Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas

Department of Informatics,
Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece
{katak,greg,vlahavas}@csd.auth.gr

Abstract. Real world text classification applications are of special interest for the machine learning and data mining community, mainly because they introduce and combine a number of special difficulties. They deal with high dimensional, streaming, unstructured, and, in many occasions, concept drifting data. Another important peculiarity of streaming text, not adequately discussed in the relative literature, is the fact that the feature space is initially unavailable. In this paper, we discuss this aspect of textual data streams. We underline the necessity for a dynamic feature space and the utility of incremental feature selection in streaming text classification tasks. In addition, we describe a computationally undemanding incremental learning framework that could serve as a baseline in the field. Finally, we introduce a new concept drifting dataset which could assist other researchers in the evaluation of new methodologies.

1 Introduction

The world wide web is a dynamic environment that offers many sources of continuous textual data, such as web pages, news-feeds, emails, chat rooms, forums, usenet groups, instant messages and blogs. There are many interesting applications involving classification of such textual streams. The most prevalent one is spam filtering. Other applications include filtering of pornographic web pages for safer child surfing and delivering personalized news feeds.

All these applications present great challenge for the data mining community mainly because they introduce and/or combine a number of special difficulties. First of all, the data is high dimensional. We usually consider as feature space a vocabulary of hundreds of thousands of words. Secondly, data in such applications always come in a stream, meaning that we cannot store documents and we are able to process them only upon their arrival. Thirdly, the phenomenon of concept drift [8] might appear. This means that the concept or the distribution of the target-class in the classification problem may change over time.

In this paper we tackle with another issue that, to the best of our knowledge, haven't been given enough attention by the research community. This is, the

initial unavailability of the feature space. There is no prior knowledge of the words that might appear over time and the use of a global vocabulary of millions of words is simply inefficient. To deal with this problem we introduce the idea of the *feature-based* classifier. This, is a special class of classifiers that can execute in a dynamic feature space.

We furthermore investigate the utility of Incremental Feature Selection (IFS) which deals with a specific type of concept drift appearing in applications, where feature selection is of vital importance. The main notion is that as time goes by, different set of features become important for classification and some totally new features with high predictive power may appear.

We also propose a computationally undemanding framework for the classification of text streams that takes under consideration the aforementioned statements: An incremental classifier that can execute in a dynamic feature space, enhanced by an IFS procedure. Due to its simplicity and effectiveness, this approach could serve as a baseline method for other, more advanced, stream learning techniques. Finally we introduce a new concept drifting dataset which we hope it will help other researchers evaluate their work.

This paper extends previous work-in-progress report [4] by introducing: a) a new concept drifting dataset from the news classification domain, b) an investigation on the effect of IFS in three classical stream learning techniques and c) a discussion on the need for a dynamic feature space. The rest of this paper is organized as follows: Section 2, presents background knowledge on text stream mining. In Section 3, we describe the proposed approach and in Section 4 we give details about the experimental setup and discuss results. Finally in Section 5 we conclude and present our future plans.

2 Text Streams and Concept Drift

Data stream mining is a field that draws attention from data mining and database community [1, 7, 3]. The main distinctiveness of the data is that we cannot store incoming records/transactions/documents and therefore we need algorithms that process the data only once. Text streams, have additional difficulties. Some of them, like the initial unavailability of the feature space and the occurrence of concept drift are already mentioned in the introduction.

A lot of effort has been directed for the effective classification of data and text streams, especially in concept drifting environments. There are some simple methodologies dealing with concept drift. We could use a single incremental classifier updating for each document arriving. The main problem with this method is that the model is strongly built on past data and cannot quickly adapt to the drift. Another simple approach is Weighted Examples (WE) which associates recent examples with a weight in order to force the classifier focus on new data. A third well known approach is Time Window (TW). In this case, we retrain the classifier on the newest N examples in order for the classifier to model only the latest knowledge. The main disadvantages of these techniques are firstly, the assumption that older knowledge is useless for future classification (which does

not apply to all cases) and secondly the fact that the classifier is trained only from a small number of examples (equal to the size of the time window)[1]. Some more advanced methodologies involve ensembles of classifiers (a nice overview of such methods can be found in [6]). The main drawbacks of many of these approaches are the fact that they are demanding in computational sources and that many of them need a step of retraining.

