Handling Concept Drift in a Text Data Stream Constrained by High Labelling Cost

Patrick Lindstrom
School of Computing
Dublin Institute of Technology
Dublin, Ireland
patrick.lindstrom@dit.ie

Sarah Jane Delany
Digital Media Centre
Dublin Institute of Technology
Dublin, Ireland
sarahjane.delany@dit.ie

Brian Mac Namee
School of Computing
Dublin Institute of Technology
Dublin, Ireland
brian.macnamee@dit.ie

Abstract

In many real-world classification problems the concept being modelled is not static but rather changes over time - a situation known as *concept drift*. Most techniques for handling concept drift rely on the true classifications of test instances being available shortly after classification so that classifiers can be retrained to handle the drift. However, in applications where labelling instances with their true class has a high cost this is not reasonable. In this paper we present an approach for keeping a classifier up-to-date in a concept drift domain which is constrained by a high cost of labelling. We use an active learning type approach to select those examples for labelling that are most useful in handling changes in concept. We show how this approach can adequately handle concept drift in a text filtering scenario requiring just 15% of the documents to be manually categorised and labelled.

Introduction

Concept drift occurs in a data stream when the target concept in a classification problem changes over time. These changes can be due to external circumstances, hidden contexts or even changes in the underlying data distribution (Kubat 1989). Examples of changing concepts can be seen in a variety of real-world applications: weather predictions are affected by seasonal weather variations, customer buying preferences can be influenced by fashion trends or seasonal inclinations, and changing users' interests can impact on information filtering. The challenge in classifying the instances in a continuous data stream is re-training the model to match the changing concepts in the stream.

A number of techniques for dealing with concept drift have been identified in the research (Kubat 1989; Klinkenberg and Joachims 2000; Delany et al. 2005; Kolter and Maloof 2003). However, most of these techniques expect that after classification, the classifier can use some, or all, of the examples classified as new training data. Re-training is only possible if the actual class of these examples is known. Consider for example a spam filter: the true classification of an email can be expected as the user is likely to correct mistakes by moving misclassified spam data out of the inbox to the spam folder and 'recovering' any legitimate emails incorrectly filtered as spam.

However, there are a number of real-world scenarios where this is not feasible. Consider a news analytics application which receives a continuous stream of news articles which it attempts to categorise as of interest to a user[1]. As news and opinions change over time concept drift is likely to be present in this data, and to keep the classifier up to date new labelled documents need to be made available as training data. The expense and effort involved in creating this new training data can be a problem, and is particularly so in text classification problems due to the effort involved in reading and categorising texts.

The contribution of this paper is a novel approach to keeping a classifier up-to-date in text classification domains that feature concept drift, and are constrained by a high cost of labelling examples. Our strategy is to use an *active learning* (Cohn, Atlas, and Ladner 1994) based approach to choose a small number of the most useful examples to label, and to use these examples to re-train the classifier to handle the concept drift. In this way only the examples that are required to keep the classifier up to date are labelled, which greatly reduces the labelling effort required. Our evaluation of the system on both artificial and real data indicates that a classifier can be kept up-to-date with the changes in concept at a labelling cost of only 15% of the documents in the stream. In domains in which large numbers of documents are classified this represents a significant reduction in labelling costs.

The next section of this paper looks at existing research on handling concept drift. After this our proposed approach is detailed, the datasets used in our experiments are described, and our evaluation methodology and results are discussed. Finally, we present the conclusions drawn from our experiments and directions for future work.

Review

Gao et al. (2007) proposes that the main causes of change in concept are either an inherent change in the data stream (such as a new category within a class or a class distribution change), known as a *feature change*; a change in the decision boundary, known as *conditional change*; or a combination of both, known as *dual change*.

These can manifest themselves in two main types of

[1] An example of this type of system which categorises based on sentiment can be viewed at http://sentiment.ucd.ie

change in concept: *sudden shift* and *gradual drift*. Sudden shift occurs when the concept changes abruptly. For example, in a news filtering application, the death of a prominent media figure can make articles about that person relevant to the user, where they were non-relevant before. Gradual drift, on the other hand, occurs when the concept gradually changes from one concept to another. For example, articles about an election might gradually become less relevant to a user after the election.

