Procedia Computer Science

Volume 53, 2015, Pages 103–112

2015 INNS Conference on Big Data

Don't pay for validation : Detecting drifts from unlabeled data using margin density

Tegjyot Singh Sethi[1] and Mehmed Kantardzic[1]

Data Mining Lab, University of Louisville, KY, USA
tegjyotsingh.sethi@louisville.edu, mehmed.kantardzic@louisville.edu

Abstract

Validating online stream classifiers has traditionally assumed the availability of labeled samples, which can be monitored over time, to detect concept drift. However, labeling in streaming domains is expensive, time consuming and in certain applications, such as land mine detection, not a possibility at all. In this paper, the Margin Density Drift Detection (MD3) approach is proposed, which can signal change using unlabeled samples and requires labeling only for retraining, in the event of a drift. The MD3 approach when evaluated on 5 synthetic and 5 real world drifting data streams, produced statistically equivalent classification accuracy to that of a fully labeled accuracy tracking drift detector, and required only a third of the samples to be labeled, on average.

Keywords: Concept drift, detection, limited labeling, SVM, margin density, online learning

1 Introduction

The Velocity and non-stationarity of Big data has warranted the need for online stream data mining systems which are capable of working in an autonomous or atleast a semi-autonomous manner [11]. An essential characteristic of these systems is the ability to detect change, referred to as Concept Drift [18], in the underlying data distribution and adapt to them effectively. Traditional change detection schemes proposed in literature [9], keep track of discrepancies in the predicted and actual labels of the incoming samples, to explicitly signal drift when the classification performance drops. This implicit assumption of availability of the actual labels is not justified in the case of streaming data, as labeling is expensive, time consuming and may not be available in most cases. Thus the ability to work with unlabeled data has gained considerable interest from the streaming data research community [19, 20]. This paper explore the ability to detect concept drift using unlabeled data and proposes the Margin Density Drift Detection (MD3) algorithm, which tracks changes in the classifier boundary by monitoring distribution of incoming samples relative to it.

This paper makes the following contributions to the state of the art: Analysis of a classifier's margin density as a signal for drift detection; Development of the MD3 algorithm capable of

Selection and peer-review under responsibility of the Scientific Programme Committee of INNS-BigData2015

doi:10.1016/j.procs.2015.07.284

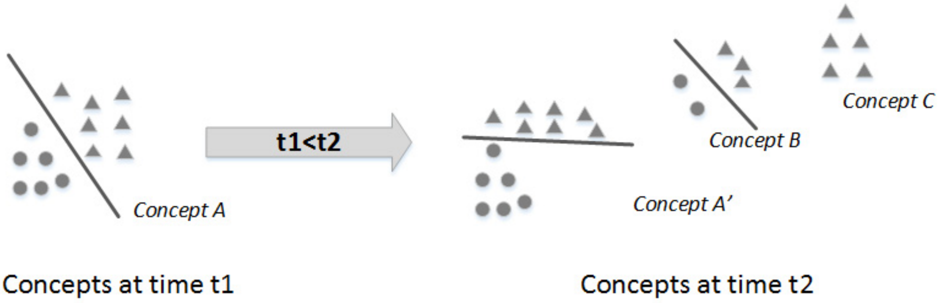


Figure 1: Illustration of drifts in data distribution, $P(x)$- as introduction of Concept B,C at t2; and drift in class boundary, $P(y|x)$- as changes in Concept A, resulting in a change in $P(x,y)$

using this signal; Experimental and statistical analysis of the proposed methodology on 5 real world and 5 synthetic datasets to understand it's efficacy and limitations.

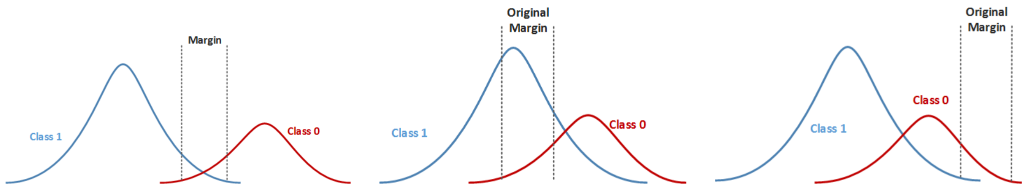
The rest of the paper is organized as follows: Section 2 presents the background research work and motivation behind the proposed MD3 approach. Section 3 describes the generic MD3 methodology and also presents a specific implementation using Support Vector Machines. Section 4 presents experimental results as proof of idea and as a comparative study over 10 datasets. Conclusion and avenues for extension are presented in Section 5.

