Using Clustering to Enhance Text Classification

Antonia Kyriakopoulou
Department of Informatics
Athens University of Economics and Business
76 Patission St., Athens, GR 104.34
tonia@aueb.gr

Theodore Kalamboukis
Department of Informatics
Athens University of Economics and Business
76 Patission St., Athens, GR 104.34
tzk@aueb.gr

ABSTRACT

This paper addresses the problem of learning to classify texts by exploiting information derived from clustering both training and testing sets. The incorporation of knowledge resulting from clustering into the feature space representation of the texts is expected to boost the performance of a classifier. Experiments conducted on several widely used datasets demonstrate the effectiveness of the proposed algorithm especially for small training sets.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning;
H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Algorithms

Keywords

Text classification, clustering

1. INTRODUCTION

Text classification is one of the first applications of machine learning and applies to the general problem of supervised inductive learning: given a set of training documents, classified manually to one or more predefined categories, the classifier learns to automatically classify new unseen documents, about which there is no prior knowledge. To boost its performance, we seek for new information about the distribution of the documents to be classified. This information comes from clustering both training and testing sets and is embodied in the data in the form of *meta*-information.

Clustering has been used in the literature of text classification either as an approach for dimensionality reduction or as a technique to enhance the training set. In the second case, the enhancement is achieved either by extending the feature vectors of the training examples with new features derived from clustering or by expanding the training set with new examples derived from a large set of unlabeled data (see [1, 5, 7] for an example of each case).

In this article, a new algorithm combining clustering and classification is proposed. Our motivation relies on the expectation that any prior knowledge about the nature of the testing examples, i.e. the ones that are to be classified, will support the construction of a more efficient classifier for the same examples. In what follows, we briefly present the algorithm and demonstrate its performance with experimental results on some commonly used data collections.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

2. THE ALGORITHM

Consider a k-class categorization problem, ($k \geq 2$), with a labeled l-training sample $\{(\vec{x_1}, y_1), \ldots, (\vec{x_l}, y_l)\}$ of feature vectors $\vec{x_i} \in \mathcal{R}^n$ and corresponding labels $y_i \in \{1, \ldots, k\}$, and an unlabeled m-testing sample $\left\{\vec{x_1^*}, \ldots, \vec{x_m^*}\right\}$ of feature vectors. We are interested in the case where $m \gg l$. The features are valued using the *TF*IDF* weighting scheme [6].

The proposed algorithm consists of three steps: clustering, expansion and classification step. For the classification task we assume that classes correspond to thematic topics. This assumption corresponds to an ideal case of clustering where all the examples of a class will be clustered together since they share the same word distribution. So we have considered that there is a one-to-one correspondence between classes, topics and clusters. Hense, in the clustering step of the algorithm, the number of clusters is chosen to be equal to k, i.e. the predefined number of classes. At present, the experimental results verify our conjecture, although other clustering algorithms should also be tested in this step. The CLUTO^{TM} Clustering Toolkit [3] is used and a divisive clustering algorithm with repeated bisections is selected for clustering both the training and testing sets.

In the expansion step, each cluster contributes one *meta*-feature to the feature space of the training and testing sets: given the total n features used in the representation of the $l + m$ feature vectors and the k clusters from the clustering step, we create *meta*-features $x_{n+1}, \ldots, x_{n+k}$. The weight of these *meta*-features is computed applying the *TF*IDF* weighting scheme to the clusters. We consider that all the documents in a cluster C_j share the same *meta*-feature whose *frequency* within a document $\vec{x}$ of the cluster equals to one, $TF(x_{n+j}, \vec{x}) = 1$, its *document frequency* equals to the size of the cluster, $DF(x_{n+j}) = |C_j|$, and its *inverse document frequency* is $IDF(x_{n+j}) = \log_2\left(\frac{l+m}{|C_j|}\right)$.

Finally, in the classification step the SVM^{light} implementation of SVMs and transductive SVMs is used [2].

3. EXPERIMENT SETTINGS

The empirical evaluation is done on four *single*-labeled datasets: a "by-date" version of 20Newsgroup, Reuters-8

Table 1: Average p/r breakeven points for different sizes of the training sets.