Addressing changes in concept can be broken down into two subtasks: *concept drift detection* and *concept drift handling*. Drift detection deals with detecting when a significant change in concept has taken place. Concept drift handling is concerned with how the classifier is updated to take account of a change in concept. Feature change, often called novelty detection (Gao et al. 2007), lends itself to automated concept drift detection; whereas conditional change, which is a change in the mapping from the data to the class label, can be impossible to detect without feedback (i.e. labels) from the user.

Most of the work to date on both drift detection and drift handling assumes that the true class of all instances in the data stream will be known shortly after classification (Kubat 1989; Klinkenberg and Joachims 2000; Delany et al. 2005; Kolter and Maloof 2003). In some domains, such as autonomous sentiment analysis, this is an unworkable assumption as the effort required to obtain these true labels is prohibitive. For this reason solutions which reduce the number of labelled instances required to keep the classifier synchronised with the current concept should be sought.

Reducing the need for labelled instances can be achieved by improvements in concept drift detection, concept drift handling, or both. There are approaches to drift detection which do not require all of the instances in the data stream to be labelled. For example using classification confidence (Lanquillon 1999), a statistical distance function (Kifer, Ben-David, and Gehrke 2004) or decision tree statistics (Fan et al. 2004). The goal here is to update the classifier only when there is strong evidence that concept drift has occurred, and so reduce the amount of manual labelling required.

Other work such as Klinkenberg (1999) and Delany et al. (2005), does not attempt to identify the concept change but continuously updates the classifier with new training data. In this scenario the labelling burden can be alleviated by reducing the amount of labelled data needed to update the classifier. Klinkenberg (1999) looks at concept drift handling in a text categorisation problem where only a subset of instances in the data stream are labelled. In the scenario described, exactly which instances are labelled is outside the control of the algorithm. The experimental evaluation indicates that this approach is a good starting point as concept drift can be handled without a fully labelled instance stream. However, there is scope for improvement if the algorithm is allowed to select those instances for which it would be most useful to obtain labels. Active learning (AL) is a powerful way of selecting which instances the classifier would benefit the most from having labelled. AL is a semi-supervised learning technique which is used to build classifiers from large collections of unlabelled examples with the assistance of an oracle, typically a human expert. The oracle is asked to label only those examples that are deemed to be most informative to the training process. Although most AL literature deals with a pool-based setting (where all examples that it is possible to label are present at the beginning of the process) (Cohn, Atlas, and Ladner 1994), there is AL literature that mentions the benefits of using AL to combat concept drift in a data streams (Saunier, Midenet, and Grumbach 2004) but without performing experiments on concept drift datasets.

Zhu et al. (2007) tries to bridge the gap between pool-based AL and concept drift handling. In their approach an ensemble of decision trees is used, and the instances that cause maximal classifier variance within the ensemble are selected for labelling. Huang and Dong (2007) combine AL with a concept drift detection technique similar to that used by Fan et al. (2004). Although their classifier is a decision tree they use a Naïve Bayes based uncertainty sampling approach to select examples for labelling. Our work differs from these in that our application focus is text classification and we concentrate on conditional change, where the decision boundary has changed due to external influences meaning feedback on class labels is required to handle the change.

Approach

Our proposed approach to dealing with concept drift adopts a fixed-size *sliding window* approach (Kubat 1989). The documents arrive as a stream of unlabelled instances, are grouped into fixed size batches, and are then presented to the classifier for classification. After classification a subset of the documents in the batch is selected using a *selection strategy* which chooses the most useful documents to label. These are presented to the oracle who is asked to label them. These accurately labelled documents are added to the existing training set from which an equivalent number of the oldest documents are removed. The classifier is then rebuilt and the next batch of documents are presented for classification. This process repeats indefinitely.

The classifier used in our approach is a Support Vector Machine (SVM) (Vapnik 1995) which has been shown to be especially suitable for text classification (Joachims, Nedellec, and Rouveirol 1998). An SVM is a binary classifier which identifies a separating hyperplane which maximises the margin between instances of the different classes. Existing work in AL has suggested that a good selection strategy for an SVM is to choose examples for labelling that are closest to the separating hyperplane, as the classifier has least confidence in its predictions for these examples (Tong and Koller 2002; Xu et al. 2003). The selection strategy used in our approach is based on this work and we select a subset of n examples closest to the separating hyperplane from the current batch of documents to present to the oracle for labelling. We refer to this as *decision value* selection.