2 Background and Motivation

Drift detection has it's roots in sequential statistical tests such as the CUSUM and the Page-Hinkley test, which explicitly measure deviation in the mean value of the classification performance (commonly accuracy), and signal change based on a threshold [8]. This idea of change was further reformed in the well known DDM and the EDDM methodologies which, based on Statistical Learning Theory, look for upward trends in the classification error as a signal for change [3]. Several other drift detectors have been proposed as a variation of these base approaches and a detailed analysis of 8 frequently used drift detection schemes is presented in [9]. All of these approaches use labeled data samples to explicitly capture change in the classifier boundary. As such, these approaches are powerful in detecting drifts in the classification boundary, but not suitable for applications where the labeling is scarce and expensive.

To account for the limitations of labeled data, unsupervised techniques, which monitor changes in the distribution of the attribute values or the classification output, have been proposed. The former consists of incremental clustering and novel class detection schemes which keep track of distribution of samples in the feature space [15]. Another such method [13], measures changes to the correlation between the attribute values of two windows of variable sizes. In case of the methodologies based on the classification output, the posterior estimates of the current classifier in the system are monitored to detect changes [19]. The earliest of these methods was proposed in [12], wherein a prototype based classifier was used to represent each class and an increase in the number of samples that fall outside the confidence range of all classes is taken as an indication of drift. The more recent Confidence Distribution Batch Detection (CDBD) approach [14], uses the KL-divergence measure to compute changes in the distribution of a classifier output confidence distribution against an initial reference distribution.

The unlabeled drift detection approaches are attractive for their inexpensive detection, but



(a) Initial distribution of samples as distance from the model boundary

(b) Drift causing increase in margin density

(c) Sudden drift causing drop in margin density

Figure 2: Drifting scenarios and their effects on the margin density

they are sensitive to changes that do not affect the classification boundary directly. An illustration of different types of drifts is presented in Fig. 1. Occurrence of Concept C's cluster does not have any direct effect on the margin boundary of Concept A, while Concept B is an indication of a new local concept with its own boundary, occurring over time. In an effort to detect drifts in the joint probability distribution of samples x and their labels y, given by $p(x, y) = p(x).p(y|x)$, the unlabeled clustering approaches tend to monitor changes in $p(x)$, shown as introduction of Concept B and C at time t2 in Fig. 1; while the changes to the class boundary $p(y|x)$ requires explicit labeled samples, shown as changes to the class boundary leading to Concept A'. Although changes to $p(x)$ can be indicative of drift, they are susceptible to false alarms, as they give equal importance to attributes whose change do not directly affect classification performance [6]. In contrast, the proposed MD3 approach tracks changes to the number of samples in a classifier's margin, to act as a pseudo to signal explicit drift in $p(y|x)$, but without using any labeled data. As such, it aims at providing robustness to false alarms - as done by the labeled drift detectors, and also leaves it open for integration with the other clustering techniques- which are good at recognizing local subspace concepts, to have a comprehensive drift detection on $p(x, y)$, in high dimensional spaces.

3 Proposed MD3 methodology

The proposed Margin Density Drift Detection methodology signals concept drift by tracking changes in the expected density distribution of the unlabeled samples with respect to a classification boundary. A classifier's margin is defined as the region of space close to the model boundary, where the predictions are highly uncertain. The intuition behind the MD3 approach is that a change in the $p(y|x)$ will result in an increase or decrease in the number of samples within the margin as shown in Fig. 2, here the plots represents the probability distribution of the samples based on their distance from the model boundary. An increase in the number of samples within the margin is caused by class distributions moving closer to the boundary, as shown in Fig. 2 b), as a result of a gradual drift, and a decrease (Fig. 2 c)) is observed in case of a sudden drift when the entire class distribution moves to a different part of the feature space. Both signals warrant further investigation and as such lead to drift detection.