% train	20Newsgroup		WebKB		Reuters-8		Reuters-52	
	SVM	CL-SVM	SVM	CL-SVM	SVM	CL-SVM	SVM	CL-SVM
0.5	40.39	56.51	48.32	58.38	54.70	56.50	42.84	43.33
1	50.26	63.47	49.28	56.97	69.83	71.04	46.36	47.56
5	67.98	72.19	51.37	58.66	80.67	81.33	54.43	55.57
10	72.29	75.14	56.59	60.45	85.42	86.72	58.06	59.05
20	76.34	77.79	64.14	68.95	88.42	89.56	68.38	69.78
50	80.29	81.04	69.08	72.65	90.17	90.84	72.27	72.98
100	82.35	82.79	72.85	74.28	91.81	93.23	80.87	81.51

and Reuters-52 subsets[1] of Reuters-21578[2], and WebKB[3]. For WebKB the settings in [2] are followed. For the rest of the corpora no feature selection is done, both stemming and stopword-removal are used.

A binary classifier is constructed for each class of the *expanded* dataset, a linear kernel is selected and the weight C of the slack variables is set to default.

A four-fold cross validation is applied to each experiment. The algorithm runs four rounds with different samples from the training set. These samples are selected randomly and uniformly by dividing the training set in smaller and equal parts and by preserving the same proportion of documents per category with that of the original dataset. All testing documents are used. The precision/recall breakeven point (BEP) is used as a measure of performance.

4. RESULTS AND CONCLUSIONS

Several experiments are conducted with one or more *meta*-features added on the expansion step. To provide a baseline for comparison, results from the standard SVM classifier without the clustering and expansion steps are also presented.

Table 1 shows the results of the experiments. The SVM combined with clustering (CL-SVM) constantly outperforms the standard SVM on all datasets raising the average of the breakeven points up to 15% in some cases. For the sake of brevity, the results from the standard TSVM are omitted. However, it should be noted that the TSVM combined with clustering has a similar behaviour also leading to an improved performance compared to the standard TSVM.

In figure 1, we demonstrate the impact of the size of the training set for the 20Newsgroup dataset. This graph shows that the advantage of using clustering as a former step to classification is largest in the case of small training sets. When the size of the training set increases, the performance of the CL-SVM approaches that of the standard SVM. The rest of the collections have a similar behaviour.

This algorithm stood out in a spam-filtering task [4] in the challenge competition of the ECML 2006 Conference.

To conclude, the use of clustering to boost classification seems to be a promising approach with several extensions that should be further investigated. Such an extension, currently under examination, is the case where each cluster contributes more than one *meta*-features. Preliminary results demonstrate further performance improvement especially when only a few training examples are available.

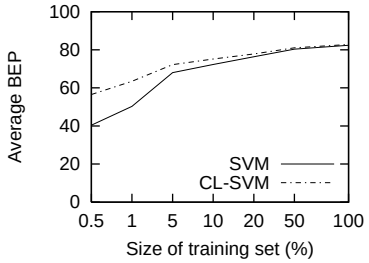

Figure 1: Average BEPs for the 20Newsgroup dataset for different sizes of the training set.

5. ACKNOWLEDGMENTS

The authors would like to thank Andreas Vlachos for his constructive ideas in the early stages of our research.

6. REFERENCES

[1] R. Bekkerman, R. El-Yaniv, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.

[2] T. Joachims. Transductive inference for text classication using support vector machines. In *16th International Conference on Machine Learning Proceedings*, pages 200–209, 1999.

[3] G. Karypis. Cluto a clustering toolkit. *Technical Report 02-017*, 2002.

[4] A. Kyriakopoulou and T. Kalamboukis. Text classification using clustering. In *ECML-PKDD Discovery Challenge Workshop Proceedings*, 2006.

[5] B. Raskutti, H. Ferr, and A. Kowalczyk. Using unlabeled data for text classification through addition of cluster parameters. In *9th International Conference on Machine Learning ICML Proceedings*, 2002.

[6] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[7] H. Zeng, X. Wang, Z. Chen, H. Lu, and W. Ma. Cbc: Clustering based text classification requiring minimal labeled data. In *3rd IEEE International Conference on Data Mining Proceedings*, 2003.

[1]Available at `http://www.gia.ist.utl.pt/~acardoso/datasets/`.

[2]Available at `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.

[3]Available at `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/`.