2013

Drift Detection using Uncertainty Distribution Divergence

Patrick Lindstrom
Technological University Dublin, patrick.lindstrom@tudublin.ie

Brian Mac Namee
Technological University Dublin, brian.macnamee@tudublin.ie

Sarah Jane Delany
Technological University Dublin, sarahjane.delany@tudublin.ie

Recommended Citation

Lindstrom, P., Mac Namee, B. & Delany, S. (2013). Drift detection using uncertainty distribution divergence. *Evolving Systems*,vol. 4, pp.13–25. doi:10.1109/ICDMW.2011.70

Noname manuscript No.
(will be inserted by the editor)

Drift Detection using Uncertainty Distribution Divergence

Patrick Lindstrom · Brian Mac Namee · Sarah Jane Delany

Received: date / Accepted: date

Abstract Data generated from naturally occurring processes tends to be non-stationary. For example, seasonal and gradual changes in climate data and sudden changes in financial data. In machine learning the degradation in classifier performance due to such changes in the data is known as *concept drift* and there are many approaches to detecting and handling it. Most approaches to detecting concept drift, however, make the assumption that true classes for test examples will be available at no cost shortly after classification and base the detection of concept drift on measures relying on these labels. The high labelling cost in many domains provides a strong motivation to reduce the number of labelled instances required to detect and handle concept drift. Triggered detection approaches that do not require labelled instances to detect concept drift show great promise for achieving this. In this paper we present Confidence Distribution Batch Detection (CDBD), an approach that provides a signal correlated to changes in concept without using labelled data. This signal combined with a trigger and a rebuild policy can maintain classifier accuracy which, in most cases, matches the accuracy achieved using classification error based detection techniques but using only a limited amount of labelled data.

Keywords concept drift · explicit drift detection · labelling cost · classifier confidence

1 Introduction

A key assumption in supervised machine learning is that the data used to train a classifier is representative of the data a classifier will later encounter. Data gathered from real life processes, however, can vary over time. For example, seasonal changes can affect classification of customer spending habits, or the sudden occurrence of

P. Lindstrom · B. Mac Namee · S.J. Delany
School of Computing
Dublin Institute of Technology,
Dublin, Ireland
E-mail: first-name.second-name@dit.ie

major events – such as elections or the introduction of new laws – can affect news filtering models. Using a static classifier in such domains is inadequate as the data is exhibiting a phenomenon known as *concept drift*. A concept can be defined (in the machine learning context) as a set of instances generated by the same underlying function (Gama et al 2004). Concept drift, then, occurs when this underlying function changes for some reason.

According to Kuncheva (2009) there are two approaches to handling concept drift. The first approach, which we refer to as *continuous rebuild*, does not attempt to identify when drift is occurring but continuously and regularly updates the classifier assuming that this will allow any drift that occurs to be handled (e.g. (Kubat 1989)). The second kind of approach, which we refer to as a *triggered rebuild*, explicitly detects when a change in concept is occurring by monitoring a suitable indicator, and only then updates the classifier (e.g. (Lanquillon 1999)). There are many examples in the literature showing how both continuous and triggered rebuild approaches can successfully handle the problem of concept drift.

The majority of continuous and triggered rebuild approaches to concept drift, however, require that the true class of instances presented to the classifier for classification become available shortly after classification occurs (Kuncheva 2009; Kubat 1989; Gama et al 2004; Nishida and Yamauchi 2007). While in many domains this is not a restriction (for example in short term stock market predictions), in domains where labelling instances with their true class has a high cost, these approaches are not feasible. For example, consider a news analytics application that receives a continuous stream of news articles and attempts to determine which pre-defined category each article belongs to. It would be unreasonable to ask an expert to provide the true label for each document given such a large volume, and so true labels will only become available when explicitly sought. The situation is similar in most document filtering tasks (Lanquillon 1999).

This high labelling cost provides a strong motivation to reduce the number of labelled instances required in techniques for handling concept drift. While the number of labels required by continuous rebuild approaches can be reduced by using sampling (e.g. (Lindstrom et al 2010; Žliobaite et al 2011)), triggered rebuild approaches also offer significant potential to reduce the number of labelled instances required.

In this article we present *Confidence Distribution Batch Detection* (CDBD); an earlier version of this study appeared in (Lindstrom et al 2011). CDBD is a concept drift handling approach that explicitly detects changes without requiring the true classes of test instances. CDBD compares the distribution of classifier output confidences in a batch of test examples to a reference distribution constructed from training data, and uses this comparison to generate a measure of concept drift. When this measure is above a given threshold, concept drift is deemed to have taken place, and the classifier is updated. CDBD only requires labelled data to update the classifier once concept drift has been identified, and so using CDBD can significantly reduce the overall amount of labelled data required to keep a classification model up to date. Furthermore, because CDBD measures the occurrence of concept drift based on classifier output alone, it can be used with any classifier capable of producing confidence scores. In a series of experiments using eight text classification datasets we show that using CDBD gives comparable results to other drift handling approaches, while using a smaller amount of labelled instances.

The remainder of this article is organised as follows. Section 2 discusses the current state-of-the art in concept drift handling, and in particular the issue of high labelling cost. Section 3 describes the CDBD approach. Section 4 describes how CDBD has been evaluated on eight text classification problems and compares the performance of this approach to existing concept drift handling techniques. Finally, conclusions and proposed directions for future work are presented in Section 5.

2 Background

In ideal classification scenarios we expect the *stationary assumption* to hold – the data that classifiers are trained on will be representative of the data the classifier will encounter in the future – which allows for good generalisation on unseen instances. Real classification scenarios, however, are rarely ideal, and in most cases concept drift occurs due to changes in the natural underlying function from which data instances arise. Gao et al (2007) identify three ways in which the underlying data generation function can change to give rise to concept drift. If P is the probability of a given event, x is a feature vector and y is a class label, the causes of concept drift can be:

- **Feature change**: a change in the probability of the occurrence of a particular set of feature values, i.e. a change in $P(x)$.
- **Conditional change**: a change in the conditional probability of a class given a particular set of feature values, i.e. a change in $P(y|x)$.
- **Dual change**: a feature and conditional change, i.e. a change in both $P(x)$ and $P(y|x)$.

When concept drift occurs, a concept drift handling technique is required in order to keep a classification model up to date. Concept drift handling techniques attempt to update the classification model when drift occurs, using recent data representative of the changed concept, so as to maintain high generalisation accuracy. As described in Section 1 the different concept drift handling techniques are categorised into two main groups based on how they deal with the problem of recognising that concept drift has occurred. *Continuous rebuild* approaches ignore this issue, and simply update the classification model at regular intervals. *Triggered rebuild* approaches, on the other hand, only update the classification model when a significant change in concept is indicated by some explicit measure of concept change. This section will first describe the state of the art in continuous rebuild and triggered rebuild approaches for handling concept drift. Following this, Section 2.3 will describe how these approaches have been adapted to handle scenarios that are constrained by high labelling costs.

