2011-12-11

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., MacNamee, B. & Delany, S. (2011) Drift Detection Using Uncertainty Distribution Divergence. 2nd International Workshop on Handling Concept Drift in Adaptive Information Systems (HaCDAIS) , Vancouver, Canada,11-14, December. doi:10.1109/ICDMW.2011.70

Drift Detection using Uncertainty Distribution Divergence

Patrick Lindstrom, Brian Mac Namee, Sarah Jane Delany
School of Computing
Dublin Institute of Technology,
Dublin, Ireland
Email: first-name.second-name@dit.ie

***Abstract*— Concept drift is believed to be prevalent in most data gathered from naturally occurring processes and thus warrants research by the machine learning community. There are a myriad of approaches to concept drift handling which have been shown to handle concept drift with varying degrees of success. However, most approaches make the key assumption that the labelled data will be available at no labelling cost shortly after classification, an assumption which is often violated. The high labelling cost in many domains provides a strong motivation to reduce the number of labelled instances required to handle concept drift. Explicit detection approaches that do not require labelled instances to detect concept drift show great promise for achieving this. Our approach Confidence Distribution Batch Detection (CDBD) provides a signal correlated to changes in concept without using labelled data. We also show how this signal combined with a trigger and a rebuild policy can maintain classifier accuracy while using a limited amount of labelled data.**

***Keywords*-concept drift; explicit drift detection; labelling cost; classifier confidence;**

I. 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. However, data gathered from real life processes can vary over time. Examples of this include seasonal changes in climate or customer spending, or the occurrence of major events, such as elections and the introduction of new laws. Using a static model in such domains is inadequate as the data is exhibiting a phenomenon known as *concept drift*.

The different ways of handling concept drift evident in the literature can be categorised into two main approaches [1]. The first approach does not attempt to identify when drift is occurring but continuously and regularly updates the classifier assuming that this will allow the classifier to handle the drift whenever it occurs. This is the most common approach to handling drift and the simplest example of this is the sliding window technique that rebuilds the classifier with new data as it arrives and discards some old data according to a forgetting mechanism. This can be considered a '*continuous rebuild*' approach. The second approach explicitly detects when a change in concept is occurring and only then adjusts the classifier. This is achieved by monitoring the value of an indicator, such as the misclassification rate and rebuilding the classifier when the indicator changes significantly. This can be considered a '*triggered rebuild*' approach.

The majority of continuous and triggered rebuild approaches tend to require the true label of the instances to be available shortly after classification [1, 2, 3]. In many domains this is not a restriction, however in domains such as document filtering where labelling instances with their true class has a high cost, these approaches are not feasible. Consider a news analytics application that receives a continuous stream of news articles and attempts to determine relevance for users. As news and opinions change over time, keeping the classifier up to date requires new labelled documents as training data. There is considerable effort involved in reading and categorising texts to create the necessary labelled data. The high labelling cost provides a strong motivation to reduce the number of labelled instances in techniques for handling drift. While the number of labels required by continuous rebuild approaches can be reduced by using sampling (e.g. active learning [4, 5]) triggered rebuild approaches can also offer significant potential to reduce the number of labelled instances used in handling concept drift.

Our contribution, Confidence Distribution Batch Detection (CDBD), is an concept drift handling approach which explicitly detects changes in the data (as opposed to changes in the decision boundary) without using labelled data.

CDBD compares the distribution of classifier confidences in a batch to a reference distribution to generate an indicator stream, which we refer to as a *signal*. We also couple the CDBD signal with a trigger which flags a change in concept when the distribution divergence is above a threshold. CDBD only requires labelled data when concept drift has been flagged and we show that it gives comparable results to other drift handling approaches while using a smaller amount of labelled instances by evaluating it on two text classification scenarios.

The paper is organised as follows. Section II discusses existing research into concept drift detection. Section III describes our approach to concept drift detection while Section IV discusses how the approach was evaluated and the results of the evaluation followed by conclusions.

II. Background

In an ideal classification scenario the classifier can be trained on data which is representative of the concept allowing it to make accurate predictions on unseen instances. However, if there is significant concept drift present in the data a drift handling technique is needed.

The simplest continuous rebuild approach is to periodically retrain the classifier using a subset of previous instances in the stream as the new training data, an approach known as the *sliding window*. Sliding window approaches can have either a fixed training window size (e.g. [2]) or adaptive size which uses an indicator to adjust the amount of data used. The goal of the indicator is to grow the window size while the concept is stable and collapse the window when a change in concept is suspected. Klinkenberg and Renz [6] developed a window resizing heuristic based on three indicators, error rate, precision, and recall, there are however many other viable error based heuristics (e.g. [3]).

