Adaptive Label-Driven Scaling for Latent Semantic Indexing

Xiaojun Quan[1,2], Enhong Chen[1,2], Qiming Luo[1,2], Hui Xiong[3]
1 MOE-MS Key Laboratory of Multimedia Computing and Communication, USTC
2 Department of Computer Science, University of Science and Technology of China(USTC)
3 MSIS Department, Rutgers University
E-mail: cheneh@ustc.edu.cn; hxiong@rutgers.edu

ABSTRACT

This paper targets on enhancing Latent Semantic Indexing (LSI) by exploiting category labels. Specifically, in the term-document matrix, the vector for each term either appearing in labels or semantically close to labels is scaled before performing Singular Value Decomposition (SVD) to boost its impact on the generated left singular vectors. As a result, the similarities among documents in the same category are increased. Furthermore, an adaptive scaling strategy is designed to better utilize the hierarchical structure of categories. Experimental results show that the proposed approach is able to significantly improve the performance of hierarchical text categorization.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Classifier design and evaluation; I.2.7 [**Artificial Intelligence**]: Text analysis

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Latent Semantic Indexing (LSI) [1] is an important technique for text retrieval and categorization, and a statistical framework for LSI has been established [2]. Several approaches for improving its performance has been proposed, such as adaptive sprinkling [3], model averaging [4], among others. In this study, we propose a novel approach to enhance Latent Semantic Indexing (LSI) by exploiting category labels. Specifically, in the term-document matrix, vectors for terms either appearing in category labels or semantically close to category labels are scaled before performing Singular Value Decomposition (SVD) to boost their impact on the generated left singular vectors. As a result the similarities among documents in the same category are increased. Furthermore, an adaptive scaling strategy is designed to better exploit the hierarchical structure of categories in hierarchical text categorization. A comparative experimental study on two typical text data sets demonstrates the significant performance enhancement of our approach for hierarchical text categorization.

2. METHODOLOGY

In text categorization, terms in a document that also appear in category labels are more effective in categorizing the document than other terms. Therefore, it is desirable to design a strategy to boost their impact. Motivated by this intuition, we propose to scale the term vectors of category labels in the term-document matrix before performing Singular Value Decomposition (SVD). Formally, a term vector $\boldsymbol{t}$ is scaled as $\hat{\boldsymbol{t}} = (1+q)\boldsymbol{t}$, where q is a positive real number.

Given that there are a number of terms that are semantically similar to labels, it is natural to extend the scaling to such terms. We refer to terms either appearing in category labels or semantically close to category labels as *label-relevant* terms. Formally, for a term t appearing in labels, the corresponding set of *label-relevant* terms is defined as

$$label\text{-}relevant(t) = \{s | rank(sim(s,t)) \leq l\} \qquad (1)$$

where $sim(s,t)$ (the similarity between s and t) is defined to be the (s,t) entry in the term-term similarity matrix XX^T based on LSI: $X = U_r \Sigma_r V_r^T$, $XX^T = U_k \Sigma_k^2 U_k^T$, and the $rank$ operator is applied to select the l terms closest to t.

We will show that our *label-driven scaling* method increases the similarity of a query with a document of the same category by making two assumptions. Let $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ denote the query vector, and $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ refer to the vector of a training document belonging to the same category as the query. Without loss of generality, assume that the first k components of the vectors correspond to *label-relevant* terms, which are scaled by a factor t. After scaling, the document vector is denoted as $\boldsymbol{y'} = (y'_1, y'_2, ..., y'_n)$.

The cosine distance between the two vectors $dist(t)$ is $dist(t) = \frac{\sum_{i=1}^{n} x_i y'_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} {y'_i}^2}} = \frac{t\sum_{i=1}^{k} x_i y_i + \sum_{i=k+1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{t^2\sum_{i=1}^{k} y_i^2 + \sum_{i=k+1}^{n} y_i^2}}$. To simplify the notation, the following abbreviations are introduced: $C_1 = \sum_{i=1}^{k} x_i y_i$, $C_2 = \sum_{i=k+1}^{n} x_i y_i$, $T = \sqrt{\sum_{i=1}^{n} x_i^2}$, $S_1 = \sum_{i=1}^{k} y_i^2$, $S_2 = \sum_{i=k+1}^{n} y_i^2$. Then, we have $dist(t) = \frac{tC_1 + C_2}{T\sqrt{t^2 S_1 + S_2}}$. If we take the derivative of $dist(t)$ with respect to t, it can be verified that $dist'(t) = \frac{C_1 S_2 - C_2 S_1 t}{T(t^2 S_1 + S_2)^{3/2}}$.

