Combining Clustering with Classification for Spam Detection in Social Bookmarking Systems *

Antonia Kyriakopoulou and Theodore Kalamboukis

Department of Informatics,
Athens University of Economics and Business,
76 Patission St., Athens, GR 104.34
{tonia.tzk}@aueb.gr
http://pages.cs.aueb.gr/ipl/ir/

Abstract. *This paper addresses the problem of learning to classify texts by exploiting information derived from both training and testing sets. To accomplish this, clustering is used as a complementary step to text classification, and is applied not only to the training set but also to the testing set. This approach allows us to study the location of the testing examples and the structure of the whole dataset, which is not possible for an inductive learner. The incorporation of this knowledge to the feature representation of the texts is expected to boost the performance of a SVM/TSVM classifier. Experiments conducted on tasks and datasets provided in the framework of the ECML/PKDD 2008 Challenge Discovery on spam detection on social bookmarking systems, demonstrate the effectiveness of our approach. The experiments show substantial improvements on classification performance.*

Key words: classification, clustering, spam detection

1 Introduction

Text classification and clustering have been the focus of critical research in the areas of machine learning and artificial intelligence. In the literature, these two streams flow independently of each other, despite their akin close conceptual and practical relations. However, there are several important research issues encapsulated into text classification tasks and the role of clustering in support of these tasks is also of great significance.

A standard research issue for text classification is the creation of compact representations of the feature space and the discovery of the complex relationships that exist between features, documents and classes. In this vein, an important area of research where clustering is used to aid text classification is the area of dimensionality reduction. Clustering is used as a *feature compression and/or*

* This paper is an improved version of the paper presented in the ECML/PKDD 2008 Discovery Challenge.

extraction method: features are clustered into groups based on selected clustering criteria. Feature clustering methods create new, reduced-size event spaces by joining similar features into groups. They define a similarity measure between features, and collapse similar features into single events that no longer distinguish among their constituent features. Typically, the parameters of the cluster become the weighted average of the parameters of its constituent features. Two types of clustering have been identified: i) one-way clustering, i.e. feature clustering based on the distributions of features in the documents or classes [2],[16] and ii) co-clustering, i.e. clustering both features and documents [4].

A second research area of text classification where clustering has a lot to offer, is the area of *semi-supervised learning*. Training data contain both labelled and unlabelled examples. Obtaining a fully labelled training set is a difficult task; labelling is usually done using human expertise, which is expensive, time consuming, and error prone. Obtaining unlabelled data is much easier since it involves collecting data that are known to belong to one of the classes without having to label them. Clustering is used as a method to extract information from the unlabelled data in order to boost the classification task. In particularly, clustering is used: i) to create a training set from the unlabelled data [5], ii) to augment the training set with new documents from the unlabelled data [18], [19], iii) to augment the dataset with new features [13], [9], [10], and iv) to co-train a classifier [14], [11].

Finally, *clustering in large-scale classification problems* is another major research area in text classification. A considerable amount of work is done on using clustering to reduce the training time of a classifier when dealing with large data sets. In particular, while SVM classifiers (see [3] for a tutorial) have proved to be a great success in many areas, their training time is at least $O(N^2)$ for training data of size N, which makes them non favourable for large data sets. The same problem applies to other classifiers as well. In this vein, clustering is used as a down-sampling pre-process to classification, in order to reduce the size of the training set resulting in a reduced dimensionality and a smaller, less complex classification problem, easier and quicker to solve [17], [1]. However, it should be noted that dimensionality reduction is not accomplished directly using clustering as a feature reduction technique as discussed earlier, but rather in an indirect way through the removal of training examples that are most probably not useful to the classification task and the selection of the most representative redundant training set. In most of the cases this involves the collaboration of both clustering and classification techniques.

For a detailed review and interpretation of the role of clustering in different fields of text classification see [12].

In this paper, we deal with the text classification aided by clustering scenario and apply it to the problem of spam detection in social resource sharing systems. Social resource sharing systems are web–based systems that allow users to upload their resources, and to label them with arbitrary words, so–called *tags*. The systems can be distinguished according to what kind of resources are sup-

ported. The system under investigation is called BibSonomy[1] and it is a social bookmark and publication sharing system that allows sharing bookmarks and BibTex entries simultaneously. A formal description of the underlying structure which is called *folksonomy* is given in [7].

The paper is organized as follows: next section presents the algorithm. Section 3 presents the empirical evaluation. We conclude by pointing out open issues and limitations of the algorithm presented.

