Combining LSI with other Classifiers to Improve Accuracy of Single-label Text Categorization

Ana Cardoso-Cachopo
IST — TULisbon / INESC-ID; Av. Rovisco Pais, 1;
1049-001 Lisboa — Portugal; acardoso@ist.utl.pt

Arlindo L. Oliveira
IST — TULisbon / INESC-ID; Rua Alves Redol, 9;
1000-029 Lisboa — Portugal; aml@inesc-id.pt

Abstract—**This paper describes the combination of k-NN and SVM with LSI to improve their performance in single-label text categorization tasks, and the experiments performed with six datasets to show that both k-NN-LSI (the combination of k-NN with LSI) and SVM-LSI (the combination of SVM with LSI) outperform the original methods for a significant fraction of the datasets. Overall, both combinations present an average Accuracy over the six datasets used in this work that is higher than the average Accuracy of each original method. Having in mind that SVM is usually considered the best performing classification method, it is particularly interesting that the combinations perform even better for some datasets.**

I. Introduction and Experimental Setting

The main goal of *text categorization* (TC) is to derive methods for the categorization of natural language text. The objective is to derive methods that, given a set of training documents with known categories, and a new document, which is usually called the *query*, will predict the query´s category. In this paper, we are interested in the case where the query belongs to a single category, also called *single-label text categorization*. In our work, we present the results obtained using some well known classification methods, namely the Vector method [8], k-NN [7], [9], LSI [4], and SVM [6], and compare them with the results obtained using a combination of k-NN and SVM with LSI.

To allow the comparison of our work with previously published results, we used three standard TC collections in our evaluation, namely the 20-Newsgroups, Reuters-21578 and Webkb. We also used two other collections, Bank which is a collection of messages sent to a bank's help-desk along with their respective answers, and Cade which is a collection of webpages from a Brazilian web directory. To be consistent with 20-Newsgroups and Reuters-21578, for Webkb, Bank, and Cade, we randomly split the documents into two thirds for training and the remaining for testing. Table I contains relevant information regarding the sizes and document distributions for the six datasets.

Regarding algorithm implementation and the parameters that were used, for the Vector method we used a Sourceforge project called IGLU [5]. For k-NN we implemented a "voting strategy", where the possible classes of a document are voted on by the documents that belong to that class; we used cosine similarity and considered only the 10 nearest documents. For LSI we used FAQO [2], and considered a reduced matrix with 200 dimensions. For SVM we used LIBSVM [3]and used a linear kernel in our experiments. We implemented the combinations between methods as described bellow.

II. Combinations Between Methods

This section describes the rationale behind the combination of k-NN and SVM with LSI, that ideally will perform better than the original methods.

The difference to the original approaches is that now, instead of applying their transformations to the usual term/document matrix used in the vector method, the combinations apply their transformations to the concept space that was previously obtained using Singular Value Decomposition. Then, given another p-dimensional vector representing the query document, choose one of the options:

- Apply to the query document the same transformation as the one applied to the initial term/document matrix; apply cosine similarity to the transformed query and to each of the transformed train documents, select the k most similar documents, apply a voting strategy, where each transformed document "votes" for its class, weighted by its similarity to the transformed query; the class of the query is the most voted class — k-NN-LSI method. This method was already proposed in [1].
- Apply a kernel function to the transformed concept matrix, so that concepts are represented in a high dimensional feature space, where each class is linearly separable from the others; apply to the query document the same transformation applied to the initial term/document matrix; apply a voting strategy, where possible classes are ranked according to the number of votes that they had in a one-against-one classification approach; the class of the query is the class which got more votes — SVM-LSI method.

III. Comparing the combinations with the original methods

Table II shows Accuracy values obtained by each method for each of the six datasets, and average Accuracy over all the datasets for each method. The Dumb classifier ignores the query and always predicts the most frequent class in the training set, and is included here to provide a baseline Accuracy value.