2.1 Continuous Rebuild

The most common continuous rebuild approach to handling concept drift is to periodically retrain the classifier using a set of recent instances as the new training data. This approach is known as the *sliding window*. Using a fixed-size sliding window (in which the size of the set of recent instances used to retrain the classifier is always the same) has been shown to be a simple and effective way to handle

concept drift (Kubat 1989). Using a fixed-size window, however, requires a trade-off between maximising the use of historic data to fine-tune classifier performance and prioritising new data so as to counteract the impact of concept drift.

Adaptive-sized sliding window approaches use an explicit indicator of the occurrence of concept drift to adjust the size of the set of recent data used to update the classifier. These approaches attempt to use a long data horizon to update the classifier when the concept is stable, but to collapse this horizon to just the most recent data when a change in concept is suspected. Klinkenberg and Renz (1998) developed a seminal window resizing heuristic based on monitoring the error rate, precision and recall achieved by a classifier on batches of test instances. There have since been many other viable error-based heuristics used in adaptive-sized sliding window approaches to handling drift (e.g. (Gama et al 2004)).

2.2 Triggered Rebuild

Triggered rebuild approaches to handling concept drift monitor the value of an indicator which is believed to be correlated to a change in concept. A change in concept is flagged when the value of the indicator moves above some threshold. Klinkenberg and Renz (1998) categorized concept change indicators into three groups, each derived from a different source: using properties of the classifier, using properties of the data, and using properties of the classification output.

The first set of indicators are derived from the internal workings of the classifier, with decision trees being particularly common. Decision tree characteristics such as leaf changing statistics (Fan et al 2004) and expected loss (Fan et al 2004; Huang and Dong 2007) have been found to be well correlated with changes in concept. Such approaches are obviously limited to working with a single classifier type, however.

The second group of concept drift indicators are derived from the feature values in the data, and tend to be domain specific. For example, concept drift in textual data streams can be identified by monitoring word frequencies (Swan and Allan 1999) and the formation of new word clusters (Hsiao and Chang 2008; Spinosa et al 2007). Kifer et al (2004) introduce a more generic approach that uses a two window paradigm to detect changes in feature distribution. The distribution of a single feature inside a reference batch is compared to the distribution inside the test batches to determine if the data in both batches is likely to have been generated by the same underlying process. This is achieved using a statistical distance function based on Chernoff bounds. Sebastio and Gama (2007) take a similar approach but use Kullback-Leibler divergence to measure the difference. While both approaches have been shown to be effective they are unlikely to be suitable in high-dimensional classification problems such as text classification as a single feature is unlikely to yield a concept drift signal of sufficient strength and reliability.

The final group of indicators are those derived from the output of a classifier. Kuncheva (2009) enhances the sliding window heuristic in Klinkenberg and Renz (1998) to create an explicit detection algorithm, Window Resize Algorithm for Batch Data (WRABD). WRABD monitors the error rate and flags a change in concept if there is a significant change in the error rate. The advantages of using classifier output as a concept change indicator are that it is classifier and data

independent, and does not presuppose knowledge about which feature(s) might be indicative of a change in concept if monitored.

2.3 Handling Concept Drift When Label Costs Are High

The continuous rebuild approaches to handling concept drift described in Section 2.1 rely on the full availability of true class labels for test instances shortly after classification so that the classifier can be continuously retrained. Similarly, the classifier-output-based triggered rebuild approaches described in Section 2.2, require the full availability of true class labels so that classifier error rate can be calculated.

In certain real life scenarios, e.g. news filtering, the full availability of true class labels is unrealistic due to the high cost of labelling (accessing true labels requires an explicit, time-consuming manual intervention). In such scenarios concept drift handling approaches that do not require full access to true class labels are required.

There are modifications to continuous rebuild approaches suggested in the literature that specifically address the issue of label cost. In particular, active learning (AL) techniques (Cohn et al 1994) can help maintain classification accuracy in the presence of concept drift while maintaining a low labelling cost, by only labelling the test instances believed to be of most benefit in updating the classifier. Lindstrom et al (2010) propose a sliding-window-based approach that samples a fixed number of instances closest to the classifier decision boundary from every test batch and retrains the classifier periodically using a training set augmented with these instances. Zhu et al (2007), similarly, use the variance of the base classifiers in an ensemble classifier to sample a fixed number of instances in each batch of data while updating the ensemble periodically to handle concept drift.

Žliobaite et al (2011) present an approach that processes the data at an instance level, rather than in batches. The approach uses a sampling heuristic which ensures that instances close to the decision boundary are more likely to be labelled when a change in concept occurs. Instances further from the decision boundary are sampled when the concept is stable. The label budget is a parameter to the sampling heuristic ensuring that the labelling budget is spread fairly evenly over the stream. There are also approaches which combine semi-supervised learning with clustering to deal with streams of data, where only a fixed percentage of the stream is labelled (Masud et al 2008; Woolam et al 2009) . The task is to classify the stream with high accuracy and handling concept drift, whilst only using a small amount of labelled data.

While the approaches described above have all been shown to effectively handle concept drift without requiring full access to true class labels for test instances, they all do so using a fixed sample rate (e.g. 10% of the test data). If the concept does not change frequently, the label cost of keeping the classifier up to date can still be high as labelled instances are required regardless of whether or not the concept is changing. Conversely when the concept does change, the low sampling rate slows down adaptation to the new concept as it can take some time to collect sufficient examples to model the new concept. Triggered rebuild approaches can overcome these difficulties and only require labelled data when a change in concept is suspected.

The triggered rebuild techniques based on classifier properties and data distributions described in Section 2.2 do not require full access to true class labels and can be used in scenarios constrained by high labelling costs. These techniques are, however, heavily dependent on classifier type and application specific data structures respectively, and so, more generic approaches based on classifier output are preferable. Although there are not many examples of such techniques in the literature there are some that have been shown to be successful.

One of the earliest works in autonomous text classification systems (Lewis 1995) uses an estimated classifier accuracy measure based on a combination of classifier output and class membership probabilities which can be used to notify an expert when these estimates indicate a problem with handling new data. Žliobaite (2010) uses a window paradigm to compare the posterior class probabilities in a reference window to the probability of class memberships in the current test window, using statistical tests to flag significant differences, and hence the occurrence of concept drift. Although these approaches have been shown to be capable of detecting concept drift without requiring full labelling of test datasets neither develop an end-to-end system which detects a change, updates the model and evaluates the resultant classification accuracy. Lanquillon (1999) does develop an end-to-end approach which estimates the classifier output range in which classifications can be deemed confident and flags a change in concept when the number of predictions in the confident range for a test batch exceeds a threshold. However this approach is only evaluated on very simplistic artificial data.

Before presenting the CDBD approach that has been designed to address these shortcomings, it is important to consider the kinds of concept drift (outlined at the beginning of Section 2) that are feasible to detect using classifier-output-based triggered rebuild approaches that do not require extensive labelled data. Classifier outputs will act as a strong proxy measure for *feature change* (i.e. a change in the probability of the occurrence of a particular set of feature values). So, for example, in an email filtering task the occurrence of a new kind of spam email in the data presented to a classifier will result in changes in classifier output, and a subsequent concept drift detection.

