

# Classifier Ensembles for Detecting Concept Change in Streaming Data: Overview and Perspectives

Ludmila I. Kuncheva<sup>1</sup>

**Abstract.** We address adaptive classification of streaming data in the presence of concept change. An overview of the machine learning approaches reveals a deficit of methods for explicit change detection. Typically, classifier ensembles designed for changing environments do not have a bespoke change detector. Here we take a systematic look at the types of changes in streaming data and at the current approaches and techniques in online classification. Classifier ensembles for change detection are discussed. An example is carried through to illustrate individual and ensemble change detectors for both unlabelled and labelled data. While this paper does not offer ready-made solutions, it outlines possibilities for novel approaches to classification of streaming data.

**Keywords:** Concept drift, Change detection, Changing Environments, Classifier ensembles, Online classifiers

## 1 INTRODUCTION

It has been argued that the current state-of-the-art in pattern recognition and machine learning is falling short to answer modern challenges in classification [12]. Among these challenges is classification of streaming data, especially when the data distribution changes over time. The ideal classification scenario is to detect the changes when they come, and retrain the classifier automatically to suit the new distributions. Most methods for novelty detection rely on some form of modelling of the probability distributions of the data and monitoring the likelihood of the new-coming observations [27]. These methods work well for small number of features and static distributions. They typically require access to all past data without strong considerations about processing time. Detecting changes in multi-dimensional streaming data has been gaining speed in recent years [25, 36]. The most useful indicator of adverse change, however, is a peak or a steady upward trend in the running error of the online classifier (or classifier ensemble). Thus the change detection method receives as input a binary string (correct/wrong label), and raises an alarm if a “sizable” increase in the error rate has occurred. Simple as it looks, the problem of designing a reliable and, at the same time, sensitive change detector is far from being solved.

## 2 A LANDMAP OF CLASSIFICATION APPROACHES FOR STREAMING DATA

The approaches to handling concept drift in streaming data can be categorised with respect to the following properties

- *Instances versus batches of data.* Streaming instances of data can be converted into streaming batches or “chunks” of data. The reverse is also possible but batch data usually comes in massive quantities, and instance-based processing may be too time-consuming.
- *Explicit versus implicit change detection.* After explicit change detection, action is taken, for example, setting up a window of latest data to re-train the classifier. Implicit detection is equivalent to using a forgetting heuristic regardless of whether or not change is suspected. For example, using an online classifier ensemble, the weights of the ensemble members are modified after each instance, based on the recent accuracy records of the ensemble members. Without concept drift, the classification accuracy will be stable and the weights will converge. Conversely, if change does occur, the weights will change to reflect it without the need of explicit detection.
- *Classifier-specific versus classifier-free.* In the classifier-specific case, the forgetting mechanism can only be applied to a chosen and fixed classifier model [1, 3] or classifier ensemble [30]. In the classifier-free case any classifier can be used because the change detection and the window update depend only on the accuracy of the classification decision, and not on the model [10, 20].
- *Classifier ensembles versus single classifiers.* A categorisation of classifier ensemble methods for changing environment is offered in [22]. The methods are grouped with respect to how they adapt to the concept drift: by updating the combination rule for fixed classifiers (“horse racing”); by using adaptive online classifiers as the ensemble members; and by changing the ensemble structure (“replace the loser”).

Explicit or implicit detection of concept drift can be based upon change in:

**A. Probability distributions.** If the class-conditional distributions or prior probabilities for the classes drift away from their initial values, the new data will not fit the old distributions. Based on how well the assumed distribution accommodates most recent data, a change can be detected and old data should be forgotten [4, 7, 9, 11, 27, 33]. Methods for change detection in this case include estimating the likelihood of new data with respect to the assumed distributions, and comparing the likelihood with a threshold.

**B. Feature relevance.** A concept drift may lead to a different relevance pattern of the features describing the instances. Features or even combinations of attribute values that were relevant in the past may no longer be sufficiently discriminative [3, 8, 13, 40]. Keeping track on the best combination of predictive features (clues) makes it possible to train an up-to-date classifier for the most recent data distribution.