Most triggered rebuild approaches to handling 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 is above a threshold. Klinkenberg and Renz [6] introduced three sources of concept change indicators; properties of the classifier, properties of the data, and properties of the classification output.

The first source is indicators derived from the internal workings of the classifier, with decision trees being particularly common. Decision tree characteristics such as leaf changing statistics [7] and expected loss [7, 8] have been found to be well correlated with changes in concept without the need for the document stream to be labelled.

The second source is indicators derived from the data. This type of indicator tend to be domain specific. Examples of drift detection from textual data streams include monitoring word frequencies [9] and the formation of new word clusters [10, 11]. Kifer et al. [12] introduces a less data dependant approach which uses a two window paradigm. The distribution of a single instance feature inside a reference batch is compared to the distribution inside the current batch 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. Sebastião and Gama [13] take a similar approach but use Kullback-Leibler divergence to measure the difference. Both approaches require the identification of a feature distribution which is sensitive to change in concept. In a dataset such as a text dataset where each document is represented by word frequencies, monitoring the distribution of one word is unlikely to yield satisfactory detection.

The final source is indicators derived from the output of a classifier. The advantages of using classifier output is that it is classifier and data independent, and does not presuppose knowledge about feature distributions. Kuncheva [1] enhances the sliding window heuristic in [6] 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. This requires a labelled data stream but there are approaches that do not require the data to be labelled. One of the earliest works in autonomous text classification systems uses an effectiveness measure which evaluates the decisions made by the system [14]. The actual effectiveness requires the true labels, so an estimated effectiveness is used which is calculated as a function of the classifier prediction and probability of class membership. Lanquillon [15] estimates a class confidence range from the training data. A change in concept is flagged if the number of predictions in the range exceeds the average calculated over a number of previous batches by more than three standard deviations. Žliobaite [16] also uses a window paradigm to compare the posterior probabilities in a reference window to the probability of class memberships in the current window using Kolmogorov-Smirnov, ranksum Wilcoxon and a two sample t-test.

Our approach uses the two window paradigm from [12, 13, 16] coupled with the histogram divergence from [13] but applied to an indicator, the classifier confidence, which is classifier and data independent.

III. Approach

A high level overview of CDBD is as follows. The classifier, built from the initial training data, classifies instances in the stream, storing the output of the classifier in batches. The detection algorithm calculates the indicator value for the current batch and flags a change in concept if the value causes a trigger to fire.

CDBD can be used on any classifier which produces a score which can be considered an estimate of classifier confidence that a prediction is correct. The indicator used is a measure of the divergence between the distribution of the classifier scores in $batch_i$ and the distribution in a reference batch $batch_{ref}$, the reference batch being the first batch of instances classified after training. Divergence between distributions is estimated by comparing histograms constructed from the classifier scores. High divergence can be indicative of a change in concept. The choice of divergence measure effects the indicator value and subsequently the detection ability of the algorithm. Sebastião and Gama [13] provide a good comparison of such measures and Kullback-Leibler divergence was found to be particularly effective.

The trigger is the rule or rules which use the indicator to determine if a rebuild should take place or not. A variation of the 'Western Electric rules' [17] is used as trigger. The trigger fires when x out of the last y indicator values are above a threshold. 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 we use an approach similar to the one used by Lanquillon [15], the histogram dissimilarity of n batches after the reference window is calculated and the threshold is set to one standard deviation above the mean of the n dissimilarities.

If a change of concept is flagged the classifier is updated and the reference window refreshed.

IV. Evaluation

CDBD was evaluated on a document filtering problem, a domain with high labelling costs. The datasets used simulate a news filtering problem where a system tries to distinguish if a document is relevant or not to a user[1].

The evaluation used two datasets derived from two text corpora, the *Reuters*[2] and *20 Newsgroups*[3] collections. The documents in each corpus were sorted chronologically to simulate a data stream. The documents are parsed to a bag-of-words representation with stop-word removal and Porter's stemming applied.

All documents in both collections belong to one of a set of predefined topics. A subset of these topics are labelled as *relevant* to a reader at a particular time. Drift is induced in this relevance concept by changing the topics which are considered relevant over time. One dataset was built from each corpus which was then divided into intervals as shown in Tables I and II. All documents in the target topic in each interval are labelled as relevant for that interval and all documents belonging to the non-target topic are removed from that interval. Documents belonging to neither the target or non-target topic are labelled non-relevant[4]. As the target topic changes across intervals the effect is a data stream where the type of document considered relevant changes over time.