Conditional change (i.e. a change in the conditional probability of classes given particular feature values), however, is not detectable without using labelled data, as measures of changes in classifier output themselves cannot act as a proxy for such change. So, for example, in an email filtering task when a decision is made by a user that a particular newsletter is no longer interesting and should be considered spam, there is no way to trigger an associated concept drift detection using classifier-output-based approaches without the use of extensive labelled data. Continuous rebuild approaches can obviously cope in these kind of concept drift scenarios, although with the associated issues outlined at the beginning of this section. In the results presented in Section 4, however, it will be shown that the CDBD approach can handle the concept drift present in a real-world email filtering dataset which, like all real datasets that exhibit concept drift, is likely to contain a combination of feature change and conditional change.

3 Confidence Distribution Batch Detection

The Confidence Distribution Batch Detection (CDBD) approach uses a two window paradigm (similar to Kifer et al (2004); Sebastio and Gama (2007); Žliobaite (2010)) coupled with a distribution divergence measure (similar to Sebastio and Gama (2007)) applied to an indicator, the classifier confidence, that is classifier and data independent. The usage scenario envisaged is that a constant stream of test instances (divided into batches) arrive at a classifier (trained using a set of initial labelled training examples) to be filtered. It is expected that, while labels for test instances are expensive to obtain and so labelling effort should be minimised, true class labels for test instances are available immediately when explicitly requested.

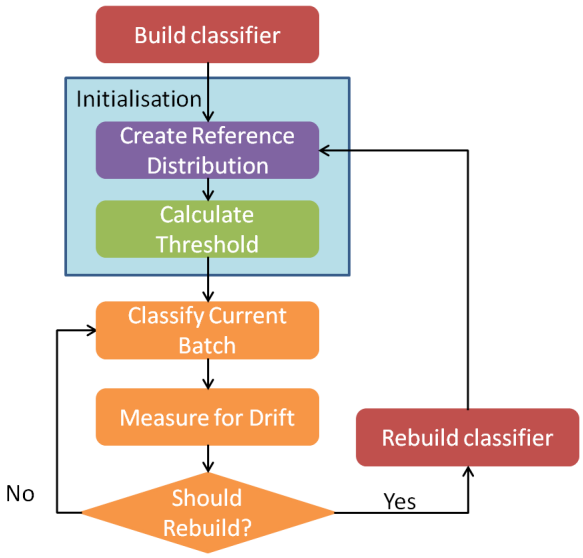

Fig. 1 An overview of the CDBD approach

Figure 1 shows an overview of the CDBD process. At a high level CDBD monitors an indicator for the occurrence of concept drift and when it triggers the classifier is rebuilt using recent data. The important components of the approach are thus: the classifier, the indicator signal, the trigger used to flag the occurrence of concept drift and a process for rebuilding the classifier.

CDBD can be used with any classifier that produces a score that can be interpreted as an estimate of confidence that the prediction made by the classifier is correct, such as k-NN and decision trees. In the current implementation of CDBD a *support vector machine* (SVM) (Vapnik 1995) with a linear kernel is used, as the linear kernel has been found to be very suitable for text classification (Yang and Liu 1999). The confidence score produced by the SVM is a function of the distance between a test instance and the classification hyperplane.

The indicator used in CDBD is a measure of the divergence between the distribution of the classifier outputs for a batch of test instances, $batch_i$, and the distribution of classifier outputs in a batch of reference instances, $batch_{ref}$. This

is based on the expectation that when feature change occurs, the distributions of classifier outputs will change significantly. The reference batch used in the current implementation of CDBD is the first batch of instances classified immediately after the classifier is trained, and the distribution is generated by discretising the values of the indicator over the possible range of values. The bins used were selected by first identifying a range in which most of the classifier output lay, then dividing the range into uniformly sized bins. Initial experiments seemed to indicate that between 7 and 13 bins produced a signal which was well correlated to changes in concept so we used $\{-2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0\}$ as our bins in all subsequent experiments, with any value outside the range put in the first and last bin respectively. The choice of measure used to calculate the divergence between distributions can significantly affect the indicator and subsequently the concept drift detection ability of the algorithm. Sebastio and Gama (2007) provide a good comparison of such measures and we use Kullback-Leibler divergence which was found to be particularly effective.

The trigger in CDBD is the rule, or rules, which use the indicator to determine that concept drift has occurred and the classifier should be updated. In CDBD a variation of the *Western Electric rules* (Montgomery 2004) is used. The trigger fires when the indicator values for x out of the last y test batches have been above a threshold. For example, a 3/5 trigger fires when three out of the last five indicator values are above the threshold, and so on. To set the threshold value for the trigger we use an approach similar to that used by Lanquillon (1999). The distribution divergence between each of the first n test batches immediately after the reference batch and the reference batch itself are calculated. The threshold is set to one standard deviation above the mean of these n divergences.

When the occurrence of concept drift is flagged by the trigger the classifier needs to be updated using true class labels obtained for a set of recent test instances. The simplest approach to updating the classifier is to retrain it using the instances in the current test batch as the new training data. However, class imbalance can be very common in real life data streams such as in document filtering scenarios, so it is unreasonable to assume that the current test batch will give a balanced dataset from which to retrain the classifier. Instead, a balanced training set with d instances from each class, is constructed from as many test batches as are required. The batch where the detection takes place becomes the beginning of the new training window and new test batches are added to the training window as they arrive until the number of instances of each class is equal to, or greater than d. True class labels must be sought for every test instance in the training window. The reference histogram is then reconstructed from the batch following the end of the training window and the trigger threshold is recalculated using the n batches after the reference window.

4 Evaluation

To evaluate CDBD, its performance on eight document filtering problems, exhibiting a mixture of *natural* and *artificially induced* concept drift, was compared to existing approaches that use different amounts of labelled data. This section will describe the datasets used in these evaluations, and experiments to evaluate the

performance of the concept drift indicator used by CDBD and the CDBD approach in its entirety.

4.1 Datasets

Two datasets exhibiting natural concept drift are used in these evaluations. These are both based on a spam detection problem and were collected from real users' email accounts (Delany et al 2005). The classification task is to classify an email as *relevant* (*ham*) or *non-relevant* (*spam*) to a given user. These datasets are characterised by very high class imbalance and nothing is known about the amount or nature of the natural concept drift present. Emails were sorted chronologically and parsed to a bag-of-words representation with stop-word removal and Porter's stemming applied. Details of the spam datasets, Spam-1 and Spam-2, are given in Table 1.

The lack of knowledge about the amount and nature of the concept drift in real datasets makes formal evaluation of drift handling or detection approaches troublesome. While it is paramount that drift handling approaches are evaluated on real datasets exhibiting natural drift, it is also useful to evaluate approaches on datasets exhibiting controlled artificial drift. Artificial datasets tend to be one of two types:

- **Synthetic data**: in which the data is generated algorithmically in such a way as to ensure that drift occurs (for example the STAGGER approach (Schlimmer and Granger 1986) and the moving hyperplane approach (Kolter and Maloof 2003)).
- **Drift induced data**: in which real data is tweaked to introduce concept drift (for example, as in Huang and Dong (2007) and Klinkenberg (2004)).