3 Our Approach

3.1 Motivation

A type of concept drift that appears in textual data streams, concerns the appearance of new highly predictive features (words) that do not belong to the original feature vector. In spam filtering for example, new words must be learned by a classifier as new unsolicited commercials come into vogue [2]. In addition, spammers exercise the practice of obfuscating perpetual spam topics by replacing letters with numbers or by inserting irrelevant characters between letters (e.g. viagra becomes v1agra or v.i.a.g.r.a etc). Another application where the importance of words changes over time is personalized news filtering. In this case, the interests of the user might change over time. Therefore, new words that can better discriminate the new interests of the user must be introduced in the feature vector.

So far, the feature vector in text classification approaches has been static. The features that are selected based on an initial collection of training documents, are the ones that are subsequently considered by the classifier during the operation of the system. New words are introduced only with periodic retraining, which includes rebuilding the vocabulary and re-vectorization. Retraining demands the storage of all the documents seen so far, a requirement that might either be infeasible or too costly for textual streams. In addition, retraining has the disadvantage that it does not update the model online as new documents arrive, so it might take some time until it catches up with the drift.

Another point, not adequately discussed in the literature is the fact that in personalized applications an initial training set and consequently the feature space is unavailable. Therefore we need to use flexible algorithms and feature selection techniques that are able to execute in a dynamic feature space that would be empty in the beginning and add features when new documents arrive.

A third point is that in many cases we need classification techniques that are flexible, incremental, and, at the same time require minimum computational sources. Consider for example a web-based personalized newspaper. Each user subscribes to certain topics of interest (Sports, Arts), and a classifier is trained (by taking into consideration user feedback) in order to separate interesting messages for the user. Eventually a classifier per user and per topic is required. Therefore, a large number of computationally undemanding classifiers is required.

[1] The introduction of Adaptive Time Windows actually alleviates these problems [8]

3.2 Framework

Our approach uses two components in conjunction: a) an incremental feature ranking method, and b) an incremental learning algorithm that can consider a subset of the features during prediction. Feature selection methods that are commonly used for text classification are filters that evaluate the predictive power of each feature and select the N best. Such methods evaluate each word based on cumulative statistics concerning the number of times that it appears in each different class of documents. This renders such methods inherently incremental: When a new labeled document arrives, the statistics are updated and the evaluation can be immediately calculated without the need of re-processing past data. These methods can also handle new words by including them in the vocabulary (and fulfil the dynamic feature space requirement discussed earlier) and initializing their statistics. Therefore the first component of our approach can be instantiated using a variety of such methods, including information gain, the x^2 statistic or mutual information [10].

The incremental re-evaluation and addition of words will inevitably result into certain words being promoted to / demoted from the top N words. This raises a problem that requires the second component of the proposed approach: a learning algorithm that is able to classify a new instance taking into account different features over time. We call learning algorithms that can deal with it *feature-based*, because learning is based on the new subset of features, in the same way that in instance based algorithms, learning is based on the new instance.

Two inherently feature based algorithms are Naive Bayes (NB) and k Nearest Neighbors (kNN). In both of these algorithms each feature makes an independent contribution towards the prediction of a class. Therefore, these algorithms can be easily expanded in order to instantiate the second component of our approach. Specifically, when these algorithms are used for the classification of a new instance, they should also be provided with an additional parameter denoting the subset of the selected features. NB for example will only consider the calculated probabilities of this subset, while kNN will measure the distance of the new instance with the stored examples based only on this feature subset. Note, that the framework could apply to any classifier that could be converted into a feature-based classifier.

It is also worth noticing that the proposed approach could work without an initial training set and fulfil the dynamic feature space requirement discussed earlier. This is useful in personalized web-content (e-mail, news, etc.) filtering applications that work based largely on the user's perception of the target class.

Figure 1 presents algorithm UPDATE for the incremental update of our approach. When a new Document arrives as an example of class DocClass, the first thing that happens is to check if it contains any new words. If a new Word is present then it is added to the vocabulary (ADDWORD) and the WordStats of this Word are initialized to zero. Then, for each Word in the Vocabulary we update the counts based on the new document and re-calculate the feature evaluation metric. Finally, the Classifier must also be vertically updated based on the new example and also take into account any new words. For the classification

of a new unlabeled Document, the algorithm selects the top-N words based on their evaluation and then predicts the class of the document by taking under consideration only the selected feature subset.

```
input  : Document, DocClass, Classes, Vocabulary
output: Classifier, Vocabulary, WordStats, Evaluation
begin
    foreach Word ∈ Document do
        if Word ∉ Vocabulary then
            AddWord(Word, Vocabulary)
            foreach Class ∈ Classes do
                WordStats [Word][Class][1] ← 0
                WordStats [Word][Class][0] ← 0
    foreach Word ∈ Vocabulary do
        if Word ∈ Document then
            WordStats [Word][DocClass][1] ← WordStats [Word][DocClass][1] + 1
        else
            WordStats [Word][DocClass][0] ← WordStats [Word][DocClass][0] + 1
    foreach Word ∈ Vocabulary do
        Evaluation ← EvaluateFeature(Word, WordStats)
    Classifier ← UpdateClassifier(Document, DocClass)
end
```