This approach to generating concept drift datasets is similar to that used by Lanquillon [15].

The classifier was trained initially on the training data interval data. In our evaluation we used a Support Vector Machine (SVM) [18] which produces a score which is a function of the distance between an instance and the hyperplane.

The data in the intervals after the training interval was considered in batches of 100 documents. A reference histogram is constructed from the first batch using the bins $\{-2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0\}$, any value below -2 or above 2 were put in the first and last bin respectively. The threshold was set using the next five batches. The indicator for each subsequent batch was calculated using Kullback-Leibler divergence and 1/1, 2/3 and 3/5 triggers were used.

Interval	**Target Topic**	**Interval Size**	**#Rel.**	**#Non-rel.**
Training Data	earn	300	150	150
C1	earn	4000	948	3052
C2	acq	4000	683	3317
C3	earn	4000	590	3410
C4	acq	3900	561	3339
C5	earn	1800	436	1364
		18000	3368	14632

Table I
Class Distribution of Reuters Dataset

Interval	**Target Topic**	**Interval Size**	**#Rel.**	**#Non-rel.**
Training Data	comp.*	300	150	150
C1	comp.*	4000	1261	2739
C2	rec.*	4000	1246	2754
C3	comp.*	4000	1143	2857
C4	rec.*	1900	90	1810
		14200	3890	10310

Table II
Class Distribution of 20 Newsgroups Dataset

The simplest approach to rebuilding is to use the current batch as the new training data. However this does not work on these datasets due to the significant class imbalance. Instead the batch where the detection takes place becomes the beginning of the new training window. New batches get added to the training window as they arrive until the number of instances of each class is equal to, or greater than the number in the original training data, in this example 150 instances of each class. From this window the most recent 150 instances from each class are used to form the new training data used to retrain the classifier. The reference histogram is reconstructed from the following batch and once the threshold is recalculated from the following five batches detection can restart.

1) Signal Experiment: The first experiment aimed to evaluate if a signal derived from the distribution of classifier output coupled with a trigger can detect concept drift. This is evaluated on a subset of both datasets, namely the intervals Training Data, C1 and C2. The expectation is that drift should not be detected in C1 but should be detected on each batch in C2. The metrics used are True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) rates. A detection in C1 is a FP, while a detection in C2 is a TP. Conversely, a non-detection in C1 is a TN, while a non-detection in C2 is a FN. These figures can be further refined into Accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$), Precision ($\frac{TP}{TP+FP}$) and Recall ($\frac{TP}{TP+FN}$) numbers. The final metric used is the run length (RL) which is defined as the number of batches between the batch where the concept shift occurred and where the detection algorithm flags a change in concept.

Figure 1 shows the indicator, mean and standard deviation over two concepts. The concept shift point is marked by the dashed vertical line.

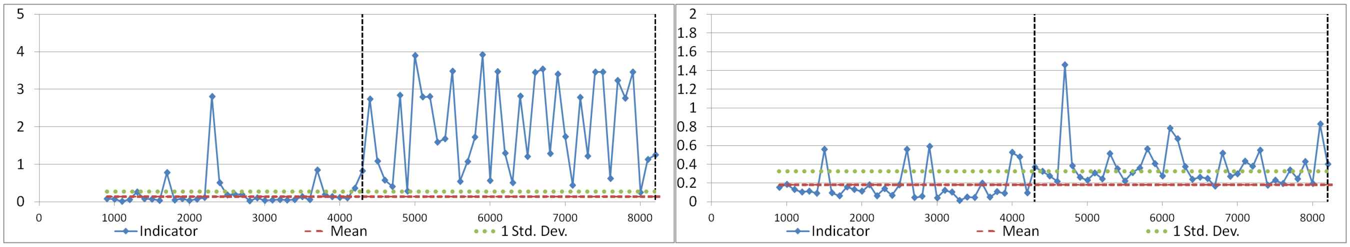
Figure 1 seems to indicate that the signal is not perfect, but

[1] Available at: http://www.comp.dit.ie/aigroup/plindstrom

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

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

[4] 19 instances were removed from the Reuters corpus before the drift induction process began as they are categorised as both *earn* and *acq*.



(a) Signal over time on the Reuters dataset

(b) Signal over time on the 20 Newsgroups dataset

Figure 1. The signal over time on two concepts without rebuilding.

in general the indicator values before the change in concept are significantly different from the values after the change in concept. This was confirmed using an unpaired t-test which gives a two-tailed P value of less than 0.0001 on both the Reuters and 20 Newsgroups dataset.