Datasets

Evaluating research on concept drift can be problematic due to the lack of benchmark concept drift datasets

(Narasimhamurthy and Kuncheva 2007). Existing research into handling concept drift generally has two sources of data: real-world and artificial datasets.

Real-world datasets are collected from naturally occurring processes. Examples include financial (Abdullah and Ganapathy 2000), biological (Tsymbal et al. 2006) and spam filtering (Delany et al. 2005) data. There are problems associated with using these datasets for benchmarking in that often such datasets, which can include personal or financial information, are not made publicly available for confidentiality reasons. Also, it can often be hard to ascertain when and why the concept drift occurred in a real-world dataset.

Due to the difficulties associated with real-world datasets, researchers often use artificially created datasets (Kifer, Ben-David, and Gehrke 2004; Fan et al. 2004; Huang and Dong 2007; Zhu et al. 2007). Artificial datasets can be one of two types: *synthetic* datasets and *drift-induced* datasets. Synthetic datasets are entirely artificial and can be created using frameworks such as STAGGER (Schlimmer and Granger 1986), the moving hyperplane (Kolter and Maloof 2003) and Narasimhamurthy's framework (Narasimhamurthy and Kuncheva 2007). Drift-induced datasets are created by artificially adding concept drift to existing datasets, for example by changing the class distribution within a data stream (Zhu et al. 2007).

In this paper we use both types of data: real-world datasets from the spam filtering domain and artificially created drift-induced datasets generated from large corpora of text documents. The real-world datasets used in our experiments are two spam filtering datasets (*Spam Dataset 1* and *Spam Dataset 2*) introduced in Delany et al. (2005). The classification task is to distinguish between emails relevant to the user ('ham') and emails not relevant to the user ('spam'). The details of class distributions of these two datasets are given in Table 1.

(a) Spam Dataset 1

Topic	**Size**
spam	9,364
ham	1,572
Total:	**10,936**

(b) Spam Dataset 2

Topic	**Size**
spam	8,071
ham	1,193
Total:	**10,264**

Table 1: Topic distribution for the real-world spam datasets

The text corpora used to generate the drift-induced datasets are the *Reuters*[2] and *20 Newsgroups*[3] collections. All documents in both collections are categorised as one of a set of predefined topics. A subset of the topics are categorised as *relevant* to a reader at a particular time. All documents in the remaining topic categories are considered *non-relevant* to the reader. The classification problem is to distinguish between the relevant and non-relevant categories. Drift is induced in the *relevance concept* by changing the topics which are considered relevant over time. These are therefore drift-induced datasets with conditional concept change.

(a) Reuters corpus

Topic	**Size**
acq	2,429
earn	3,968
others	15,162
Total:	**21,559**[4]

(b) 20 Newsgroups corpus

Topic	**Size**
religion	2,997
computers	5,000
others	12,000
Total:	**19,997**

Table 2: Topic distributions for the drift induced datasets

This approach to generating concept drift datasets is similar to that used by Lanquillon (1999) in that the documents are divided across relevant/non-relevant classes associated with topic categories. However, it differs in a number of respects. Firstly, the datasets we create are significantly bigger. Secondly, the class distribution in the data stream is not fixed, whereas Lanquillon's work assumed a balanced number of relevant and non-relevant documents at all times. Lastly, we include documents in each training set from non-relevant topics that will become relevant at a later stage in the processing of the data stream. Lanquillon changes the relevance concept to previously unseen topics, only introducing documents from these topics into the data stream once they are to be considered relevant. This effectively assumes that a reader's interest can only change to new topics of interest, rather than switching between existing topics. This latter scenario, that we address, is more realistic and can be considered a more difficult classification problem.

The documents in each corpus were sorted chronologically to simulate a data stream and two datasets were built from each corpus: one that exhibits a sudden shift in concept; and one that exhibits a gradual drift in concept. Each document was labelled using a probability function to determine its classification (relevant or non-relevant) based on its topic. The topics used for simulating concept drift, and the frequencies of each topic, are outlined in Table 2 for both corpora.