Fig. 1. Algorithm Update

4 Experimental Results

4.1 Feature Selection and Learning Algorithm

The x^2 metric was selected for instantiating the feature evaluation method of the proposed approach, due to its simplicity and effectiveness [10]. We extended the implementation of the x^2 feature evaluation method of Weka [9] with a function that allows incremental updates. As we mentioned in the previous section, there are many other similarly simple metrics that could be used for instantiating our framework. Here, we are not focusing on the effectiveness of different feature evaluation methods, but rather on whether a feature based classifier coupled with an incremental version of such a method is useful in textual data stream classification.

The learning algorithm that was selected for instantiating the learning module of the proposed approach was Naive Bayes. kNN is inefficient for data-streams, as it requires the storage of training examples. NB on the other hand stores only the necessary statistics and is also widely used in text classification

applications. In addition, NB can take advantage of the already stored feature statistics for the purpose of feature ranking and thus integrates easier in the proposed approach. We extended Weka's implementation of NB with a function that accepts a feature subset along with a new instance and uses only the features of this subset for the classification of the instance. Note that we are not focusing on the effectiveness of the specific algorithm. Any incremental machine learning algorithm could be used as long as it is, or, could be transformed to, a feature-based classifier.

4.2 Data Sets

The first requirement of an empirical study of the proposed approach is a data set with documents obtained from a real word textual data stream. We actually experimented with two content filtering domains, spam and news.

For the domain of spam filtering we ideally need real-world spam and legitimate emails chronologically ordered according to their date and time of arrival. In this way we can approximate the time-evolving nature of the problem and consequently evaluate more properly the proposed approach. For that reasons, we used the SpamAssasin (http://spamassasin.apache.org/) data collection because a) Every mail of the collection is available with the headers, thus we were are able to extract the exact date and time that the mail was sent or received, and b) It contains both spam and legitimate (ham) messages with a decent spam ratio (about 20 %). This dataset consists of 9324 instances and initially 40000 features. This datasets represents the so-called gradual concept drift.

For the domain of news filtering we needed a collection of news documents corresponding to the interests of a user over time. As such a collection was not available, we tried to simulate it using usenet posts from the 20 Newsgroups collection[2]. The data set was created to simulate concept drift. The scenario involves a user that over time subscribes to and removes from different general mailing lists (or news feeds) (e.g. sports, science etc) but is interested only on certain subcategories of these mailing lists. Table 1 shows, the particular interests of the user and how her general interests change over time. For example the user is initially interested in sports, but then loses this interest and subscribes in a science mailing list. The user is perpetually interested in driving, while in the last part she also gets into religion issues and at the same time unsubscribes from the hardware list. This dataset consists of 6000 instances and initially 28000 features. In both datasets, we have removed headers and used a boolean bag-of-words approach for the representation of documents. Other methods for IFS presented in the literature like [5] haven't been tested on such high-dimensional datasets. This dataset represents the sudden concept drift[3].

[2] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html
[3] both datasets are available at http://mlkd.csd.auth.gr/datasets.html

Table 1. Interest of user in the various newsgroups over time

	instances		
newsgroup	**1-3000**	**3001-6000**	**4501-6000**
comp.pc.hardware	Yes	Yes	-
comp.mac.hardware	No	No	-
rec.autos	Yes	Yes	Yes
rec.motorcycles	No	No	No
rec.sport.baseball	Yes	-	-
rec.sport.hockey	No	-	-
sci.med	-	No	No
sci.space	-	Yes	Yes
soc.religion.christian	-	-	Yes
alt.atheism	-	-	No

4.3 Methods

To evaluate the effectiveness of our methodology, we applied the basic framework proposed before (IFS) to the three simple learning techniques discussed in the introduction (Simple Incremental Classifier, Weighted Examples, Time Windows).