The overall architecture of a drift handling classification system is shown in Fig. 3. The stream is processed as a sliding window of size S samples moving at a rate of S_r samples. This combines the advantages of incremental classifiers, which have low latency of drift detection, and chunk based approaches, which have a computational advantage of being able to process several samples at once. The concept of the margin density (ρ) for sliding window defined below, is employed as a signal for drift detection. As seen in Fig. 3, the margin density is

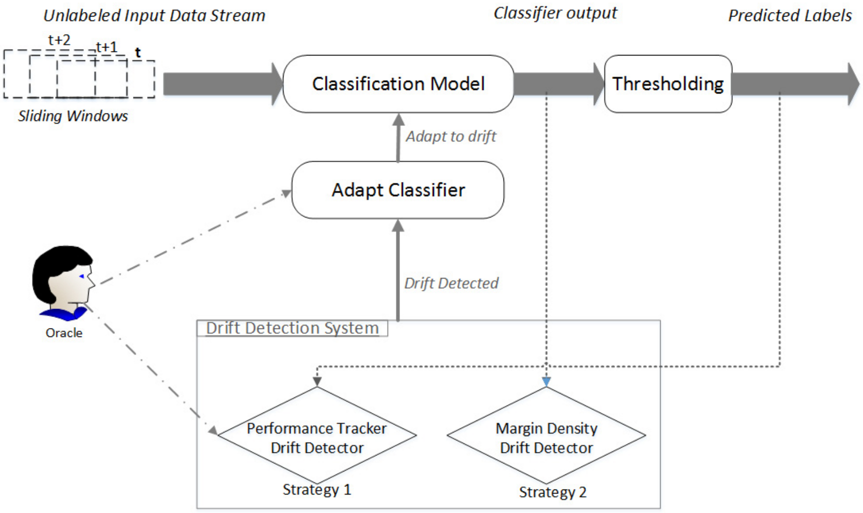


Figure 3: Framework of a basic drift handling system. The Performance tracker and the Margin density drift detection systems are two alternate strategies for drift detection.

computed from the classifier's predicted posterior estimates and is used in the drift detection process. The **Perf**ormance **Tr**acker drift detector (PerfTr), is the baseline fully labeled drift detector which explicitly tracks classification performance, based on the final predicted values and corresponding feedback from an Oracle. These are two alternate strategies for drift detection and while labels are required by both approaches to perform adaptation of the classifier, the MD3 approach does not need any labels till a drift is detected.

Def. 1 (**Margin density-ρ**). *The ratio of unlabeled samples in the current sliding window that fall within the classifier's region of uncertainty.*

3.1 Margin density and SVM

The concept of margin density is applicable for any probabilistic classifier which generates posterior probability estimates of class labels. This concept of margin is intuitive in case of a Linear SVM with soft margins, which finds an optimal maximum width separating hyperplane, between samples of two classes [16]. The soft margin allows for non separable cases, by introduction of non negative slack variables ξ_i, to measure the degree of misclassification of sample x_i. The optimization function of the soft margin SVM is given in Equation 1 and 2, where w is the normal vector of the separating hyperplane given by $w.x + b = 0$, y_i is the class label and C is the cost parameter which determines misclassification weight.

$$min \quad \frac{1}{2}w^T w \quad + \quad C\sum_{i=1}^{m}\xi_i \tag{1}$$

$$s.t. \quad y_i \quad (x_i^T w + b) \quad \geq \quad 1- \quad \xi_i \quad \xi_i \geq 0 \tag{2}$$

Soft constraints allow samples to enter the SVM margin while trying to minimize Equation 1. After training an SVM based on these constraints, there is an expected number of samples that fall within the margins, represented as $w.x - b = \pm 1$, which is the margin density to be monitored as a signal for drift indication. The margin density for every sliding window is computed based on the decision function of the SVM and in case of the linear kernel it is given by Equation 3.

$$\rho = \frac{\#samples\ with\ abs(w.x + b) \leq 1}{\#samples} \tag{3}$$

The signal described by Equation 3 is computed for every sliding window and is used in drift detection as described in Algorithm 1. The procedure receives a chunk of unlabeled samples at timestep t, an existing classification model (for linear SVM given by w,b) representing the concept at t-1, history statistics representing range variation in the margin density since the last drift(ρ_{min}, ρ_{max}) and the threshold to signal drift(θ_ρ). It proceeds by computing the margin density measure for the current chunk as given by Equation 3. The algorithm keeps track of the maximum and minimum margin density seen since the last adaptation and triggers an alert when this range exceeds the density threshold θ_ρ. This range based approach is much more effective at capturing gradual drift when compared with the sequential approaches of CUSUM and the Page-Hinkley test [8], which compute averages of signal so far and as a result miss out on slow moving drifts. After a drift is signaled, the margin density approach reassigns the ρ_{max} and ρ_{min} to the margin density on chunk t+1, using the adapted model, and continues with its operations as earlier.