Figure 1 shows how the probability distribution affects the conditional probabilities of relevance for the datasets. Consider, for example, the Reuters corpus. For documents in the early part of the stream the *acq* topic was considered to be the relevant topic with $P(relevant|acq) = 1$ and *earn* and all other topics were considered non-relevant, i.e. $P(relevant|topic) = 0$ where $topic \neq acq$. Then, at a particular point in the data stream, the *earn* topic becomes relevant leading $P(relevant|earn)$ to increase and $P(relevant|acq)$ to decrease. For sudden shift, this is an immediate change, i.e. $P(relevant|earn) = 1$ and $P(relevant|acq) = 0$, while for gradual drift $P(relevant|earn)$ increases linearly towards 1 and $P(relevant|acq)$ decreases linearly towards 0 over a specified number of documents in the data stream.

Evaluation

The goal of this evaluation is to show that concept drift handling can be achieved in a text classification domain when

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

[3]http://people.csail.mit.edu/jrennie/20Newsgroups

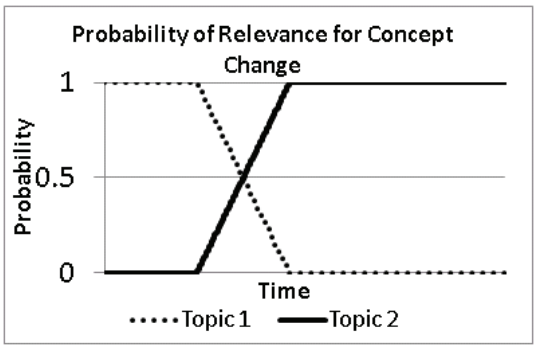


Figure 1: Probabilities used to introduce concept change in the drift induced datasets.

only a carefully selected subset of the documents in the data stream are labelled with their true classifications and made available for re-training. As we are aiming to minimise the need for labelled instances, the number of documents chosen, n, should be as small as possible while maintaining high classification generalisation accuracy. The selection strategy used to select the documents for labelling is therefore the most important factor in the effectiveness of our approach.

Methodology

The documents in each dataset were represented as a bag-of-words with stop-word removal and Porter's stemming applied. In each experiment the first 150 documents in the data stream of each class were selected as initial training data. The rest of the documents in the data stream were grouped into batches of 100 documents to be presented to the classifier for classification. After classifying the documents in each batch the classifier is rebuilt using a training set augmented with n documents from the latest test batch, each of which has its true class label associated with it. Since all of the datasets used have imbalanced class distributions (e.g. many non-relevant documents compared to few relevant ones, see Table 2) simply replacing the oldest training documents with n new ones would result in training sets becoming heavily skewed towards the majority class over time. Instead, the documents removed from the training set at each iteration are the oldest documents of the same class as the n new documents. As a result the class distribution of the training set is kept constant.

Using the *decision value* selection strategy the n documents selected for addition to the training set are those that were positioned closest to the SVM's maximal-margin hyperplane. We expect that, as the classifier has the least confidence in its predictions for these documents, they are most indicative of the changed concept and that the system would benefit most from their true class labels.

As a baseline to compare this selection strategy against we also use a *random sampling* approach which randomly picks n documents from each batch for labelling. As random sampling is non-deterministic, the random sampling experiments were run 10 times each, and the average performance for each batch across the 10 runs is presented.

Evaluation Measures

The datasets have a high class imbalance so the average class accuracy per batch is used to measure performance, calculated as:

$$avgAcc_i = \frac{\sum_{j=1}^{|C|} \frac{m_{ij}}{c_{ij}}}{|C|}$$

where C is the set of possible classes, m_{ij} is the number of correctly classified instances of class j in batch i and c_{ij} is the total number of documents of class j in batch i. It is worth noting that in rare cases only examples of the majority class will be present in a batch and so $|C| = 1$.