In the evaluations presented in this article drift induced datasets are used, as we believe they retain some of the interesting artefacts which fully artificial data lacks (including underlying *natural drift* in addition to that induced artificially) and therefore lead to more informative evaluations. The drift induced datasets used in our evaluations are derived from three text corpora: the *Reuters*[1] collection, the *20 Newsgroups*[2] collection and a custom collection of news articles, *News Sources*, collected from national and international sources over a period of two months in 2011. The documents in the three corpora each belong to one of a set of predefined topics. Each classification task attempts to classify documents from one topic (the *target* topic) as *relevant* to a user, and classifies documents of all other topics as *non-relevant* to that user. To introduce concept drift the relevant topic is changed over time - an example of feature change. This approach to inducing concept drift into datasets is similar to that used by Lanquillon (1999).

The documents in each corpus were sorted chronologically and parsed to a bag-of-words representation with stop-word removal and Porter's stemming applied. One dataset was built from each corpus which was then divided into subsections or intervals as shown in Table 1. All documents with the target topic in each interval are labelled as relevant for that interval. All other documents are labelled

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

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

as non-relevant. As the target topic changes between intervals, documents that belong to a target topic will not appear in any other interval that has a different target topic to prevent introducing concept drift based on a conditional change. So, for example, for the News Sources datasets no documents from the *Business* topic will appear in interval C2 or C3.

We suspect that data streams with a balanced class distribution will be easier to handle so both balanced and skewed datasets are generated from the three text corpora. For the balanced datasets the labelled data stream was further processed to construct batches after which non-relevant documents were removed from the stream to ensure that each batch contained an equal number of relevant and non-relevant documents.

	Interval				
Dataset	**Interval**	**Target Topic**	**Size**	**#Rel.**	**#Non-rel.**
Reuters: 109,881 features	Training	earn	300	150	150
	C1	earn	4000	924	3076
	C2	acq	4000	642	3358
	C3	earn	4000	570	3430
	C4	acq	4000	536	3464
	C5	earn	1700	434	1266
20 Newsgroups: 66,581 features	Training	comp.*	300	150	150
	C1	comp.*	3900	1199	2701
	C2	rec.*	4000	1217	2783
	C3	comp.*	4000	1119	2881
News Sources: 123,502 features	Training	Business	300	150	150
	C1	Business	14900	3712	11188
	C2	Sports	15000	4867	10133
	C3	Entertainment	15000	2350	12650
Reuters Balanced: 46,360 features	Training	earn	300	150	150
	C1	earn	2000	1000	1000
	C2	acq	2000	1000	1000
	C3	earn	2200	1100	1100
20 Newsgroups Balanced: 63,244 features	Training	comp.*	300	150	150
	C1	comp.*	2000	1000	1000
	C2	rec.*	3800	1900	1900
	C3	comp.*	2900	1450	1450
News Sources Balanced: 89,733 features	Training	Business	300	150	150
	C1	Business	7900	3950	3950
	C2	Sports	8000	4000	4000
	C3	Entertainment	5700	2850	2850
Spam-1: 137,817 features	Training	-	300	150	150
	Test Set		8200	688	7512
Spam-2: 206,852 features	Training	-	300	150	150
	Test Set		9900	1036	8864

Table 1 Details of the datasets used in the evaluation.

The data in the intervals after the training interval was considered in batches of 100 documents. The reference histogram was constructed from the first batch after the training window and the threshold was set using the next 5 batches.

4.2 Signal Experiment

The aim of the first experiment was to test whether a signal derived from the distribution of classifier output, coupled with a trigger based on the Western electric rules could be used to detect concept drift. This evaluation was performed using the Reuters, 20 Newsgroups and News Sources datasets. The signal was evaluated on the imbalanced versions of the datasets as these are the datasets on which the signal was expected to perform the poorest. The signal experiment was run on only the first two intervals, C1 and C2, described in Table 1. Intuitively the signal is expected to be low and stable during interval C1 as the target topic is the same as that used in the training data. The signal should be significantly higher during interval C2 as the target topic is different.

Figures 2, 3 and 4 show the indicator, mean and standard deviations (calculated as described in Section 3) over the two intervals for each dataset. The concept drift point is marked by the dashed vertical line. Figures 2, 3 and 4 illustrate that although the signal is not perfect, in general the indicator values before the change in concept are substantially different from the values after the change in concept. This was confirmed using unpaired two-tailed t-tests which showed a statistically significant difference (at the 95% confidence level) in indicator values before and after the concept drift on all three datasets.

On the News Sourcesdataset it is evident from Figure 4 that the signal starts increasing before the change in concept. We suggest that this is due to naturally occurring concept drift in the data, as might be expected in real data (particularly in the News Sourcesdataset as it is a lot larger than the other datasets used) making a gradual change in concept more likely to occur. Figure 10, which will be discussed in Section 4.3, supports this interpretation of the results as there is a significant drop in classification accuracy in the same part of the graph (around instance $7,500$). Performing the same experiment but with the instance order randomized within each concept showed no major increase in the signal before the change in concept, which seems to support our argument that the signal increase seen in Figure 4 is due to natural concept drift. It also strengthens the argument that ordering the data chronologically can preserve interesting characteristics of the data which might otherwise be lost.

Figures 2, 3 and 4 also show that most signal values are below the mean plus one standard deviation before the change in concept, and above the mean plus one standard deviation after the change in concept. A detection threshold of the mean plus two or three times the standard deviation seems unable to separate the before and after signal so a detection threshold of the *mean plus one standard deviation* was used for all the experiments. The signal graphs also show the importance of using the Western Electric rules. Even though the signal is reasonably well behaved, it does often break the mean plus one standard deviation threshold before the concept changes. This would make a 1/1 trigger, although able to detect a change in concept the fastest, susceptible to false positives. A 3/5 trigger is a more cautious approach, slower to react to changes in concept. In a domain where the cost of labelling is high and a false positive trigger would result in unnecessary labelling, a 3/5 trigger would be better suited than a 1/1 trigger.