2 The Algorithm–Using Clustering For Text Classification

Consider a k–class categorization problem, ($k = 1$ in the case of spam detection on social bookmarking systems), with a labeled training sample $Tr = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_l}, y_l)\}$ of feature vectors $\boldsymbol{x} \in \mathcal{R}^n$ and corresponding labels $y_i \in \{1, \ldots, k\}$, and an unlabeled testing sample $Te = \{\boldsymbol{x_1^*}, \ldots, \boldsymbol{x_m^*}\}$ of feature vectors. The features are valued using the *TF*IDF* weighting scheme [15], defined by

$$W(f_i) = TF(f_i, \boldsymbol{x}) * IDF(f_i) \tag{1}$$

where the *term frequency* of feature f_i, $TF(f_i, \boldsymbol{x})$, is the number of its occurences in document $\boldsymbol{x}$, the *inverse document frequency* is

$$IDF(f_i) = \log_2 \left(\frac{|D|}{DF(f_i)} \right) \tag{2}$$

where $|D| = |Tr \cup Te|$ is the number of documents in the dataset, and the *document frequency*, $DF(f_i)$, is the number of documents that contain f_i at least ones. All feature vectors are normalized to unit length.

The algorithm consists of the following three steps:

- *Clustering step:* to cluster both the training and testing set.
- *Expansion step:* to augment the dataset with *meta*–features originated from the clustering step.
- *Classification step:* to train a classifier with the expanded dataset.

2.1 Clustering Step

For the clustering step of the algorithm, we need to define the desired number of clusters into which the dataset should be clustered. Results from experiments [10] conducted on three widely used corpora (Reuters, 20Newsgroup, and WebKB) have shown an increase of performance of classification when the number of clusters is equal to the number of the predefined classes. In traditional classification tasks it can be assumed that the classes correspond to topics, and there is a one-to-one correspondence between the topic and the class under which the

[1] http://www.bibsonomy.org

data are classified. Moreover, the examples of a class are clustered together which is logical since they share the same word distribution. So we can assume that there is a one-to-one correspondence between classes, topics and clusters, and use this information to define the desired number of clusters. In spam detection on social bookmarking systems we can't make such safe assumptions. Spam user posts can deal with many different topics, there is a one-to-many correspondence between the class *spam* and the topics of the posts that fall under it. So, the number of topics can't be determined beforehand. Hense, the number of clusters to select is two: one cluster with the spam posts and one cluster with the non-spam.

The $CLUTO^{TM}$ Clustering Toolkit [8] is used and a divisive clustering algorithm with repeated bisections is selected for clustering *both* the training and testing sets. In this method, the disired k–way clustering solution is computed by performing a sequence of $k-1$ repeated bisections. The dataset is first clustered into two groups, then one of these groups is selected and disected further. This process continuous until the desired k number of clusters is found. During each step, the cluster is bisected so that the resulting k–way clustering solution optimizes the internal criterion function

$$\max \sum_{g=1}^{k} \sqrt{\sum_{u,v \in S_g} sim(u,v)} \tag{3}$$

where S_g is the set of documents assigned to the g^{th} cluster, u and v represent two documents, and $sim(u,v)$ is the similarity between two documents. The generated set of clusters $G = \{G_1, G_2, \ldots, G_k\}$ consists of k non–overlapping clusters.

2.2 Expansion Step

In the expansion step, each cluster in G contributes one *meta*–feature to the feature space of the training and testing sets, i.e. k *meta*–features are created. 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 G_j share the same *meta*–feature mf_j whose *frequency* within a document $\boldsymbol{x}$ of the cluster equals to one, $TF(mf_j, \boldsymbol{x}) = 1$, its *document frequency* equals to the size of the cluster, $DF(mf_j) = |G_j|$, and its *inverse document frequency* is $IDF(mf_j) = \log_2\left(\frac{|D|}{|G_j|}\right)$. Then by properly adjusting Equation 1 the weight of mf_j is defined by

$$W(mf_j) = \log_2\left(\frac{|D|}{|G_j|}\right) \tag{4}$$

2.3 Classification Step

Finally, in the classification step the SVM^{light} implementation of SVMs and TSVMs is used. A binary classifier is constructed for the *expanded* dataset, a linear kernel is used and the weight C of the slack variables is set to default.

3 A Performance Study

3.1 Experiment Settings

The empirical evaluation is done in two tasks created and published in the framework of ECML/PKDD 2008 Discovery Challenge[2].

- Task A deals with spam detection in social bookmarking systems. The goal of this task is to learn a model which predicts whether a user is a spammer or not. In order to detect spammers as early as possible, the model should make good predictions for a user when he submits his first post.
- In Task B the aim is to support the user during the tagging process and to facilitate the tagging. BibSonomy includes a tag recommender. This means that when a user finds an interesting bibtex or bookmark and posts it to BibSonomy, the system offers up to ten recommended tags on the posting page. The goal is to learn a model which effectively predicts the tags a user will use to describe his post.