When the drift detecting system detects a drift, the classifier is adapted based on feedback from an Oracle. This step is common for both MD3 and the labeled PerfTr techniques of Fig. 3 and is the only step in the margin density's pipeline which needs labeled data. In this paper, the adaptation is done by labeling and retraining a model based on the recent half of the sliding window samples. This is done to have a uniform evaluation of the drift detectors only. Active learning strategies such as uncertainty sampling [20], can be used to reduce the complexity and expense of retraining.

```
Input  : Classification Model: {w,b}, Current unlabeled data chunk X[start:end],
         History Statistics: [$\rho_{min}$, $\rho_{max}$], Drift Threshold: $\theta_\rho$
Output: True if drift is detected/False otherwise, Updated [$\rho_{min}$, $\rho_{max}$]
isDrift=False ;
// Compute margin density of current chunk : $\rho_{current}$;
while $x_i$ in $X$ do
    if $absolute(w.x_i + b \leq 1)$ then
        $\rho_{current}$=$\rho_{current}$+1 ;
    end
end
$\rho_{current}$=$\rho_{current}$/(end-start+1) ;
if $\rho_{current} < \rho_{min}$ then
    $\rho_{min}$=$\rho_{current}$ ;
end
if $\rho_{current} > \rho_{max}$ then
    $\rho_{max}$=$\rho_{current}$ ;
end
if $\rho_{max}$-$\rho_{min} > \theta_\rho$ then
    isDrift = True ;
end
return isDrift, $\rho_{max}$ ,$\rho_{min}$;
```

Algorithm 1: The MD3 drift detection algorithm using a Linear SVM as a base model

4 Experimental analysis of the MD3 methodology

The proposed MD3 methodology for drift detection is evaluated on 5 synthetic and 5 real world concept drift datasets and its performance, in terms of accuracy and the labeling used is presented in Section 4.2. An illustrative proof of idea on a simple 2 dimensional synthetic dataset is presented in Section 4.1. All implementation and experimentation was performed using Python 2.7 and the scikit-learn machine learning library [1]. The experiments take the first 5% of the stream as the initial labeled datasets to build the initial model, and then test the performance on the rest of the stream. The SVM classifier with linear kernel is used as a base model for the system and the following are the methodologies used for the comparative analysis:

- **Baseline static model (NoChange):** Assumes that there is no drift in the stream after the initial training phase, thereby needs no adaptation or labeling.
- **Performance Tracking Drift Detection (PerfTr):** PerfTr model explicitly requires labels for every sample to compute error and a significant drop in accuracy signals a drift requiring adaptation.
- **Margin Density Drift Detection(MD3):** The MD3 model uses the Algorithm1 to perform the drift detection and needs labeled samples only when adaptation is needed after drift detection.

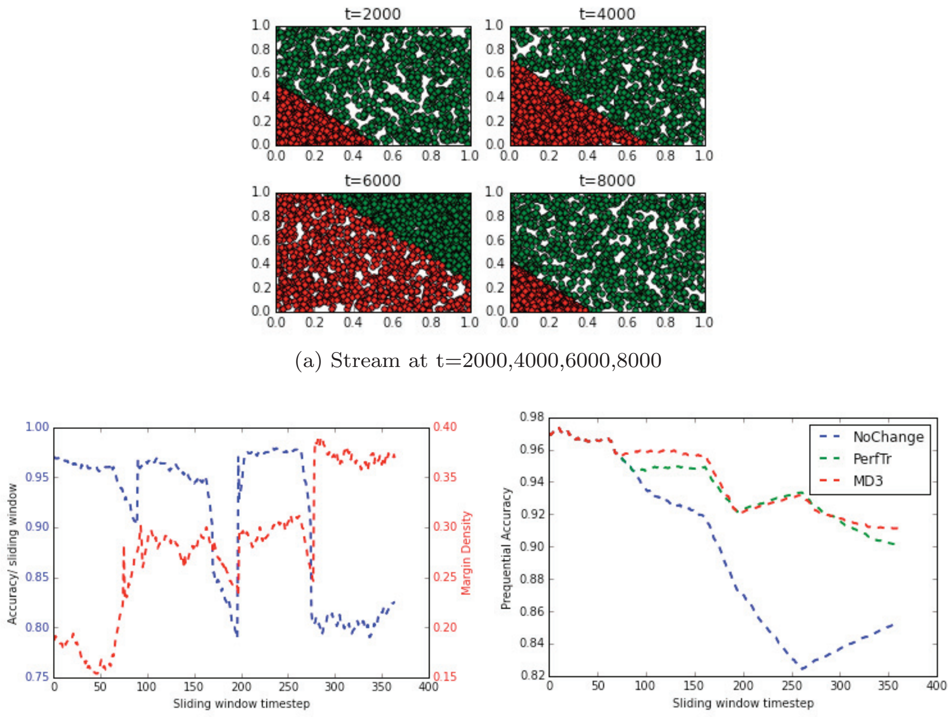
The goal of the experimentation is to evaluate the efficacy of the margin density as a viable signal to detect drift and to see if the MD3 approach can provide statistically similar performance as the PerfTr approach while requiring significantly lower labeled samples.