4.4 Results

We compare the predictive performance of the above methodologies (using x^2 for feature ranking with a classical Incremental Naive Bayes (INB) classifier in the non-IFS approaches and a Feature Based NB classifier for the IFS-enhanced approaches. All methods are executed on the two document collections (spam, news), using as initial training set the first 10, 20 and 30% of the documents. The rest of the documents were used for testing: all methods first predict the class for each new document and then update their models based on the actual class of it. We fixed the number of features to select to 500, as past results have shown that a few hundreds of words are an appropriate size of features. Table 2 shows the results of the experiments. After preliminary experimentation, we concluded that 300 instances was a good window size for the TW method and that a decent way to update the weights in the WE method was $w(n) = w(n-1) + n^2$, where $w(n)$ is the weight of the n-th instance. Note, that we are not focusing on the accuracy of the aforementioned methodologies, but rather on the effect of IFS on them. We first notice that all methods when enhanced with IFS have better predictive performance than the classical approaches in both data sets, all percentages of initial training documents and both metrics. This shows that incremental feature selection manages to catch up with the new predictive words that are introduced over time. In the spam domain, the inclusion of more training data increases the predictive performance of all methodologies due to the inclusion from the beginning of the important features that appear early in the data set. In the news domain on the other hand the inclusion of more training data does

Table 2. Accuracy (acc) and area under the ROC curve (au) for the two data sets and the three different percentages of training documents for three learning methodologies (with and without IFS)

		10%		20%		30%	
Dataset	**Method**	**acc**	**auc**	**acc**	**auc**	**acc**	**auc**
spam	INB	66.06	81.64	51.44	81.53	88.55	93.23
	INB+IFS	86.28	92.48	90.27	95.42	94.02	97.11
	TW	89.71	93.08	90.62	93.03	91.86	92.44
	TW+IFS	90.99	94.42	91.80	94.67	93.56	94.68
	WE	89.76	93.67	92.35	94.60	96.00	97.08
	WE+IFS	93.61	96.75	95.56	98.01	95.81	97.56
news	INB	76.04	87.74	76.06	87.57	74.11	85.28
	INB+IFS	84.07	93.57	84.11	93.53	83.77	93.19
	TW	78.38	86.38	78.41	86.10	78.12	85.80
	TW+IFS	79.27	87.50	79.54	87.55	79.59	87.42
	WE	80.38	88.77	80.00	89.06	78.38	87.63
	WE+IFS	84.98	93.33	85.03	93.15	85.08	93.07

not increase performance significantly as it becomes harder for the classifiers to forget the initial knowledge and adapt to the new predictive features that appear later on. Figures 4a to 4c shows the moving average (over 200 instances) of the prediction accuracy of all methods (with and without IFS) using the first 20% of all messages of the news collection for training[4]. We notice that for the first instances the performance is comparable, but from then on the performance of IFS-enhanced methods becomes and remains much better than simple methods. This happened because at that time-point the user subscribed to new lists and new predictive words appeared. The same thing occurred after the first 3200 examples when the user changed interests for the second time. Non-IFS methods failed to keep up with the new user interests, while IFS-enhanced methods managed to maintain their initial predictive performance. The TW method is the only one that is not significantly affected from IFS, and that can be seen in table 2 (see accuracy TW versus TW+IFS in both datasets) and also from figure 4c. This is mainly because of the small window size that the IFS is applied to. Figures 4e and 4d show the moving average (over 200 instances) of the number of words promoted to/demoted from the top 500 words in both datasets for the INB+FS method. Note that in the news domain, in the beginning more words are promoted to/demoted from the top 500 words as the evaluation scores of already included words continue to change with more training examples, while towards the end they stabilize. The peak in the spam domain is due to the skewness of the collection (a large number of new spam messages arrived at that time point).

SVMs are well known accurate text classifiers and independent of feature selection. Indicatively, we applied an SVM (Weka implementation (SMO), default

[4] The respective figures for the spam corpus are similar

parameter setting (Polynomial Kernel of degree 1, C=1, L=0,001), no initial training) in the news dataset with retraining for every 300 instances and obtained an average accuracy of 70.02%. With TW+IFS method (no retraining) we obtained 77.95% accuracy. Naturally the time needed for the execution of the SVM was much larger (approx. 4 times).