<sup>1</sup> School of Computer Science, Bangor University, United Kingdom, e-mail: l.i.kuncheva@bangor.ac.uk. In *Proceedings of the Second Workshop SUEMA, ECAI 2008, Partas, Greece*, July 2008, pp. 5–9.

**C. Model complexity.** Some classifier models are sensitive to change in the data distribution. For example, explosion of the number of rules in rule-based classifiers or the number of support vectors in SVM classifiers may signify concept drift.

**D. Classification accuracy.** This is the most widely used criterion for implicit or explicit change detection. Included in this group are most ensemble methods for changing environments: Winnov variants [24, 28], AdaBoost variants [5, 30], “replace-the-loser” approaches [21, 22, 34, 35, 39]. Many single classifier models also evaluate the accuracy either to select the window size for the next classifier [2, 10, 18–20, 23] or to update the current model [1, 15, 41].

### 3 TYPES OF CHANGES AND THEIR DETECTION

Change may come as a result of the changing environment of the problem (we call this “novelty”), e.g., floating probability distributions, migrating clusters of data, loss of old and appearance of new classes and/or features, class label swaps, etc.

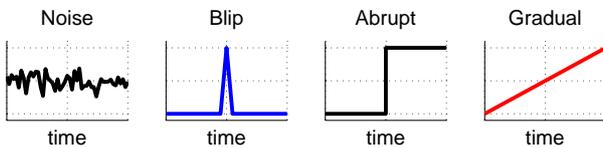


Figure 1. Types of concept change in streaming data

Figure 1 shows four examples of simple changes that may occur in a single variable along time. The first plot (Noise) shows changes that are deemed non-significant and are perceived as noise. The classifier should not respond to minor fluctuations, and can use the noisy data to improve its robustness for the underlying stationary distribution. The second plot (Blip) represents a ‘rare event’. Rare events can be regarded as outliers in a static distribution. Examples of such events include anomalies in landfill gas emission, fraudulent card transactions, network intrusion and rare medical conditions. Finding a rare event in streaming data can signify the onset of a concept drift. Hence the methods for online detection of rare events can be a component of the novelty detection paradigm. The last two plots in Figure 1 (Abrupt) and (Gradual) show typical examples of the two major types of concept drift represented in a single dimension.

#### 3.1 Rare events detection in batch data

Finding outliers in batch 1-dimensional static data has a longstanding history within statistical literature [14]. Hodge and Austin [14] point out that there is no universally accepted definition of outlier, and that alternative terminologies pervade in the recent literature, e.g., novelty detection, anomaly detection, noise detection, deviation detection or exception mining. Here we adopt the following definition: “An outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.” In this simplest set up, one can model the relevant probability distributions and calculate the likelihood of the suspect observations. A threshold on this likelihood will determine whether an observation should be marked as an outlier. Outlier detection methods for static data typically assume unlimited (finite) computation time, and unrestricted access to every observation in the data.

In multi-dimensional static data, outliers are not as obvious. Such outliers, e.g., fraudulent telephone calls, may need to be labelled by hand and communicated to the system designer for an outlier detection algorithm to be trained. The main difficulty compared to the one-dimensional case is that the approximation of multi-dimensional probability density functions is not as trivial. Detecting outliers in multidimensional batch data can be approached as a pattern recognition task. Assuming that outliers are very rare, the data can be used to train the so called “one-class classifier”. In a way, the training will lead to a form of approximation of the pdf for class “norm” or will evaluate geometrical boundaries in the multi-dimensional space, beyond which an observation is labelled as an outlier [16, 26, 38]. Alternatively, the probability density functions of the individual variables can be modelled more precisely, and an ensemble of one-class classifiers may be more accurate in identifying rare events [6, 37].

#### 3.2 Novelty (concept drift) detection in streaming data

For streaming data, the one-dimensional change detection has been extensively studied in engineering for the purposes of process quality control (control charts). Consider a streaming line of objects with probability  $p$  of an object being defective. Samples of  $N$  objects (batches) are taken for inspection at regular intervals. The number of defective objects is counted and an estimate  $\hat{p}$  is plotted on the chart. It is assumed that the true value of  $p$  is known (from product specification, trading standards, etc.) Using a threshold of  $f\sigma$ , where  $\sigma = \sqrt{p(1-p)/N}$ , a change is detected if  $\hat{p} > p + f\sigma$ . This model is known as *Shewhart* control chart, or also  $p$ -chart when binary data is being monitored. The typical value of  $f$  is 3, but many other alternative and compound criteria have been used<sup>2</sup>. Better results have been reported with the so called CUSUM charts (CUmulative SUM) in terms of detecting small changes [29, 32]. Reynolds and Stoumbos [31] advocate another detection scheme based on Wald’s Sequential Probability Ratio Test (SPRT). SPRT charts are claimed to be even more successful than CUSUM charts.