These graphs show that the signal is relatively stable before the concept change, the few spikes that exist may be caused by artefacts of the data, but after the change the signal is consistently above the mean on both datasets. On the Reuters dataset the signal is consistently above the threshold of the mean plus one standard deviation however on the 20 Newsgroups dataset the signal is above the mean, but not above the mean plus one standard deviation.

The detection results are shown in Tables III and IV.

	#Detections									
Ind	**C1**	**C2**	**#FP**	**#TP**	**#FN**	**#TN**	**Acc**	**Prec**	**Rec**	**RL**
1/1	5	39	5	39	1	35	0.93	0.89	0.98	0
2/3	2	40	2	40	0	38	0.98	0.95	1	0
3/5	0	39	0	39	1	40	0.99	1	0.98	1

Table III
DETECTION SUMMARY TABLE ON THE REUTERS DATASET

	#Detections									
Ind	**C1**	**C2**	**#FP**	**#TP**	**#FN**	**#TN**	**Acc**	**Prec**	**Rec**	**RL**
1/1	5	20	5	20	20	35	0.69	0.80	0.50	0
2/3	2	20	2	20	20	38	0.73	0.91	0.50	0
3/5	0	20	0	20	20	40	0.75	1	0.50	0

Table IV
DETECTION SUMMARY TABLE ON THE 20 NEWSGROUPS DATASET

	#Detections									
Ind	**C1**	**C2**	**#FP**	**#TP**	**#FN**	**#TN**	**Acc**	**Prec**	**Rec**	**RL**
1/1	8	38	8	38	2	32	0.88	0.83	0.95	0
2/3	2	40	2	40	0	38	0.98	0.95	1	0
3/5	0	40	0	40	0	40	1	1	1	0

Table V
DETECTION SUMMARY TABLE ON THE 20 NEWSGROUPS DATASET WITH LOWERED THRESHOLD

The trigger behaves as expected on the Reuters dataset. The 1/1 trigger gives more FPs and a perfect run length and the 3/5 trigger gives the least FPs but is slower to react giving less TPs and FNs and the highest RL. However the results are not as clear on the 20 Newsgroups dataset which has high precision rates, but is plagued by low recall rates. Lowering the detection threshold from the mean plus one standard deviation to above the mean on the 20 Newsgroups dataset produces the significantly better detection table V. This would suggest that the threshold is an important dataset specific parameter, which is a drawback of our approach that we intend to address in future work.

2) Detection and Rebuild Experiment: The signal experiment suggests that the signal can be used to detect concept changes in the document stream. In a real world application the classifier gets rebuilt and the detection algorithm reinitialised when the detection occurs. The second experiment therefore aimed to evaluate whether the CDBD detection mechanism coupled with a rebuild policy can handle concept drift. The experiment used a simple forward rebuild mechanism and was run on the full Reuters (five intervals) and 20 Newsgroups (four intervals) datasets.

This evaluation compared CDBD to no drift handling and to a fixed distribution, fixed window size sliding window. CDBD was also compared to the explicit detection techniques WRABD which detects drift when the classification error is above the mean plus three standard deviations, and the Perfect Trigger approach which is set to detect drift at each concept shift and retrained using forward rebuild.

Due to the imbalance in the datasets the performance measure used was average class accuracy. The number of labels used is also reported. The number of labelled instances required should be minimized because of the cost of labelling in this domain.

The results are presented in Tables VI and VII. Not rebuilding the classifier gives a surprisingly high average class accuracy, however the two class accuracies (Rel. and Non.) show that this is due to the class imbalance.

CDBD follows a similar pattern to the signal experiments, using the 1/1 trigger results in the most FPs and using the 3/5 trigger gives the least. The number of FPs is not directly correlated to the number of labels required due to a combination of the datasets and rebuild policy used. Certain parts of the datasets feature a larger class imbalance than others. If a detection takes place in a particularly imbalanced region of the dataset more labelled data is required as the training window keeps growing until the original, balanced class distribution is reached. The perfect detection results show that even if the concept change points are known 20% is needed to maintain classifier accuracy when this pairing of datasets and rebuild policy is used.

The results seem to confirm the intuition that rebuilding often gives higher accuracy, which is why the sliding window obtains the highest average class accuracy on both datasets. It also explains why perfect detection does not beat CDBD 1/1 on the 20 Newsgroups dataset as 1/1 rebuilds more times (due to the two FPs). However rebuilding often incurs significant labelling cost witch is not desirable.