The process of repeatedly updating the classifier and classifying subsequent batches is evaluated by calculating the aggregate average class accuracy after each batch, and plotting how this changes over time. The aggregate average class accuracy is calculated as:

$$aggAcc_B = \frac{\sum_{b=1}^{B} avgAcc_b}{B}$$

where B is the current batch number and $avgAcc$ is as described above. Figure 2 shows examples of how the aggregate average class accuracy is plotted over time.

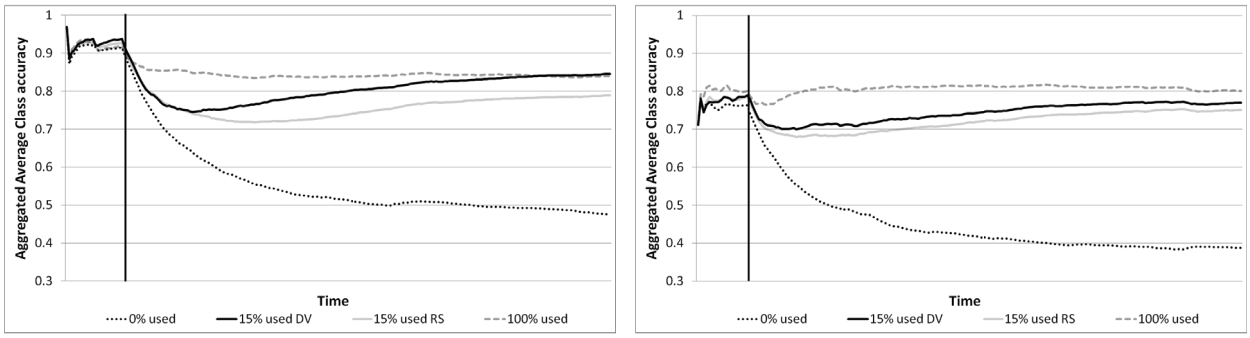
Results

Figure 2 shows the results of applying our concept drift handling approach to the Reuters and 20 Newsgroup datasets when concept shift is induced with $n = 15$ (although in experiments many values of n were tested, results for $n = 15$ are shown as they give a good balance between the labelling effort and the performance achieved). On each graph *0% used* refers to a scenario in which no concept drift handling is used, while *100% used* refers to a sliding window approach that uses true labels for all examples in each batch (i.e. n = 100). *15% used DV* and *15% used RS* refer to the scenarios where decision value and random sampling selection strategies, respectively, are used. The point where the concept shift occurs in the data stream is marked with a vertical line.

It is clear from Figure 2 that when no concept drift handling is performed (*0% used*) the classifier performance deteriorates rapidly once the shift in concept occurs. Conversely, when all examples in each batch are labelled and used to update the classifier (*100% used*) the performance of the classifier remains stable even after the concept shift occurs.

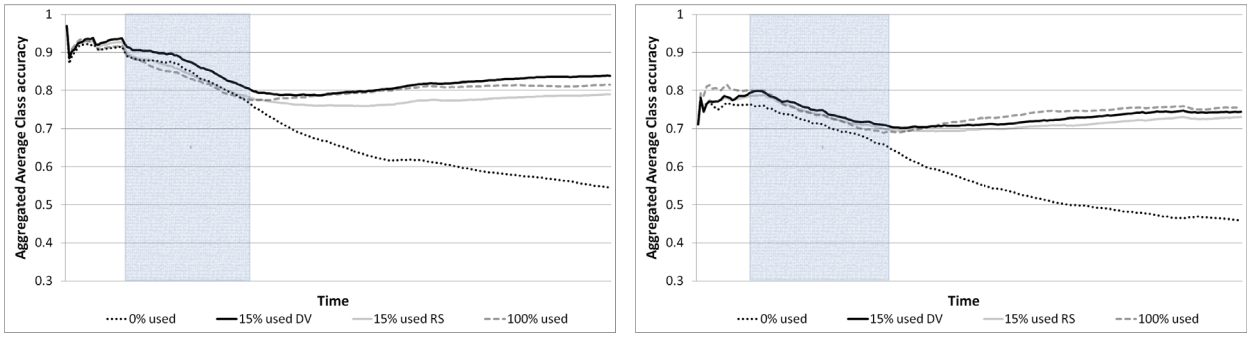
Both the random sampling (*15% used RS*) and decision value (*15% used DV*) approaches to concept drift handling perform well. This indicates that in the sudden shift scenario fully labelled data streams are not required in order to maintain high levels of classifier performance. Furthermore, the decision value approach is considerably better than the random approach for both datasets, and is almost comparable to using 100% of the data.