4.1 Proof of idea on SimpleStream synthetic dataset

As a proof of idea, the MD3 approach is evaluated on an illustrative 2D synthetic dataset: the SimpleStream dataset, consisting of 8000 samples and 4 concepts, as shown in Fig. 4a. This dataset enables us to gain intuition into the working of the algorithm as the drifts are pre known and the ability of the MD3 approach to detect the drifts swiftly can be easily visualized. For the purposes of fine grained initial analysis, the sliding window size was taken as 800 samples and the slide rate was taken to be 800/40 = 20 samples. The change threshold is taken as 6.5%, for illustration of change, from the suggested range of [5-10%].

The superimposed plots of the margin density signal and the accuracy signal are shown in Fig. 4b. It is observed that, every drop in accuracy, has a corresponding rise/drop in the margin density with high temporal proximity, making it suitable as an alias for tracking changes using unlabeled data. Drift was detected by the PerfTr approach at timesteps: 2560, 4140, 4660, 6200 and by the MD3 at : 2260, 2620, 2680, 6260. The resulting performance in Fig. 4c, computed as the prequential accuracy [4] over the stream, shows that both approaches produce similar results which are significantly better than static model, that assumes no change. However in doing so, the margin density approach uses labeled data only from 4 windows, in which the drift was detected, for retraining the model, while the PerTr approach needed labeled data from all windows for validation. The MD3 used only 44.4% labeling to produce similar performance as the 100% labeled performance tracker.

The distributions with respect to the initial SVM margin during the first drift at t=2000 of the SimpleStream is shown in Fig. 5. The change was caused by the movement of the class 0 samples(red) towards the hyperplane. This drift causes the accuracy of the model to drop and

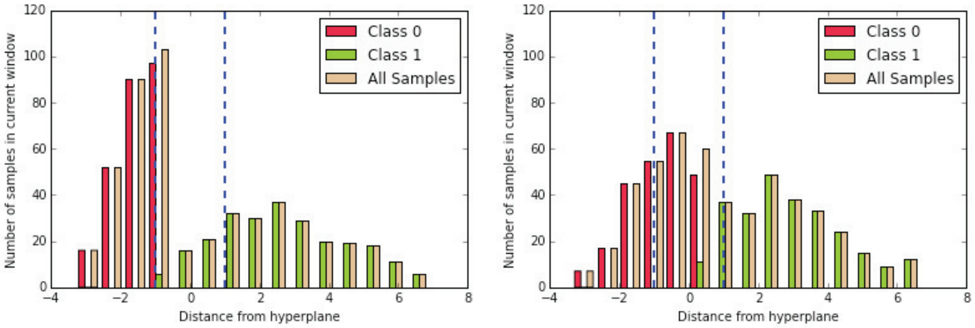


(a) Stream at t=2000,4000,6000,8000

(b) Tracking Accuracy(Blue) & Margin Density(Red)

(c) Prequential accuracy over time

Figure 4: Experimental results on the SimpleStream dataset



(a) Distribution of samples 1500-2000

(b) Distribution of samples 2000-2500

Figure 5: Changes in the distribution of samples, computed as distance from the SVM hyperplane, from timestep [1500,2000] to [2000,2500]

the density of the region to increase, similar to the scenario depicted in Fig. 2 b) and seen in Fig. 4b as a spike in density close to timestep 100. Either of these signals indicate drift and the ability of the margin density approach to recognize the latter, using only unlabeled data, is what makes it attractive as a concept drift detection approach.