It is evident that if making either one of the following two

Copyright is held by the author/owner(s).
SIGIR'08, July 20–24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

Table 1: Experimental results for Reuters-21578 Tree (a)

Category	Precision (%)				Recall (%)				F1 (%)			
	K-LSI	NADP	SLSI	SVM	K-LSI	NADP	SLSI	SVM	K-LSI	NADP	SLSI	SVM
corn	86.21	*89.66*	**90.00**	96.88	73.53	*76.47*	**79.41**	91.18	79.37	*82.54*	**84.38**	93.94
wheat	86.96	*89.13*	**89.13**	93.88	85.11	*87.23*	**87.23**	97.87	86.02	*88.17*	**88.17**	95.83
ship	94.19	*95.40*	**97.65**	95.29	94.19	*96.51*	**96.51**	94.19	94.19	*95.95*	**97.08**	94.74
gas	80.00	*85.29*	**82.86**	83.33	96.55	*100.0*	**100.0**	86.21	87.50	*92.06*	**90.63**	84.75
Micro-Average	88.78	*91.32*	**91.84**	93.37	88.78	*91.33*	**91.84**	93.37	88.78	*91.33*	**91.84**	93.37
Macro-Average	86.84	*89.87*	**89.91**	92.35	87.34	*90.05*	**90.79**	92.36	87.09	*89.96*	**90.35**	92.35

Table 2: Experimental results for Reuters-21578 Tree (b)

Category	Precision (%)				Recall (%)				F1 (%)			
	K-LSI	NADP	SLSI	SVM	K-LSI	NADP	SLSI	SVM	K-LSI	NADP	SLSI	SVM
barley	100.0	*100.0*	**100.0**	92.31	83.33	*83.33*	**91.67**	100.0	90.91	*90.91*	**95.65**	96.00
rice	89.47	*88.89*	**94.44**	95.24	80.95	*76.19*	**80.95**	95.24	85.00	*82.05*	**87.18**	95.24
cocoa	78.26	*75.00*	**81.82**	85.71	100.0	*100.0*	**100.0**	100.0	87.80	*85.71*	**90.00**	92.31
copper	89.47	*94.74*	**100.0**	93.75	94.44	*100.0*	**100.0**	83.33	91.89	*97.30*	**100.0**	88.24
iron	100.0	*100.0*	**100.0**	100.0	90.91	*90.91*	**100.0**	90.91	95.24	*95.24*	**100.0**	95.24
tin	92.31	*92.31*	**92.86**	84.62	85.71	*85.71*	**92.86**	78.57	88.89	*88.89*	**92.86**	81.48
Micro-Average	89.36	*89.36*	**93.62**	91.49	89.36	*89.36*	**93.62**	91.49	89.36	*89.36*	**93.62**	91.49
Macro-Average	91.59	*91.82*	**94.85**	91.94	89.23	*89.36*	**94.25**	91.34	90.39	*90.57*	**94.55**	91.64

Table 3: Experimental results for 20 Newsgroups

Category	Precision (%)				Recall (%)				F1 (%)			
	K-LSI	NADP	SLSI	SVM	K-LSI	NADP	SLSI	SVM	K-LSI	NADP	SLSI	SVM
auto	76.67	*78.02*	**78.89**	69.07	69.00	*71.00*	**71.00**	67.00	72.63	*74.35*	**74.74**	68.02
baseball	68.81	*70.64*	**70.64**	74.74	75.00	*77.00*	**77.00**	71.00	71.77	*73.68*	**73.68**	72.82
hockey	77.66	*79.12*	**79.12**	62.59	73.00	*72.00*	**72.00**	87.00	75.26	*75.39*	**75.39**	72.80
motorcycle	73.39	*75.00*	**75.93**	76.00	80.00	*81.00*	**82.00**	76.00	76.56	*77.88*	**78.85**	76.00
crypt	79.44	*78.18*	**82.69**	83.63	85.00	*86.00*	**86.00**	85.00	82.13	*81.90*	**84.31**	86.29
electronics	69.07	*68.75*	**66.67**	75.73	67.00	*66.00*	**68.00**	78.00	68.02	*67.35*	**67.33**	76.85
medicine	77.55	*77.00*	**78.57**	78.16	76.00	*77.00*	**77.00**	68.00	76.77	*77.00*	**77.78**	72.73
space	79.17	*80.00*	**78.57**	83.50	76.00	*76.00*	**77.00**	70.00	77.55	*77.95*	**77.78**	77.78
Micro-Average	75.13	*75.75*	**76.25**	75.25	75.13	*75.75*	**76.25**	75.25	75.13	*75.75*	**76.25**	75.25
Macro-Average	75.22	*75.84*	**76.38**	75.43	75.13	*75.75*	**76.25**	75.25	75.17	*75.79*	**76.32**	75.83