The pattern is very similar when a gradual drift in concept is induced into the two datasets. The results of these experiments are shown in Figure 3, with the region over which the



(a) Reuters shift dataset (b) 20 Newsgroup shift

Figure 2: Showing the effect of different concept handling scenarios on a dataset where concept shift occurs. The point where the shift occurs is marked with a vertical line.



(a) Reuters drift dataset (b) 20Newsgroup drift dataset

Figure 3: Showing the effect of different concept handling scenarios on a dataset where concept drift occurs. The region over which the concept drift occurs is shown as shaded.

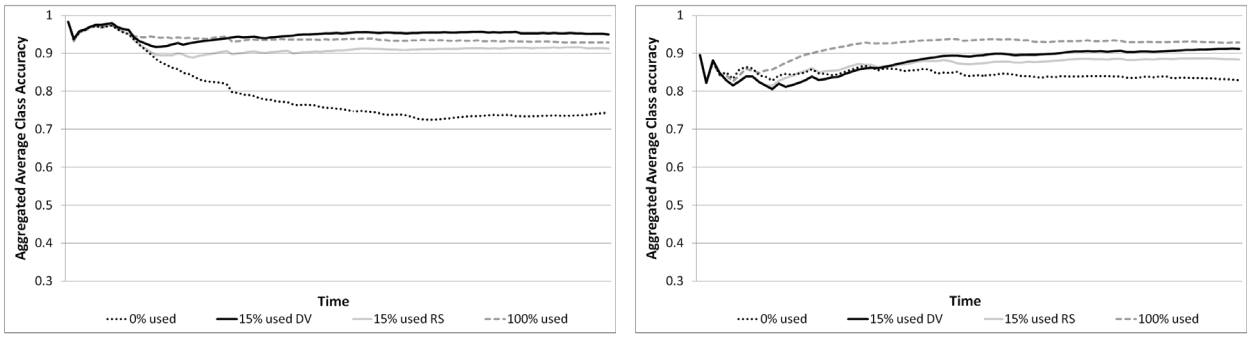
gradual drift occurs shown shaded in grey. Once the concept starts changing the performance of the non-updating classifier (*0% used*) drops sharply. When 100% of the data is used for re-training the performance improves over time, but not as quickly as in the sudden shift scenario. We believe this is because the classifier cannot perform well until the concept has stabilised. This is also evident in the two scenarios in which only 15% of the data is labelled. Again, using labels for only a small percentage of the data, performance similar to that when 100% of the data is labelled can be achieved. For both datasets the decision value approach noticeably outperforms random sampling.

In the final set of experiments the concept drift handling approaches were applied to the real-world spam datasets. The concept drift present in Spam Dataset 1 (Figure 4(a)) is obvious from the fact that the classification accuracy shown on the *0% used* line in Figure 4(a) deteriorates over time. Random sampling gives results close to using all of the data in the batch. However, using decision value sampling produces a result better than using all the data in the batch. This is an interesting result and we suspect it arises because there is some noise in the data stream and the decision value sampling avoids the noisy instances to produce a better classifier.

For Spam Dataset 2 (Figure 4(b)) there does not seem to be noticeable drift as the *0% used* line does not dip significantly. However, the *100% used* line indicates improved performance can be achieved by updating the classifier, which suggests some change in concept. Similarly to the previous cases, the graph also shows that labelling a random 15% of the data per batch improves the performance over time, but using the decision value sampling technique improves the result even further.

Conclusions and Future Work

The goal of this work is to reduce the need for labelled instances in handling changing concepts in text classification problems. The need for labelled instances is reduced by using an active learning based approach to selectively sample the most useful instances for labelling each time the classifier is to be retrained. Experiments were performed on two text datasets from commonly used text corpora in which drift was induced, and on two real-world spam filtering datasets. On all of these datasets it was possible to maintain classification accuracies comparable with those achieved using full labelling of the data stream by labelling only 15% of the incoming documents - a significant reduction in the labelling effort required. In all examples the decision value selection strategy boosted performance more than the random sampling selection strategy showing the usefulness of targeted



(a) Spam Dataset 1

(b) Spam Dataset 2

Figure 4: Showing the effect of different concept handling scenarios on the real-world spam datasets

selection of documents for labelling.

