text) format which is defined in the package as the input of SVD operation.

Table 1 and Table 2 demonstrate that by incorporating LSI into SGT method, we can improve the performance on both English and Chinese text classification datasets. Note that on these two datasets, we just set the value of k in truncated SVD to 50(We can see it may not be optimal from Figure1).

Table 1 shows the performance on each topic and the $macroF1$/$microF1$ measurements on the whole selected ten categories on Reuters21578 dataset. When enough training samples are available, such as those on topics of Earn and Acq, good results are achieved by all these three algorithms. However,on most of the ten topics, no matter the training samples are small or large enough, SGT performs better than SVM. Further on most of the ten topics, LSI+SGT significantly outperforms other approaches. Moreover the results of the $macroF1$ and $microF1$ measures demonstrate that we can significantly improve text

Table 1: Dataset size and performance on Reuters21578

Topics	Dataset Size		F1-measure		
	Training	Testing	SVM	SGT	LSI(k=50)+SGT
Earn	1080	2861	88.18%	93.11%	**97.57%**
Acq	718	1648	**94.39%**	74.47%	93.38%
Money-fx	179	534	55.21%	80.10%	**84.33%**
Crude	186	385	7.50%	**91.50%**	90.66%
Grain	148	428	55.86%	92.71%	**92.98%**
Trade	116	367	66.88%	83.94%	**88.56%**
Interest	131	345	17.86%	**75.84%**	71.25%
Wheat	71	211	45.95%	62.57%	**64.37%**
Ship	87	191	12.29%	75.93%	**82.76%**
Corn	56	180	12.75%	65.04%	**72.82%**
macro-F1	2532	6454	45.70%	79.52%	**83.87%**
micro-F1	2532	6454	77.17%	84.05%	**91.65%**

Table 2: Dataset size and performance on TanCorp-12

Topics	Dataset Size		F1-measure		
	Training	Testing	SVM	SGT	LSI(k=50)+SGT
Talent	121	487	75.12%	**90.58%**	89.10%
Sport	561	2244	97.9%	**99.17%**	98.84%
Sanitation	281	1125	**88.72%**	83.90%	84.92%
Clime	30	120	13.89%	75.69%	**78.68%**
Entertainment	300	1200	75.68%	**82.45%**	76.33%
House	187	748	90.94%	**97.40%**	94.90%
Education	161	647	**78.75%**	72.96%	77.17%
Car	118	472	68.06%	76.24%	**87.97%**
Computer	588	2355	90.09%	95.72%	**97.29%**
Science	208	832	**72.92%**	58.98%	67.92%
Art	109	437	45.08%	**60.62%**	51.35%
Economic	163	656	**69.98%**	57.88%	69.79%
macro-F1	2827	11323	72.26%	79.30%	**81.19%**
micro-F1	2827	11323	83.28%	85.65%	**86.68%**

classification results by combining LSI and SGT. Table 2 shows the results on TanCorp-12 dataset. Although there is no significant improvement on some topics, the best results of macro-F1 and micro-F1 are still achieved by LSI+SGT.

For LSI, the choice of k for truncated SVD is very important. In order to evaluate the influence of k value on LSI, we explored the possible values for $k = 50/100/150/200$ on the two datasets. Figure1 illustrates the results of using different k. We found that when k is set to 50, we achieve the best results in most cases while the best $microF1$ on TanCorp12 is obtained when k is set to 100. The research on analyzing the detailed relationship between value of k and the classification results is one of our future research directions.

Conclusion

In this paper, we proposed an approach to improve the performance of text classification by incorporating LSI into Spectral Graph Transducer. We performed LSI to map the original document representation into that in a latent semantic space. The refined document representation is considered to contain semantic knowledge in this specific domain, and consequently reflect the documents and their semantic relationships more pertinently. The experiment was conducted on both English and Chinese language text classifica-

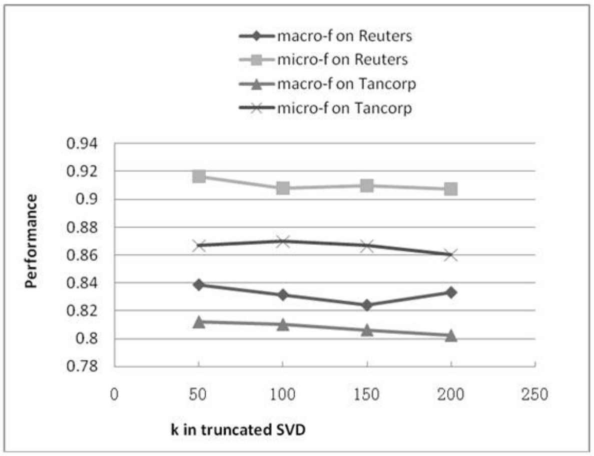

Figure 1: Influence of k in LSI on two datasets

tion datasets, and the results demonstrated the effectiveness of combining LSI with SGT.

LSI and SGT are both based on spectral analysis techniques. In LSI, term-document matrix is analyzed to represent the documents in a latent semantic subspace. While in SGT, spectral method is used on document-document matrix to find the partition among documents more exactly. From this paper, we find that they are seemly complementary to each other in text classification task. The analysis on the deep relationship between these two techniques is an interesting future issue.

Acknowledgments

This work is supported by the NSFC (60673043) and the National High Technology Development Program(863 program) of China (2006AA01Z143, 2006AA01Z139).

References

Blum, A., and Chawala, S. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of 18th International Conference on Machine Learning*, 19–26. Morgan Kaufmann, San Francisco, CA.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of 11th annual conference on Computational learning theory*, 92–110. Morgan Kaufmann, San Francisco, CA.

Blum, A.; Lafferty, J.; Rwebangira, M.; and Reddy, R. 2004. Semi-supervised learning using randomized mincuts. In *Proceedings of the 21st International Conference on Machine Learning.*

Chang, C., and Lin, C. 2001. Libsvm: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/.

Chung, F. R. 1997. *Spectral Graph Theory*. American Mathematical Society.

Deerwester, S.; Dumais, S.; Thomas, K.; George, W.; and Richard, A. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* Volume of:41-6.

Fabrizio, S. 2002. Machine learning in automated text classification,. *ACM Computing Surveys* Volume of:34-1.

Hagen, L., and Kahng, A. 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on CAD* 1074–1085.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International conference on Machine Learning*, 200–209.

Joachims, T. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of the 20th international conference on machine learning*.

Kumar, C., and Srinivas, S. 2006. Latent semantic indexing using eigenvalue analysis for efficient information retrieval. *Applied mathematics and computer science* 16-4:551–558.

Lewis, D. Reuters-21578 text categorization test collections,http://www.daviddlewis.com/resources/testcollections/.

Nigam, K., and Ghani, R. 2000. Analyzing the effectiveness and applicability of co-training. In *Proceedings of 9th international conference on Information and knowledge management*, 86–93.

Nigam, K.; McCallum, K.; Sebastian, T.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using em. *Machine Learning* 39:103–134.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. PAMI.

Tan, S. Chinese text classification dataset - tancorpv1.0. http://www.searchforum.org.cn/tansongbo/corpus.htm.

Vapnik, V. 1998. *Statistical Learning Theory*. Wiley.

Zelikovitz, S., and Hirsh, H. 2003. Integrating background knowledge into text classification. In *Proceedings of international joint conference of artificial intelligence*.

Zelikovitz, S., and Marquez, F. 2005. Transductive learning for short-text classification problems using latent semantic indexing. 143–163. IJPRAI.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*.

Zhu, X. 2005. *Semi-supervised learning with graphs*. Ph.D. Dissertation, Carnegie Mellon University.