The dataset provided consists of data, in the form of posts, collected from 2.638 active non-spam users and 36.282 spam users by manually labeling spammers and non–spammers. It was devided in a training set which was provided at the beginning of the competition, and a testing set which was released 48 hours before the deadline. Users' posts are either bibtex or bookmarks. They include all public information such as the url, the description, the title and the user defined tags. The evaluation criterion prescribed by the competition is the AUC value.

3.2 Results

Several experiments were conducted during the contest on the given training set as well as after the publication of the true classification labels of the testing set. In this paper only the results from experiments on the whole dataset are presented. In all the experiments, the given dataset was pre-processed as follows. First, for each user, all his posts, BibTex and bookmarks, were considered as one record (feature vector), i.e. there were no multiple records per user as in the original dataset. Then, different versions of this dataset were created. One containing all public information including tags and urls, a second containing all public information except from tags and urls, a third without the tags, and a fourth

[2] Additional information can be found in http://www.kde.cs.uni-kassel.de/ws/rsdc08/

without the urls. The reason for the different versions created was to examine the impact of using tags and urls in the spam detection process. As stated in [6] tags are considered as one of the ways that a spam user can use to corrupt a bookmarking system. The more often a web page is submitted and tagged, the better chance it has of being found. Spam users bookmark the same web page multiple times and tag each page of their web site using a lot of popular tags. According to this fact, we can hypothesize that a url is a serious indicator of a spam user, whereas a tag is a deceptive one.

Also, we wanted to examine the effect of applying stemming and stopword removal mechanisms to the dataset. A series of experiments was conducted in this basis too. It should be noted that numbers, words with length less than two and punctuation marks where discarded for all datasets. Finally, the *TF*IDF* weighting scheme is applied and all users' vectors are normalized to unit length.

The following experiment scenarios were conducted:

- *case1*: the dataset is used without the tags and urls, whereas stemming and stopword removal are applied .
- *case2*: the dataset is used without the tags and urls, and stemming and stopword removal are not applied.
- *case3*: the dataset is used with the tags and urls, and stemming and stopword removal are applied.
- *case4*: the dataset is used with the tags and urls, whereas stemming and stopword removal are not applied.
- *case5*: the dataset is used without the tags, with the urls, whereas stemming and stopword removal are applied.
- *case6*: the dataset is used without the tags, with the urls, and stemming and stopword removal are not applied.
- *case7*: the dataset is used with the tags, without the urls, whereas stemming and stopword removal are applied.
- *case8*: the dataset is used with the tags, without the urls, and stemming and stopword removal are not applied.

We applied our algorithm only on Task A. The results of these experiments are presented in Table 1. To provide a baseline for comparison, results from the standard SVM and transductive SVM (TSVM) classifiers are also presented. C-SVM and C-TSVM correspond to the perfomance of an SVM/TSVM classifier when it is used in conjuction with clustering according to our algorithm.

First of all, we can see that the C-SVM and C-TSVM classifiers perform better than the standard SVM and TSVM classifiers. In all cases, the C-SVM classifier has the best performance over the rest of the classifiers. The best result of 98.96% is obtained when the tags and urls are both included in the dataset and stemming and stopword removal are not applied (case 4). In almost all cases, the application of stemming and stopword removal deteriorates performance. The inclusion or not of tags and urls in the dataset also affects performance. When tags and urls are both included in the dataset as in cases 3 and 4, the classification performance is higher in contrast with the cases 1 and 2 where tags and urls are excluded. Also, when tags and urls are used in turn as in cases 5 to

Table 1. Results for Task A.

	SVM	C-SVM	TSVM	C-TSVM
Case1	96.01	97.56	96.56	97.60
Case2	96.66	97.31	96.59	96.60
Case3	97.71	98.84	97.90	98.10
Case4	97.03	98.96	97.62	98.61
Case5	97.22	98.68	97.36	97.87
Case6	97.05	97.83	96.96	96.99
Case7	97.12	98.58	97.16	97.80
Case8	97.01	97.21	96.66	96.94

8, we can see that the performance is better when urls are used whereas tags are not used in the dataset. The results confirm the original hypothesis that stated that a url is a serious indicator of a spam user, whereas a tag is a deceptive one.

4 Conclusions

We presented empirical results on datasets given in the framework of the ECML/ PKDD Discovery Challenge 2008 on spam detection in social bookmarking systems. On all experiments conducted, the clustering approach combined with a SVM/TSVM classifier showed improvements over the use of a standard SVM/ TSVM classifier on its own. The applycation of stemming and stopword removal mechanisms, revealed a deterioration in performance and their use is not encouraged in this context. Also, the results confirm the hypothesis that urls can be considered as a serious indicator of spam users, whereas tags can be deceptive.