Table 1: Description of datasets for experimental evaluation

Dataset	#Instances	#Attributes	Default accuracy	WindowSize
SimpleStream	8000	2	50%	300
Hyperplane1	10000	10	50.52%	300
Hyperplane2	10000	10	50.17%	300
Hyperplane3	10000	10	50.22%	300
SEA	60000	3	62.69%	800
Usenet	5931	659	50.43%	300
Spam	6213	500	66.7%	300
EM	45312	7	57.55%	800
Covtype	218515	54	67.93%	4000
Poker	765952	25	54.25%	4000

4.2 Experimental evaluation on real and synthetic datasets

For the purposes of experimental comparison and evaluation of the NoChange, MD3 and the PerfTr approach, 5 synthetic and 5 real world datasets were chosen. The characteristics of these datasets is shown in Table 1. The SimpleStream dataset was described in Section 4.1 and it represents a sudden drift dataset. The Hyperplane datasets are generated to have gradual drift with different rates and severity of drift [7]. The number of attributes affected by the drift and the rate of drift, given by (k,t) respectively [7], were chosen as (2,0.1) for Hyperplane1, (5,1.0) for Hyperplane2 and (8.1.0) for Hyperplane3. The SEA dataset demonstrates drift by transitioning between 4 different concepts which are described in terms of 2 out of 3 attributes of the data [17]. The real world datasets represent more practical scenarios where the drift is unknown and could potentially be caused by a variety of factors. Usenet and Spam are text concept drifting datasets with high number of dimensions [10]. The Electricity market(EM) dataset, Forest cover type(Covtype) and Poker datasets obtained from [2],are also popular for concept drift experimentation. All datasets were pre-processed by converting nominal attributes to binary and by normalizing the data to the range of [0,1]. Since the experimentation here are presented for binary classes only, the two most frequently occurring classes in the Covtype and Poker datasets were retained and the rest were eliminated.

To account for the diversity in the datasets, the window sizes were chosen, based on initial experimentation, according to the following rule: WindowSize= 300 if #Instances$\leq$10000, 800 if $\leq$ 100000 and 4000 otherwise. The window slide rate is taken as WindowSize/10. The SVM with Linear kernel is chosen as a base model for all approaches and the initial 5% of the stream is considered totally labeled for building the initial model. Based on several experiments, omitted here for brevity, the value of the change threshold for both the PerfTr and the MD3 approach is fixed at 7.5%.

The results shown in Table 2 indicate that all datasets have a significant concept drift which needs updation, as the NoChange approach has the lowest prequential accuracy in all cases. The Friedman's non parametric test followed by a post-hoc Nemenyi test and the Bonferroni-Dunn test [5] over the results showed that the performance, in terms of the prequential accuracy, of the PerfTr and the MD3 approaches are not significantly different at a p-value of 0.05, while both are significantly better than the NoChange approach. In term of the labeling, it is seen that MD3 uses only 13.8% labeled samples for the synthetic dataset and 48.66% for the real world datasets, which is significantly lower than the PerfTr which uses 100% labeling. It can be

Table 2: Prequential accuracy, Labeling used and Drifts detected for the NoChange, PerfTr & MD3 approaches

Dataset	*Prequential Accuracy*			*Labeling %*			*#Drifts detected*		
	NoChange	PerfTr	MD3	NoChange	PerfTr	MD3	NoChange	PerfTr	MD3
SimpleStream	86.33	93.05	93.34	0	100	27.63	0	5	14
Hyperplane1	63.89	86.37	86.68	0	100	17.37	0	5	11
Hyperplane2	64.26	86.85	87.1	0	100	20.53	0	7	13
Hyperplane3	72.03	87.05	87.18	0	100	12.63	0	5	8
SEA	84.49	84.58	84.64	0	100	4.91	0	3	7
Usenet	51.68	59.98	59.66	0	100	42.59	0	19	16
Spam	33.71	90.58	91.14	0	100	68.61	0	25	27
EM	62.97	66.46	65.75	0	100	47.39	0	37	51
Covtype	74.17	79.54	78.99	0	100	28.9	0	30	30
Poker	52.76	62.52	57.94	0	100	55.8	0	230	203