The signal experiment shows that the CDBD signal can be used to detect concept drift in a document stream. The signal experiment is, however, artificial as when the full CDBD approach is applied the classifier will be rebuilt and the

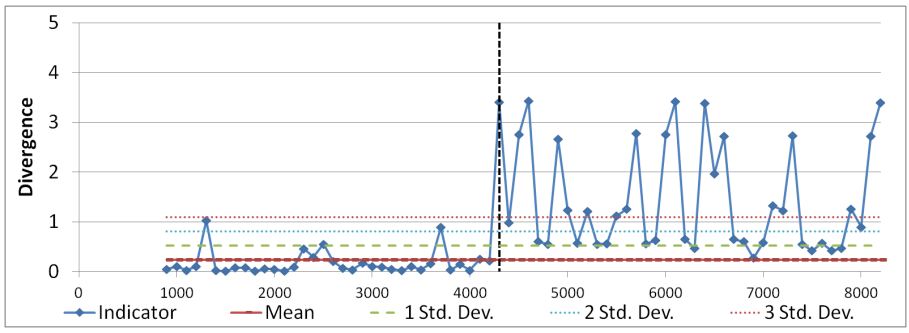

Fig. 2 Signal over time on the Reuters dataset

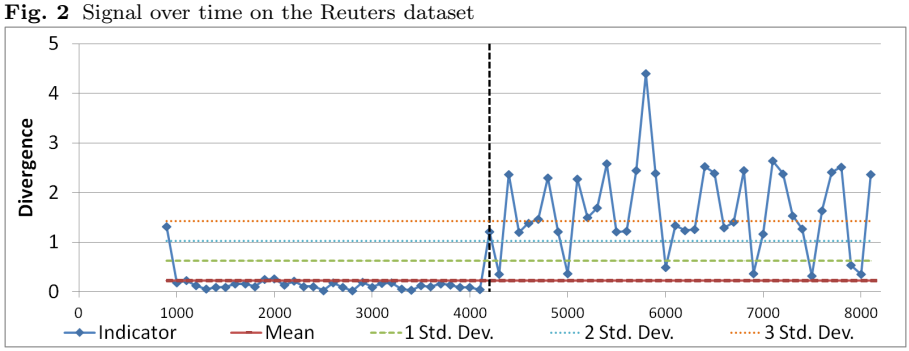

Fig. 3 Signal over time on the 20 Newsgroups dataset

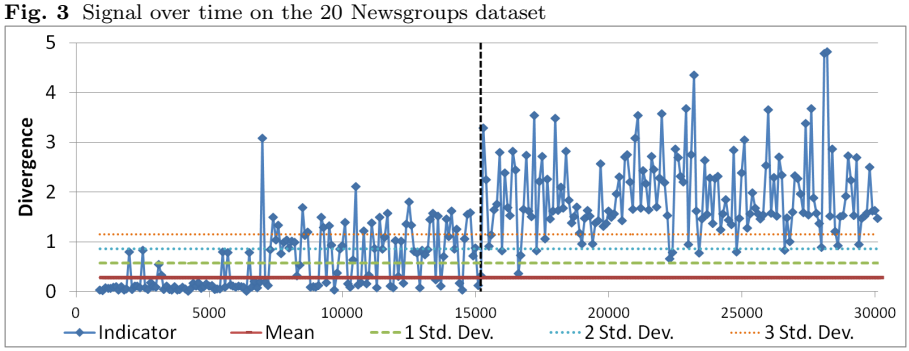

Fig. 4 Signal over time on the News Sourcesdataset

detection algorithm re-initialised when detection occurs. The aim of the next experiment was to evaluate whether the CDBD detection mechanism coupled with a rebuild policy could handle concept drift.

4.3 Detection and Rebuild Experiment

This evaluation compared the performance of the full CDBD approach to the performance of:

– **No Update**: No concept drift handling.

	Reuters		20 NG		NS		Spam-1	Spam-2
	Bal	Skew	Bal	Skew	Bal	Skew		
No update	77.98% – –	73.43% – –	70.32% – –	73.52% – –	59.67% – –	58.45% – –	72.11% – –	80.87% – –
CDBD 1/1	79.05% 900 14.52%	78.30% 6,054 34.20%	74.67% 900 10.34%	76.62% 1,573 13.22%	83.50% 2,700 12.50%	81.80% 8,018 17.86%	85.59% 3,575 44.14%	95.79% 4,499 45.44%
CDBD 2/3	83.85% 900 14.52%	77.96% 4,919 27.79%	79.79% 600 6.90%	71.68% 366 3.08%	80.81% 1,200 5.56%	83.75% 7,541 16.80%	87.79% 3,412 42.12%	93.16% 2,388 24.12%
CDBD 3/5	79.26% 600 9.68%	74.90% 3,211 18.14%	78.48% 600 6.90%	78.22% 984 8.27%	83.42% 900 4.17%	83.55% 4,332 9.65%	83.69% 3,653 45.10%	90.38% 2,048 20.69%
WRABD	83.68% 6,200 100%	80.10% 17,700 100%	79.57% 8,700 100%	72.81% 11,900 100%	82.75% 21,600 100%	76.70% 47,300 100%	81.84% 8,100 100%	92.62% 9,900 100%
Perfect Detector	83.68% 600 9.68%	80.59% 3,437 19.42%	74.59% 600 6.90%	78.50% 977 8.21%	84.31% 600 2.78%	84.19% 1,654 3.68%	– – –	– – –
Sliding Window	87.29% 6,200 100%	85.16% 17,700 100%	83.30% 8,700 100%	81.53% 11,900 100%	85.37% 21,600 100%	84.88% 47,300 100%	94.45% 8,100 100%	97.35% 9,900 100%

Table 2 The results of seven approaches on each of the eight evaluation datasets. For each approach the first row is the average class accuracy, the second row is the number of instances labelled and the third row is the percentage of instances labelled.

- **Sliding Window**: A fixed distribution, fixed-size sliding window approach. Training instances are added to the training window as they arrive and an equal number of training instances of the same class are discarded from the tail of the window.
- **WRABD**(Kuncheva 2009): The triggered rebuild WRABD approach which detects drift when the classification error in the current batch is above the mean plus three standard deviations. The mean and standard deviation is calculated using 10 batches of data, as in (Kuncheva 2009).
- **Perfect Detection** A notional approach which is set to trigger a classifier rebuild at each of the known concept drift points (this is used purely as a indicator of the performance limits of the CDBD approach)

Table 2 summarises the performance of the seven approaches compared on the balanced ("Bal") and skewed ("Skew") datasets derived from the three corpora and the two real datasets.

For each approach the first row is the average class accuracy over all the intervals, except the training data. The second row is the number of instances labelled and the third row is the percentage of instances labelled by that approach.

The first important thing to note from these results is that considering the difference in performance between the No Update and Sliding Window it is evident that all datasets exhibit significant concept drift.

It is also clear that the Sliding Window approach achieves the best performance on all datasets, although at the expense of 100% label usage. It is interesting to note that Sliding Window obtains higher accuracy than WRABD. We believe the main reason for this is because the sliding window recovers faster, in terms

of accuracy, after a change in concept occurs. We also believe that the sliding window has another advantage, that it handles gradual concept drift which might be present in the data, whereas WRABD only rebuilds when a significant change is suspected. Both the WRABD and CDBD approaches manage substantially better performance than the No Update approach on all datasets except one, where CDBD 2/3 and WRABD gets lower accuracy than No Update (Figure 9). The reason for this is that the detection threshold was too high. The signal graph from this dataset, Figure 3 seems to support this hypothesis. In this graph a larger proportion of signal values after the change in concept are below the mean plus one standard deviation than the other two signal graphs. When this experiment was rerun with the mean as the detection threshold both CDBD 2/3 and WRABD detected a change shortly after a change had taken place. A threshold of the mean plus one standard deviation works well on most of the datasets it was tested on, but is a dataset specific parameter which may need to be adjusted. A signal graph should help in the choosing of this parameter.