In the future we intend to pair our concept drift handling strategy with concept drift detection approaches and in this way further reduce the need for labelled instances. This will be of particular benefit in situations where the level and type of drift is unknown. It is evident from the beginning of Figures 2 and 3 that the concept is stable before the concept change occurs. It could be argued that up to this point the classifier does not need to be updated and asking for labels for a few instances every batch is wasteful. An improvement to the algorithm would be to either try to detect concept change and only update the classifier after a change in concept has been confirmed or use a heuristic to vary the value of n (the number of instances sampled per batch).

References

Abdullah, M., and Ganapathy, V. 2000. Neural network ensemble for financial trend prediction. *TENCON 2000. Proceedings* 3.

Cohn, D.; Atlas, L.; and Ladner, R. 1994. Improving generalization with active learning. *Machine Learning* 15(2):201–221.

Delany, S. J.; Cunningham, P.; Tsymbal, A.; and Coyle, L. 2005. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems* 18(4–5).

Fan, W.; Huang, Y.; Wang, H.; and Yu, P. S. 2004. Active mining of data streams. In *Proc. 4th SIAM ICDM*, 457.

Gao, J.; Fan, W.; Han, J.; and Yu, P. S. 2007. A general framework for mining concept-drifting data streams with skewed distributions. In *Proc. SDM'07*.

Huang, S., and Dong, Y. 2007. An active learning system for mining time-changing data streams. *Intell. Data Anal.* 11(4):401–419.

Joachims, T.; Nedellec, C.; and Rouveirol, C. 1998. Text categorization with support vector machines: learning with many relevant. In *Machine Learning: ECML-98*, 137–142.

Kifer, D.; Ben-David, S.; and Gehrke, J. 2004. Detecting change in data streams. In *Proceedings of the Thirtieth Int. Conf. on VLDB*, volume 30, 191.

Klinkenberg, R., and Joachims, T. 2000. Detecting concept drift with support vector machines. *Proc. 7th ICML* 11.

Klinkenberg, R. 1999. Learning drifting concepts with partial user feedback. *Beitrge zum Treffen der GI-Fachgruppe* 1(3):44–53.

Kolter, J., and Maloof, M. 2003. Dynamic weighted majority: a new ensemble method for tracking concept drift. In *3rd IEEE ICDM*, 123–130.

Kubat, M. 1989. Floating approximation in time-varying knowledge bases. *Pattern recognition letters* 10:223–227.

Lanquillon, C. 1999. Information filtering in changing domains. *In Workshop on Machine Learning for Information Filtering, IJCAI'99* 41–48.

Narasimhamurthy, A., and Kuncheva, L. I. 2007. A framework for generating data to simulate changing environments. In *Procs IASTED, Artificial Intelligence and Applications*, 384–389.

Saunier, N.; Midenet, S.; and Grumbach, A. 2004. Stream-based learning through data selection in a road safety application. In *STAIRS 2004, Proceedings of the Second Starting AI Researchers' Symposium*, volume 109, 107–117.

Schlimmer, J. C., and Granger, R. H. 1986. Incremental learning from noisy data. *Machine Learning* 1:317–354.

Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2:45–66.

Tsymbal, A.; Pechenizkiy, M.; Cunningham, P.; and Puuronen, S. 2006. Handling local concept drift with dynamic integration of classifiers. In *19th IEEE International Symposium on CBMS*, 679–684.

Vapnik, V. N. 1995. *The nature of statistical learning theory*.

Xu, Z.; Yu, K.; Tresp, V.; Xu, X.; and Wang, J. 2003. Representative sampling for text classification using support vector machines. In *Advances in Information Retrieval*, volume 2633 of *LNCS*.

Zhu, X.; Zhang, P.; Lin, X.; and Shi, Y. 2007. Active learning from data streams. In *Procs 7th IEEE ICDM*, 757–762.