argued that the Perf-tracker approach can be made to work with partially labeled data as used in limited labeling techniques of [20]. But it should be noted that the margin density approach reduces labeling cost by requiring labeling only in chunks which have a suspected drift. In case of an underlying partial labeled approach, the MD3 retraining will also use partial labeling and as such will still have a lower labeling rate. However, the labeling ratio can dramatically increase if the number of drifts detected and the subsequent retraining needed is high. If drift is detected in every chunk, then the approach would degrade to a fully labeled approach. This is directly related to the ability of the Margin density approaches to deal with false positives. As seen from Table 1, the average number of drifts detected by the margin density approach is higher than the PerfTr approach. Nevertheless, this increase in drift detection, does not hinder it from providing statistically equivalent predictive performance at a much lower labeling rate.

5 Conclusion and future work

Detecting drifts is an important aspect of modern day stream classification algorithms. However, this detection process uses labeled data for continuous validation of the trained models. Labeled data is an expensive resource in streaming applications and is often not available. The MD3 approach proposed in this paper, uses the number of samples mapped to the uncertainty region of a classifier as a signal for drift. In doing so, it can detect drifts using unlabeled data and needs labeling only for retraining when drift is detected. Experimental evaluation of the MD3 framework on 10 different datasets shows that the approach produces statistically equivalent accuracy as a fully labeled performance tracking approach, and used only 32.6% labeled samples on average. An interesting area for future work would be the evaluation of the margin density signal in detecting drifts from imbalanced and multi-class data streams.

References

[1] `http://scikit-learn.org/stable/`.

[2] `http://moa.cms.waikato.ac.nz/datasets/`.

[3] Manuel Baena-García, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, Ricard Gavaldà, and Rafael Morales-Bueno. Early drift detection method. 2006.

[4] Albert Bifet, Jesse Read, Indrė Žliobaitė, Bernhard Pfahringer, and Geoff Holmes. Pitfalls in benchmarking data stream classification and how to avoid them. In *Machine Learning and Knowledge Discovery in Databases*, pages 465–479. Springer, 2013.

[5] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[6] Mark Dredze, Tim Oates, and Christine Piatko. We're not in kansas anymore: detecting domain changes in streams. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 585–595. Association for Computational Linguistics, 2010.

[7] Wei Fan. Systematic data selection to mine concept-drifting data streams. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 128–137. ACM, 2004.

[8] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[9] Paulo M Gonçalves, Silas GT de Carvalho Santos, Roberto SM Barros, and Davi CL Vieira. A comparative study on concept drift detectors. *Expert Systems with Applications*, 41(18):8144–8156, 2014.

[10] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, 22(3):371–391, 2010.

[11] Georg Krempl, Indre Žliobaite, Dariusz Brzeziński, Eyke Hüllermeier, Mark Last, Vincent Lemaire, Tino Noack, Ammar Shaker, Sonja Sievi, Myra Spiliopoulou, et al. Open challenges for data stream mining research. *ACM SIGKDD Explorations Newsletter*, 16(1):1–10, 2014.

[12] Carsten Lanquillon. Information filtering in changing domains. In *Proceedings of the international joint conference on artificial intelligence*, pages 41–48. Citeseer, 1999.

[13] Jeonghoon Lee and Frederic Magoules. Detection of concept drift for learning from stream data. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*, pages 241–245. IEEE, 2012.

[14] Patrick Lindstrom, Brian Mac Namee, and Sarah Jane Delany. Drift detection using uncertainty distribution divergence. *Evolving Systems*, 4(1):13–25, 2013.

[15] Mohammad M Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *Knowledge and Data Engineering, IEEE Transactions on*, 23(6):859–874, 2011.

[16] Andrew W Moore. Support vector machines. *Tutorial. School of Computer Science of the Carnegie Mellon University. Available at http://www. cs. cmu. edu/˜ awm/tutorials.[Accessed August 16, 2009]*, 2001.

[17] W Nick Street and YongSeog Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM, 2001.

[18] Dang-Hoan Tran, Mohamed Medhat Gaber, and Kai-Uwe Sattler. Change detection in streaming data in the era of big data: models and issues. *ACM SIGKDD Explorations Newsletter*, 16(1):30–38, 2014.

[19] Indre Zliobaite. Change with delayed labeling: when is it detectable? In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 843–850. IEEE, 2010.

[20] Indre Zliobaite, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. Active learning with drifting streaming data. *IEEE transactions on neural networks and learning systems*, 25(1):27–39, 2014.