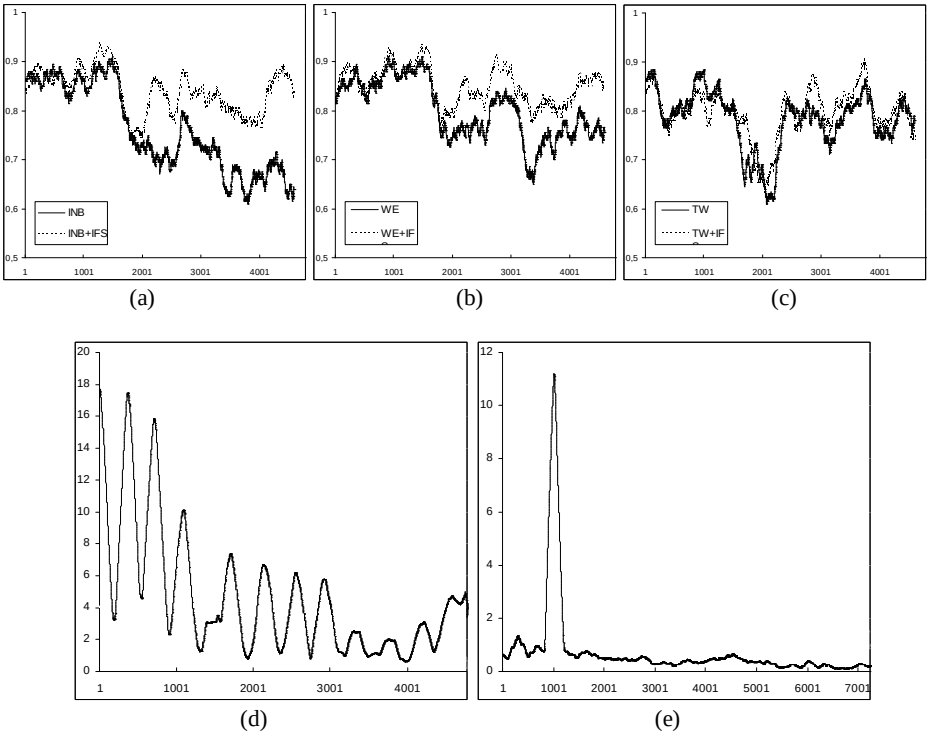

Fig. 2. (a),(b),(c) Moving average of the accuracy for the news domain, and (d),(e) Moving average of the number of words promoted to/domoted from the top 500 words, using the first 20% of all messages of the news collection (d) and spam corpus (e) for training (INB+FS method).

5 Conclusions and Future Work

This paper focused on an interesting special type of concept drift that is inherent to textual data streams: The appearance of new predictive features (words) over time. In the past, this type of concept drift has not been considered by online learning approaches to the best of our knowledge, rather it was confronted with the cumbersome approach of retraining. We presented a computational undemanding approach that combines an incremental feature selection method with

what we called a feature based learning algorithm in order to deal with this problem and we underlined the importance of a dynamic feature space. The experimental results showed that the proposed approach offers better predictive accuracy compared to classical incremental learning and are encouraging for further developments. We also hope that the use of the 20 newsgroups for simulating drifting interests will inspire other researchers for similar experiments. We believe that the proposed approach is a straightforward method for dealing with online learning in high-dimensional data streams, and could be considered as a baseline for comparison with other more complex methods, such as approaches based on ensembles of classifiers, due to its simplicity and effectiveness.

6 Acknowledgements

This work was partially supported by the Greek R&D General Secretariat through a PENED program (EPAN M.8.3.1, No. 03E Δ 73).

References

1. P. Domingos and G. Hulten. Mining high-speed data streams. In *Knowledge Discovery and Data Mining*, pages 71–80, 2000.
2. T. Fawcett. "in vivo" spam filtering: A challenge problem for data mining. *KDD Explorations*, 5(2), December 2003.
3. F. Ferrer-Troyano, J. S. Aguilar-Ruiz, and J. C. Riquelme. Incremental rule learning based on example nearness from numerical data streams. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 568–572, New York, NY, USA, 2005. ACM Press.
4. I. Katakis, G. Tsoumakas, and I. Vlahavas. On the utility of incremental feature selection for the classification of textual data streams. In *10th Panhellenic Conference on Informatics (PCI 2005)*, pages 338–348. Springer-Verlag, 2005.
5. S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333–1356, 2003.
6. M. Scholz and R. Klinkenberg. Boosting classifiers for drifting concepts. *Intelligent Data Analysis (IDA), Special Issue on Knowledge Discovery from Data Streams (accepted for publication)*, 2006.
7. P. Wang, H. Wang, X. Wu, W. Wang, and B. Shi. On reducing classifier granularity in mining concept-drifting data streams. In *ICDM*, pages 474–481. IEEE Computer Society, 2005.
8. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.
9. I. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations.* Morgan Kaufmann, 1999.
10. Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97*, pages 412–420. Morgan Kaufmann Publishers, San Francisco, US, 1997.