Overall the results show that CDBD compares favourably to both the WRABD based benchmark and the sliding window while using a lower number of labelled instances.

	Accuracies				
Experiment	**Rel.**	**Non.**	**Avg.**	**#FP**	**% Labels used**
No Rebuild	51.35	90.28	70.82	-	0
CDBD 1/1	67.05	82.45	74.75	3	25.88
CDBD 2/3	71.47	79.84	75.66	2	26.12
CDBD 3/5	64.54	80.75	72.64	1	27.56
WRABD	72.17	78.59	75.38	2	100
Perfect Detection	72.12	84.45	78.28	-	19.23
Sliding Window	81.60	83.54	82.57	-	100

Table VI
RESULTS OF THE REUTERS DATASET EXPERIMENT

	Accuracies				
Experiment	**Rel.**	**Non.**	**Avg.**	**#FP**	**% Labels used**
No Rebuild	59.22	69.21	64.22	-	0
CDBD 1/1	73.84	72.53	73.18	2	23.58
CDBD 2/3	71.05	66.26	68.66	1	14.09
CDBD 3/5	67.44	76.05	71.74	0	16.13
WRABD	58.29	71.44	64.86	0	100
Perfect Detection	72.33	69.88	71.11	-	20.24
Sliding Window	78.08	75.50	76.79	-	100

Table VII
RESULTS OF 20 NEWSGROUPS DATASET EXPERIMENT

V. CONCLUSION

The distribution of classifier confidences can be used to create a drift detection signal. The signal was coupled with a trigger and rebuild mechanism which showed some potential at reducing the need for labelled instances. Future work might include evaluating CDBD on more datasets, the optimization of parameters, and exploration of the robustness of the algorithm to different parameter values. Improvements to the rebuild policy would also be an interesting direction for future work. Another research direction of interest is to use the signal coupled with continuous rebuild, such as adjusting classifier parameters, ensemble fusion rules or as a sampling parameter in active learning.

REFERENCES

[1] L. I. Kuncheva, "Using control charts for detecting concept change in streaming data," School of Computer Science, Bangor University, UK, Tech. Rep., 2009.

[2] M. Kubat, "Floating approximation in time-varying knowledge bases," *Pattern recognition letters*, vol. 10, pp. 223–227, 1989.

[3] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues, "Learning with drift detection." in *SBIA'04*, 2004, pp. 286–295.

[4] P. Lindstrom, S. Delany, and B. Mac Namee, "Handling concept drift in a text data stream constrained by high labelling cost," in *in FLAIRS10*, 2010, p. 3237.

[5] I. Žliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with evolving streaming data," in *Machine Learning and Knowledge Discovery in Databases*, 2011, vol. 6913, pp. 597–612.

[6] R. Klinkenberg and I. Renz, "Adaptive information filtering: Learning in the presence of concept drifts," *Learning for Text Categorization*, p. 3340, 1998.

[7] W. Fan, Y. Huang, H. Wang, and P. S. Yu, "Active mining of data streams," in *In Proc. SIAM Int. Conf. on Data Mining*. Society for Industrial Mathematics, 2004, p. 457.

[8] S. Huang and Y. Dong, "An active learning system for mining time-changing data streams." *Intelligent Data Analysis*, vol. 11, no. 4, pp. 401 – 419, 2007.

[9] R. Swan and J. Allan, "Extracting significant time varying features from text," in *In Proc. Int. Conf. on Information and knowledge management*. ACM, 1999, p. 45.

[10] W. Hsiao and T. Chang, "An incremental cluster-based approach to spam filtering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1599–1608, Apr. 2008.

[11] E. J. Spinosa and F. de Leon, "Olindda: A cluster-based approach for detecting novelty and concept drift in data streams," in *In Proc. ACM symposium on Applied computing*. ACM, 2007, p. 452.

[12] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *In Proc. Int. Conf. on Very large data bases*, vol. 30, 2004, p. 191.

[13] R. Sebastião and J. Gama, "Change detection in learning histograms from data streams," in *Progress in Artificial Intelligence*, 2007, vol. 4874, pp. 112–123.

[14] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," in *In. Proc. ACM SIGIR Conf. on Research and development in information retrieval*, 1995, pp. 246–254.

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

[16] I. Žliobaite, "Change with delayed labeling: When is it detectable?" in *2010 IEEE International Conference on Data Mining Workshops*, 2010, p. 843–850.

[17] D. C. Montgomery, *Introduction to Statistical Quality Control*, 5th ed. Wiley, Aug. 2004.

[18] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc, 1995.