assumptions: $C_2S_1 = 0, C_1S_2 \neq 0$ or $t < \frac{C_1S_2}{C_2S_1}$, $dist'(t)$ is positive, and $dist(t)$ monotonically increases. This indicates that the scaling increases the similarity of the query with the document.

The rationale of the second assumption is explained as follows. When the query and the document belong to the same category, they are more likely to both contain *label-relevant* terms. The co-occurrence of these terms will cause the pairwise product C_1 to be large in comparison to C_2, since the latter only contains terms not related to labels. The extent of "large" is measured by the ratio $\frac{S_1}{S_2}$. Conversely, when the query and the document do not belong to the same category, it is likely that C_1 is not large in comparison to C_2, which may cause $dist'(t)$ to be negative. In such a case, the similarity between the query and the document will be decreased. Therefore, scaling can both increase the similarity of the query and the document when they are in the same category and may also decrease the similarity when they are in different categories. Both effects are favorable for improving the classification accuracy.

In hierarchical text categorization, the categories are organized into a hierarchy, and categories in lower levels are more specific than those in upper levels. Therefore, it is reasonable to design an adaptive scaling strategy to reflect the differences. Specifically, the scaling factors for vectors of *label-relevant* terms should be dependent on the position of the category node in the hierarchy. Assume that the categories in leaf nodes receive a scaling factor q, then for a node i, its scaling factor is defined by $q_i = \frac{q}{c(i)}$, where $c(i)$ represents the number of leaf nodes among the descendants of the node i, and $c(i) = 1$ if i represents a leaf node. Each test document is considered as a query, and its similarities with documents of different categories are computed using the cosine metric. The category of the test document is obtained by the k-Nearest-Neighbor method.

3. EXPERIMENTS

The data sets in our experiments include Reuters-21578 and the 20 Newsgroups collection. The categories in both data sets are organized as taxonomies and the label for each category is predefined. All documents in the data sets were preprocessed. After stop word removal and stemming, we filtered out terms with less than two characters. No feature selection was performed in our experiments. To decide label-relevant terms for each category, we carried out LSI with the amount of dimension reduction set at 50. In the classification process the number of nearest neighbors was set at 20, with other values generating similar results.

We compared the performance of two variants of our approach (uniform scaling marked by NADP and adaptive scaling marked by SLSI) with two other approaches: (1) kNN classifiers whose similarities are obtained in the LSI space (marked by K-LSI); (2) hierarchical SVM classifiers using the linear kernel and default parameter values (marked by SVM). We summarized the overall results in Tables 1, 2 and 3 for the three category trees. The difference between NADP and SLSI is that the former applies uniform scaling to all the nodes in the hierarchy while the latter applies adaptive scaling. In most cases, both label-driven scaling and adaptive scaling significantly improve the classification performance, and in some cases SLSI even outperforms SVM.

4. CONCLUSIONS

In this paper, we proposed a novel approach to enhance Latent Semantic Indexing (LSI) by making two contributions: label-driven scaling and hierarchy-dependent adaptive scaling. Experimental results on real-world data show that our approach is able to substantially improve the performance of hierarchical text categorization.

5. ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (No. 60573077), MSRA Internet Services Theme, MOE-MS Key Laboratory of Multimedia Computing and Communication of USTC (No. 07122807, 06120805)

6. REFERENCES

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

[2] Chris H. Q. Ding, *A similarity-based probability model for latent semantic indexing*, SIGIR 1999: 58–65

[3] S. Chakraborti, R. Mukras, R. Lothian, N. Wiratunga, S. Watt and D. Harper, *Supervised Latent Semantic Indexing Using Adaptive Sprinkling*, IJCAI 2007: 1582-1587

[4] M. Efron, *Model-averaged latent semantic indexing*, SIGIR 2007: 755–756