When comparing k-NN-LSI with k-NN and LSI, it is important to note that, from the two original methods, there

Dataset	Collection	Classes	Train Docs	Test Docs	Total Docs	Smallest Class	Largest Class
Bank37	Bank	37	928	463	1391	5	346
20Ng	20-Newsgroups	20	11293	7528	18821	628	999
R8	Reuters-21578	8	5485	2189	7674	51	3923
R52	Reuters-21578	52	6532	2568	9100	3	3923
Web4	Webkb	4	2803	1396	4199	504	1641
Cade12	Cade	12	27322	13661	40983	625	8473

TABLE I

DESCRIPTION OF THE DATASETS: COLLECTION FROM WHICH IT IS DERIVED, NUMBER OF CLASSES, NUMBER OF TRAIN DOCUMENTS, NUMBER OF TEST DOCUMENTS, TOTAL NUMBER OF DOCUMENTS, NUMBER OF DOCUMENTS IN THE SMALLEST CLASS, NUMBER OF DOCUMENTS IN THE LARGEST CLASS.

Dataset	Dumb	Vector	k-NN	SVM	LSI	k-NN-LSI	SVM-LSI
Bank37	0.2505	0.8359	0.8423	0.9071	0.8531	0.8488	0.9179
20Ng	0.0530	0.7240	0.7593	0.8284	0.7491	0.7557	0.7775
R8	0.4947	0.7889	0.8524	0.9698	0.9411	0.9488	0.9680
R52	0.4217	0.7687	0.8322	0.9377	0.9093	0.9100	0.9311
Web4	0.3897	0.6447	0.7256	0.8582	0.7357	0.7908	0.8897
Cade12	0.2083	0.4142	0.5120	0.5284	0.4329	0.4880	0.5465
Average	0.3030	0.6961	0.7540	0.8383	0.7702	0.7904	0.8385

TABLE II

ACCURACY OBTAINED BY EACH METHOD FOR EACH OF THE SIX DATASETS, AND AVERAGE ACCURACY OVER ALL THE DATASETS FOR EACH METHOD.

is none that always outperforms the other. For Bank37, R8, R52, and Web4, LSI performs better than k-NN, whereas for 20Ng and Cade12 it is k-NN that provides better results. This is probably because LSI is very effective at finding the "concepts" in the first datasets, but for the other datasets, which consist of newsgroup messages (that can quote others) and web pages (that can be copies of others), finding the most similar document is more effective. For R8, R52, and Web4, the combination of k-NN with LSI is the best performing method. For the other datasets, k-NN-LSI is second best, independently of which of the original methods shows a better performance, and its Accuracy is closer to the one achieved by the best method. As can be seen in Table II, if one considers average Accuracy over the six datasets, k-NN-LSI is the best performing method, when compared to k-NN and LSI.

When comparing SVM-LSI with SVM and LSI, we can see that, for all datasets, the worst performing method is LSI. SVM-LSI is the best for the datasets in Portuguese, Bank37 and Cade12 and also for Web4. SVM is the best for the datasets in English, R8, R52 and 20Ng. Having in mind that SVM was the best performing method in several comparisons of classification methods [9], [1], it is particularly interesting that its combination with LSI performs even better for some datasets. As can be seen in Table II, when one considers average Accuracy over the six datasets, SVM-LSI slightly outperforms SVM, even if this difference is not significant.

Given the results obtained, it is important to note that the best performing method depends on the dataset that is used. As such, it is important to test different methods and combinations to decide which one to use in each situation.

IV. CONCLUSIONS AND FUTURE WORK

We described the combination of k-NN and SVM with LSI and showed that SVM-LSI is the method that presents the best performance for some of the datasets that were used in our experiments.

We think that, given the present results, this kind of method combination represents an interesting line of research, and that more tests need to be done, namely regarding the combinations between the number of dimensions that are considered in the LSI method and the kernel function that is used by the SVM method.

REFERENCES

[1] A. Cardoso-Cachopo and A. Oliveira. An empirical comparison of text categorization methods. In *Proceedings of SPIRE-03*, pages 183–196. Springer Verlag, 2003.

[2] J. Caron. Experiments with LSA scoring: Optimal rank and basis. Presented at SIAM Computational Information Retrieval Workshop, 2000.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

[4] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[5] IGLU Java—Java Library for Information Retrieval Research, 2002.

[6] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142. Springer Verlag, 1998.

[7] B. Masand, G. Linoff, and D. Waltz. Classifying news stories using memory-based reasoning. In *Proceedings of SIGIR-92*, pages 59–65. ACM Press, 1992.

[8] G. Salton. *The SMART Retrieval System*. Prentice Hall, 1971.

[9] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-99*, pages 42–49. ACM Press, 1999.