One limitation of our algorithm is that when a new user makes his first post, the same procedure of clustering, *meta*–feature addition, and classification, should be applied again for the whole dataset, a rather time consuming, and computationally expensive process. A suggestion would be to use incremental clustering instead of the static clustering algorithm used now. Incremental clustering is a method that deals with the problem of updating clusters without frequently performing complete reclustering. This would be a more suitable way for maintaining clusters in the typical, dynamic environment of spam detection.

Another issue about our algorithm is its rather naive approach to clustering that may not capture all the *meta*–information possible hidden in the dataset. More sophisticated clustering methods have been proposed in the literature that focus on incorporating prior knowledge into the clustering process; conceptual clustering, topic–driven clustering, just to name a few. These methods are based in the idea that it is possible to use explicitly available domain knowledge to constrain or guide the clustering process. In our case, the class labels of the training set can constitute the domain knowledge and be used as guidance to a clustering algorithm. This way, it can be reassured that the created clusters will reflect the major concepts included in the corpus.

Other issues that can be further researched include the estimation and statistical basis of the optimum number of clusters and *meta*–features to be used.

References

1. Awad, M., Khan, L., Bastani, F., Yen, I. L.: An effective support vector machines (SVMs) performance using hierarchical clustering. In: 16^{th} IEEE International Conference on Tools with Artificial Intelligence, pp. 663–667, (2004)
2. Baker, L. D., McCallum, A.: Distributional Clustering of Words for Text Classification. In: ACM SIGIR 1998, pp. 96–103 (1998)
3. Burges, J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. In: Knowledge Discovery and Data Mining, vol. 2, pp. 121–167, (1998)
4. Dhillon, I. S., Mallela, S., Kumar, R.: A Divisive Information–Theoretic Feature Clustering Algorithm for Text Classification. Journal of Machine Learning Research, 3, pp. 1265–1287 (2003)
5. Fung, G., Mangasarian, O. L.: Semi–Supervised Support Vector Machines for Unlabeled Data Classification. Technical report (2001)
6. Hammond, T., Hannay, T., Lund, B., Scott, J.: Social Bookmarking Tools (I): A General Review. D-Lib Magazine 11, Nr. 4, (2005)
7. Hotho, A., Jaschke, R., Schmitz, C., Stumme, G.: BibSonomy: A Social Bookmark and Publication Sharing System. In: 1^{th} Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures. Aalborg: Aalborg Universitetsforlag, pp. 87–102 (2006)
8. Karypis, G.: CLUTO a clustering toolkit.Technical report (2002)
9. Kyriakopoulou, A., Kalamboukis, T.: Text classification using clustering. In: ECML-PKDD Discovery Challenge Workshop (2006)
10. Kyriakopoulou, A., Kalamboukis, T.: Using clustering to enhance text classification. In SIGIR 2007, 30^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp805–806 (2007)
11. Kyriakopoulou, A.: Using Clustering and Co–Training to Boost Classification Performance. In: ICTAI (2). IEEE Computer Society, pp. 325–330 (2007)
12. Kyriakopoulou, A.: Text Classification Aided by Clustering: a Literature Review. I-Tech Education and Publishing KG, Vienna, Austria (2008)
13. Raskutti, B., Ferra, H. L., Kowalczyk, A.: Combining clustering and co–training to enhance text classification using unlabelled data. In: KDD ACM 2000, pp. 620–625 (2000)
14. Raskutti, B., Ferra, H. L., Kowalczyk, A.: Using Unlabelled Data for Text Classification through Addition of Cluster Parameters. In: 19^{th} International Conference for Machine Learning (ICML 2002), pp. 114-121 (2002)
15. Salton, G., McGill, M. J.: Introduction to Modern Information Retrieval. McGraw Hill (1983)
16. Slonim, N., Tishby, N.: The power of word clustering for text classification. In: European Colloquium on IR Research, ECIR 2001, (2001)
17. Yu, H., Yang, J., Han, J.: Classifying large data sets using SVMs with hierarchical clusters. In: 9^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 306–315, Washington, DC, USA, ACM (2003)
18. Zeng, H. J., Wang, X. H., Chen, Z., Lu, H., Ma, W. Y.: CBC: Clustering Based Text Classification Requiring Minimal Labeled Data. In ICDM 2003, pp. 443–450, IEEE Computer Society (2003)
19. Zhou, D., Bousquet, O., Lal, T. N., Weston, J., Scholkopf, B.: Learning with Local and Global Consistency. In: NIPS 2003, MIT Press (2003)