Finding rare events in multi-dimensional streaming data, and especially when changing environments may be present, has not been systematically explored. While there are tailor-made solutions for specific problems, new technologies are still in demand. The prospective solutions may draw upon sources of information A-D detailed in Section 2.

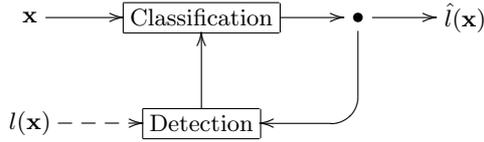
### 4 CLASSIFIER ENSEMBLES FOR CHANGE DETECTION

Note that the change-detection process is separate from the classification. Below we illustrate the operation of the classifier equipped with a change detector. Upon receiving a new data point  $\mathbf{x}$  the following set of actions takes place before proceeding with the next point in the stream:

- (1) The classifier predicts a label for  $\mathbf{x}$ .
- (2) The true label is received.
- (3)  $\mathbf{x}$ , the predicted and the true labels are submitted to the change detector.
- (4) If change is detected, the classifier is re-trained.

The classification of a data point from the stream is shown diagrammatically in Figure 2.

<sup>2</sup> [http://en.wikipedia.org/wiki/Control\\_chart](http://en.wikipedia.org/wiki/Control_chart)



**Figure 2.** Online classification with explicit detection of concept change. Notations:  $\hat{l}(\mathbf{x})$  is the class label assigned by the classification block (a single classifier or an ensemble);  $l(\mathbf{x})$  is the true label of  $\mathbf{x}$  received after the classification. The result of the change detection is an action related to the classifier training.

The ideal detector will signal a change as soon as it occurs (true positive) and will not signal false changes. In practice, the data needed to detect the change come after the change, so a delay of  $\Delta$  observations is incurred. The expected number of batches of data to detection is called the Average Run Length (ARL) in control chart literature. Here we are interested in the average number of observations to detection. The desired characteristics of a change detector are  $\Delta_{\text{true}+} \rightarrow 0$  and  $\Delta_{\text{false}+} \rightarrow \infty$ . Hence change detectors can be compared with respect to their  $\Delta$ s

While classification ensembles are a popular choice now for classification of streaming data with concept drift, explicit change detection is usually only mentioned on the side. We are interested in the change detector, and in the possibility to implement it as a “detection ensemble”.

#### 4.1 Detection from unlabelled data

Changes in the unconditional probability distribution may or may not render the current classifier inaccurate. Consider the following example. In a classification problem with two equiprobable Gaussian classes, both classes migrate apart, symmetrically about the optimal discriminant line. Suppose that the perfect classifier has been trained on the original distributions. Even though there are changes in the underlying probability density function (pdf), the classifier need not change as it will stay optimal.

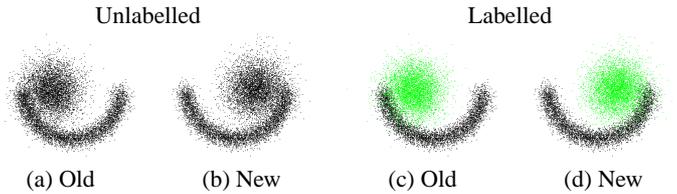
When will change detection from unlabelled data be useful? Here are three possible answers:

1. *More sensitive to error.* When change in the distributions can be detected early to herald a future error change.
2. *Label delay.* The labels are not available at the time of classification but come with a substantial delay. For example, in applying for a bank loan, the true class label (good credit/bad credit) becomes known two years after the classification has taken place [17].
3. *Missed opportunities.* The chance to improve the classifier may be easily ignored if only labelled data is used for the change detection. It is always assumed that a change will adversely affect the classifier, giving raise to the classification error. Consider again the example with the two equiprobable Gaussian classes. Suppose now that one of the classes migrates away from the discrimination line, while the other stays in its original place. The classification error of the initial classifier will drop a little because there will be much fewer errors from the migrated class. In this case, however, if change is not detected, we are missing the opportunity to train a new classifier with much smaller error.