The overall result seems to be that out of the CDBD family of approaches the 2/3 trigger achieves the best balance of high average class accuracy and low label usage. Most importantly this CDBD approach achieves comparable average class accuracies to the triggered rebuild WRABD approach, while using only a fraction of the labelled data.

More detailed explorations of these results are shown in Figures 5 to 12. These figures plot the average class accuracy over time, using a five point moving average for smoothing, for the Sliding Window, No Update, CDBD 2/3 and WRABD approaches. The concept drift points are marked by dashed vertical lines and the concept drift detection points for the CDBD 2/3 trigger are marked with squares at the top and the detection points for WRABD are marked with a triangle.

Figure 5 is illustrative of the full suite of graphs, and clearly illustrates the concept drift process. This graph shows classifier performance on the Reuters Balanced dataset as the relevant topic is changed from *earn* (which was also relevant during training) to *acq*, and back to *earn* again. The flip-flopping of relevant topics is evident in the way that the No Update classifier performance drops dramatically after the first concept change, but improves just as dramatically after the second concept change as the *earn* topic on which it was trained becomes relevant again. The Sliding Window approach manages to maintain a reasonably constant classification accuracy throughout the concept changes, albeit with a slight dip during the time that the *acq* topic is relevant. The CDBD 2/3 profile is an almost perfect template for what is expected from a triggered rebuild strategy. After the first concept change the performance falls off dramatically. A concept drift detection is triggered almost immediately, however, and performance improves again. The visible delay between detection and performance improvement is partly due to the use of a moving average line and partly due to the time it takes to gather enough data to build a new training set. The same effect, although with a slightly longer delay between drift happening and being detected, is evident around the second change in concept. There is detection towards the end of Figure 5 which would not be expected in this region as no change in concept was induced here. However the accuracy increases after the detection so it could be argued that this detection was beneficial. Figure 6 tells an almost identical story to Figure 5. WRABD performs very well in both diagrams as it detects at both shift points.

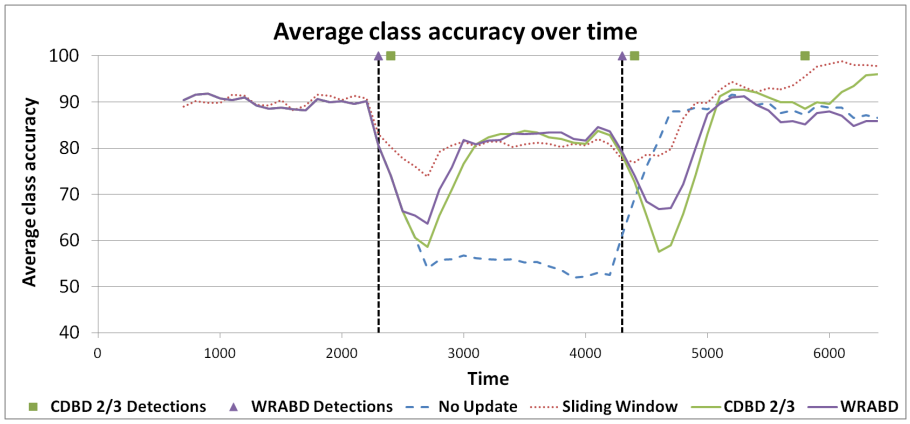

Fig. 5 Average class accuracy over time on the Reuters Balanced dataset

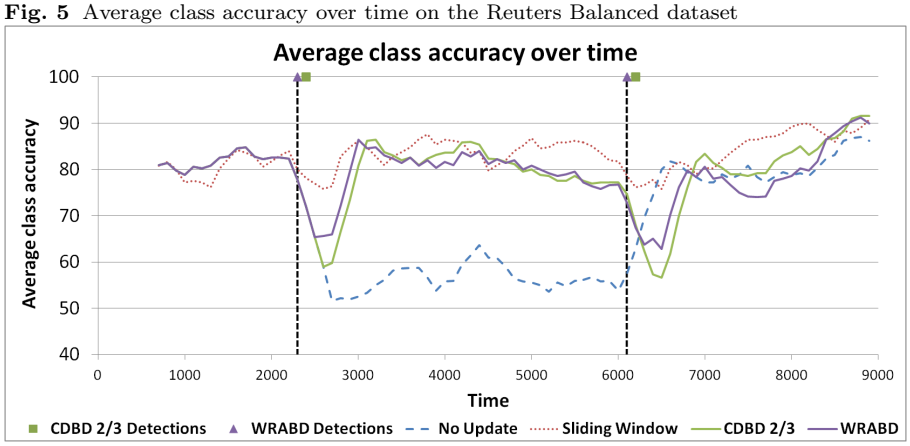

Fig. 6 Average class accuracy over time on the 20 Newsgroups Balanced dataset

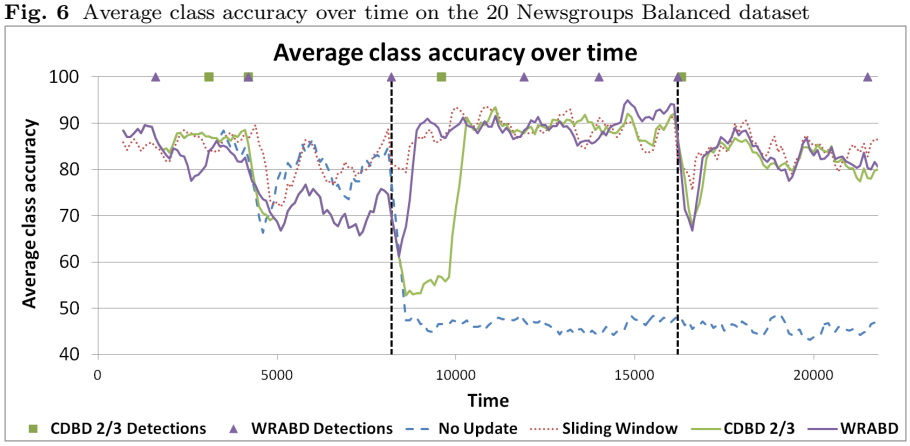

Fig. 7 Average class accuracy over time on the News SourcesBalanced dataset

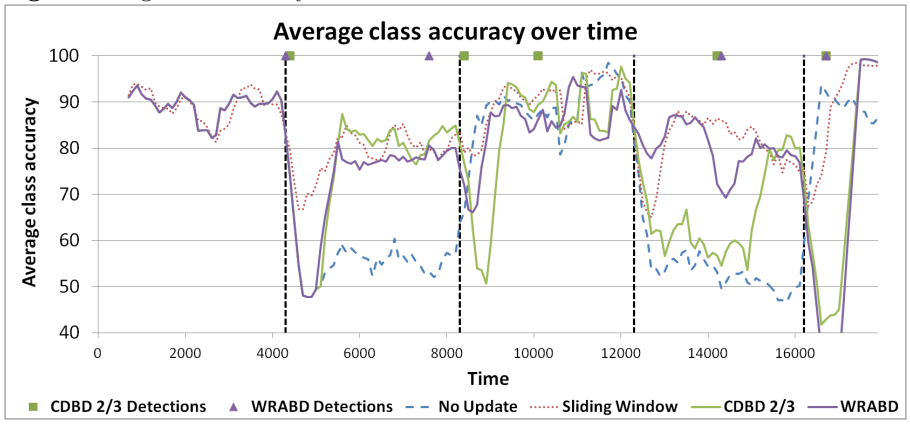

Fig. 8 Average class accuracy over time on the Reuters dataset

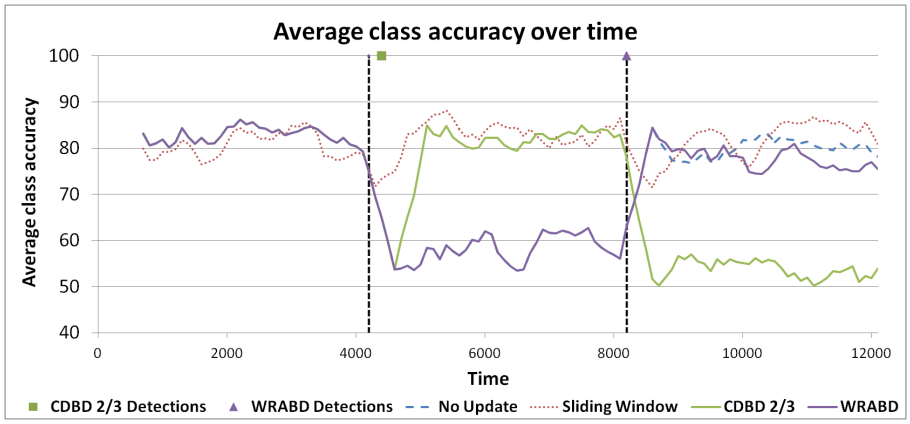

Fig. 9 Average class accuracy over time on the 20 Newsgroups dataset

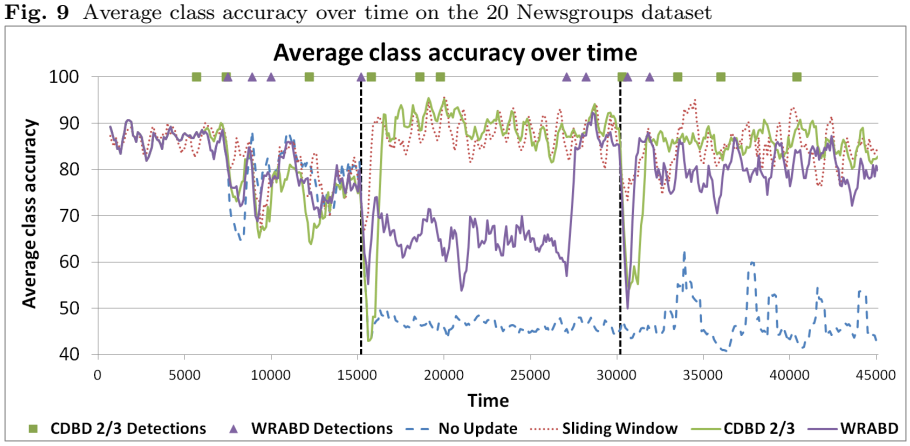

Fig. 10 Average class accuracy over time on the News Sourcesdataset

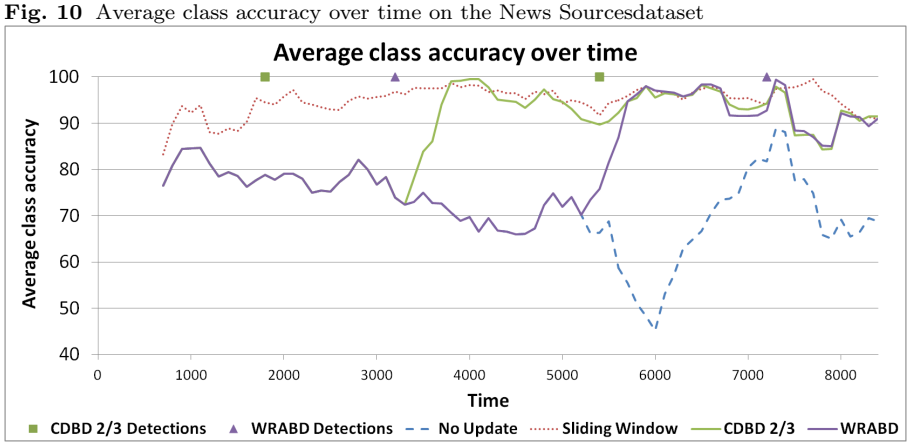

Fig. 11 Average class accuracy over time on the Spam-1 dataset

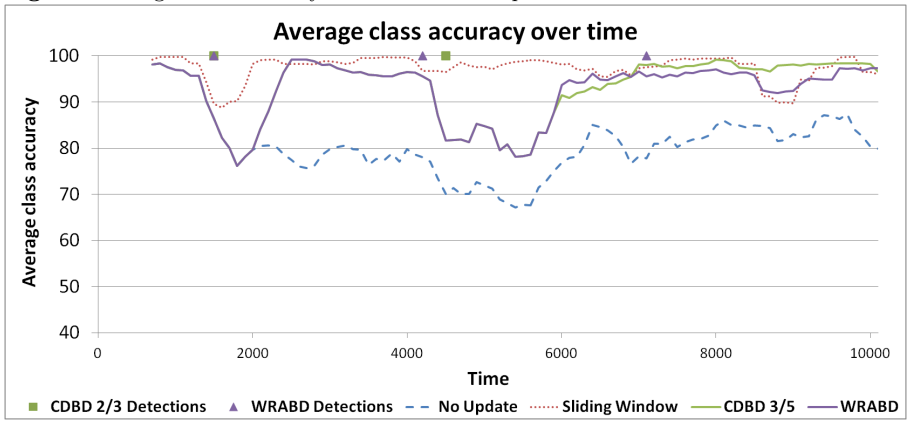

Fig. 12 Average class accuracy over time on the Spam-2 dataset

In Figure 7 both WRABD and CDBD 2/3 show a few false positive concept drift detections with drift detected in the News Sources Balanced data outside the artificially induced concept change. In this case, however, we believe that this may be due to natural concept drift within this data. At around the same period the performance of the Sliding Window and No Update approaches seem to also undergo a series of performance fluctuations which would suggest that the nature of the data is changing.

Figures 8 to 10 show that the performance of the seven approaches on unbalanced versions of the same three datasets is not quite as impressive. In two of the cases CDBD 2/3 makes false positive concept drift detections. In Figure 9 there is a missed detection, and in all cases once detections are made it takes a considerable amount of time before classification performance improves. This is because the class imbalance in the data means it takes a large number of test batches to build a new balanced training set with which to update the classifier. The amount of labelled data required in these cases is also much higher than for the balanced datasets for the same reason. In all cases, however, performance is still considerably better than the No Update approach and close to that of the Sliding Window approach which reinforces the conclusion that the CDBD approach can handle concept drift without using large numbers of labelled examples.

The classifier performances shown in Figures 11 and 12 for the real Spam-1 and Spam-2 datasets are a little harder to interpret as the nature of the concept drift present is unknown. The difference between the performance of the Sliding Window approach and No Update approach clearly show that concept drift is present. As this is real data this drift is likely to be due to a mixture of feature and conditional change, so it is interesting that the CDBD approach is able to detect drift and successfully respond to it. The response is far from perfect, however, and the rebuilding mechanism used by CDBD and WRABD suffers from the fact that it can take an excessively large number of test instance batches before an updated balanced training set can be created. During this time the out of date classifier is used to classify the stream which is suboptimal. We believe that an improvement to the rebuild process could decrease the amount of labelled data used and ensure that a classifier trained on up to date data is used earlier.

The performance of the Perfect Detection approach is included in Table 2 as an indication of the smallest amount of labelled data a triggered rebuild approach needs, while still handling concept drift effectively. Any differences between the amount of data used by the CDBD approaches and Perfect Detection show the impact of false positives, while the significantly larger amounts of labelled data required for the Reuters, 20 Newsgroups, Spam-1 and Spam-2 datasets show the impact of class imbalance.

Taken together the results of the evaluation experiments described above show that CDBD, a triggered rebuild concept drift detection approach that is based on classifier output and does not need full access to true class labels, can handle concept drift as effectively as WRABD, a triggered rebuild approach that requires full labelling. While CDBD does not perform as effectively as the Sliding Window approach, the potential that triggered rebuild approaches have for handling concept drift in scenarios constrained by high labelling cost is clearly evident.

5 Conclusion

Handling concept drift is important in building classifiers that will be effective in dynamic real-world environments. Many existing approaches to handling concept drift rely on full access to true class labels immediately after classification takes place, which is not realistic in scenarios in which the cost of labels is high. In this article we presented Confidence Distribution Batch Detection (CDBD) a triggered rebuild approach to handling concept drift that bases its concept drift trigger on an analysis of classifier output that does not need any access to actual class labels. Our evaluations showed that this approach performed more effectively than a similar triggered rebuild approach that requires full access to actual class labels, while using only a fraction of the labelled data in rebuilding the classifier.

Planned directions for extending this work include improvements to the rebuild policy, exploring further refinement of the CDBD concept drift trigger and performing a major evaluation of continuous and triggered rebuild approaches with the intention of finding the best aspects of both that could be combined into hybrid techniques.

References

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

Delany SJ, Cunningham P, Tsymbal A, Coyle L (2005) A case-based technique for tracking concept drift in spam filtering. Knowledge-Based Systems 18(4–5):187–195

Fan W, Huang Y, Wang H, Yu PS (2004) Active mining of data streams. Proceedings of the Fourth SIAM International Conference on Data Mining 35(4):457–461

Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: Bazzan A, Labidi S (eds) Advances in Artificial Intelligence SBIA 2004, Lecture Notes in Computer Science, vol 3171, Springer Berlin / Heidelberg, pp 66–112

Gao J, Fan W, Han J (2007) On appropriate assumptions to mine data streams: Analysis and practice. In: Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, pp 143–152

Hsiao W, Chang T (2008) An incremental cluster-based approach to spam filtering. Expert Systems with Applications 34(3):1599–1608

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

Kifer D, Ben-David S, Gehrke J (2004) Detecting change in data streams. In: Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, VLDB Endowment, vol 30, pp 180–191

Klinkenberg R (2004) Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis 8(3):281–300

Klinkenberg R, Renz I (1998) Adaptive information filtering: Learning in the presence of concept drifts. In: Workshop Notes of the ICML/AAAI-98 Workshop Learning for Text Categorization, AAAI Press, pp 33–40

Kolter J, Maloof M (2003) Dynamic weighted majority: a new ensemble method for tracking concept drift. In: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, IEEE, pp 123–130

Kubat M (1989) Floating approximation in time-varying knowledge bases. Pattern recognition letters 10(4):223–227

Kuncheva LI (2009) Using control charts for detecting concept change in streaming data. Tech. Rep. BCS-TR-001-2009, School of Computer Science, Bangor University, UK

Lanquillon C (1999) Information filtering in changing domains. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, pp 41–48

Lewis D (1995) Evaluating and optimizing autonomous text classification systems. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 246–254

Lindstrom P, Mac Namee B, Delany SJ (2010) Handling concept drift in a text data stream constrained by high labelling cost. In: Guesgen HW, Murray RC (eds) Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, AAAI Press

Lindstrom P, Mac Namee B, Delany SJ (2011) Drift detection using uncertainty distribution divergence. In: 2nd International Workshop on Handling Concept Drift in Adaptive Information Systems (HaCDAIS), IEEE Computer Society, pp 604–608

Masud M, Gao J, Khan L, Han J, Thuraisingham B (2008) A practical approach to classify evolving data streams: Training with limited amount of labeled data. In: Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on, pp 929 –934

Montgomery DC (2004) Introduction to Statistical Quality Control, 5th edn. Wiley

Nishida K, Yamauchi K (2007) Detecting concept drift using statistical testing. In: Corruble V, Takeda M, Suzuki E (eds) Discovery Science, Lecture Notes in Computer Science, vol 4755, Springer Berlin / Heidelberg, pp 264–269

Schlimmer JC, Granger RH (1986) Incremental learning from noisy data. Machine Learning 1:317–354

Sebastio R, Gama J (2007) Change detection in learning histograms from data streams. In: Neves J, Santos M, Machado J (eds) Progress in Artificial Intelligence, Lecture Notes in Computer Science, vol 4874, Springer Berlin / Heidelberg, pp 112–123

Spinosa EJ, de Leon AP, Gama J (2007) OLINDDA: a cluster-based approach for detecting novelty and concept drift in data streams. In: Proceedings of the 2007 ACM symposium on Applied computing, ACM, New York, NY, USA, SAC '07, pp 448–452

Swan R, Allan J (1999) Extracting significant time varying features from text. In: Proceedings of the eighth international conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '99, pp 38–45

Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag New York, Inc

Žliobaite I (2010) Change with delayed labeling: When is it detectable? In: Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, IEEE Computer Society, Washington, DC, USA, ICDMW '10, pp 843–850

Žliobaite I, Bifet A, Pfahringer B, Holmes G (2011) Active learning with evolving streaming data. In: Gunopulos D, Vazirgiannis M, Malerba D, Hofmann T (eds) Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part III, Springer-Verlag, Berlin, Heidelberg, ECML PKDD'11, pp 597–612

Woolam C, Masud M, Khan L (2009) Lacking labels in the stream: Classifying evolving stream data with few labels. In: Rauch J, Ras Z, Berka P, Elomaa T (eds) Foundations of Intelligent Systems, Lecture Notes in Computer Science, vol 5722, Springer Berlin / Heidelberg, pp 552–562

Yang Y, Liu X (1999) A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, SIGIR '99, pp 42–49

Zhu X, Zhang P, Lin X, Shi Y (2007) Active learning from data streams. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, ICDM '07, pp 757–762