The most popular method for detecting change from unlabelled streaming data is to monitor the cluster structure of the data and signal a change when a “notable” change occurs, e.g. migration of clusters, merging, appearance of new clusters, etc. The monitoring can

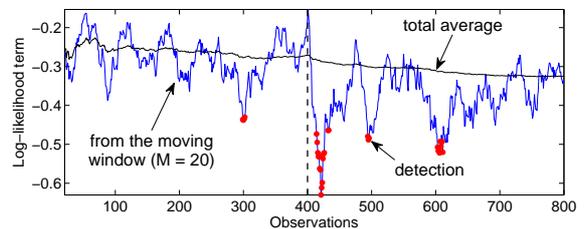
be done using a sliding window of fixed size to re-calculate the cluster parameters. These parameters are then compared with the previous ones. If the difference exceeds a given threshold, change is signalled. More formally, the cluster-monitoring approach can be cast as fitting a mixture of Gaussians and monitoring the parameters, e.g., means of the components (centres of the clusters), covariance matrices (shapes of the clusters) or mixture coefficients (cluster density). The likelihood of the sample in the window can be calculated. There are at least two parameters of the algorithm that need to be set up in advance: the size of the sliding window and the threshold for comparison of the latest likelihood with the old one. This makes way for an ensemble approach.

An ensemble for change detection from unlabelled data can be constructed by putting together detectors with different parameter values. An example is discussed next. Figure 3 shows the “old” and the “new” distributions.



**Figure 3.** Old and new distributions, with and without labels

First, 400 i.i.d data points are generated from the old distribution, followed by 400 points from the new distribution. The classes (black and green) are generated with equal prior probabilities. The following protocol is used. The first  $M$  observations are taken as the first window  $W_1$ . K-means clustering is run on this data, where the number of clusters,  $K$  is fixed in advance (we chose  $M = 20$ ,  $K = 2$ ). It can be shown that, under some basic assumptions, the log-likelihood of the data in window  $W_i$  is related to the sum of the distances of the data points to their nearest cluster centre. We compare the mean log-likelihood of window  $W_i$  ( $L_i$ ) with the mean log-likelihood of the stream so far ( $\bar{L}_i$ ). Change is signalled if  $L_i < \bar{L}_i + 3\sigma_i/\sqrt{M}$ , where  $\sigma_i$  is the standard deviation of the log-likelihood up to point  $i$ . The cluster means are updated after each observation. Figure 4 shows the running mean (total average,  $\bar{L}_i$ ), the mean of the log-likelihood of the moving window ( $L_i$ ), the change point at observation 400, and the detected changes.

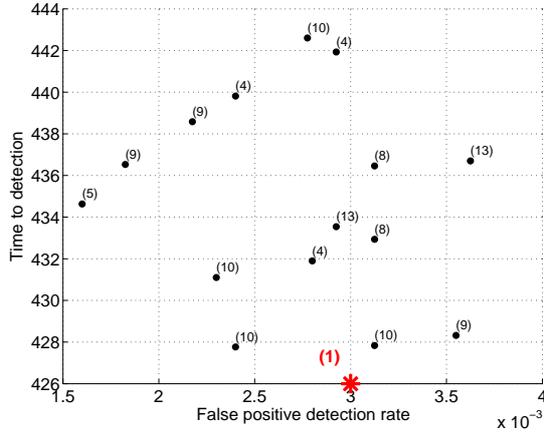


**Figure 4.** Change detection (single detector) from unlabelled data using the likelihood of a data window of size  $M = 20$ .

The detection pattern is not perfect. There were 4 false detections in this run (observations 299-302), coming before the true change. On the other hand, after the change, consistent detection is signalled for all observations from 414 to 426. Later on, since there is no for-

getting in this scenario, the means start moving towards the second distribution, and the log-likelihood will eventually stabilise.

The benefit of using an ensemble is demonstrated in Figure 5.



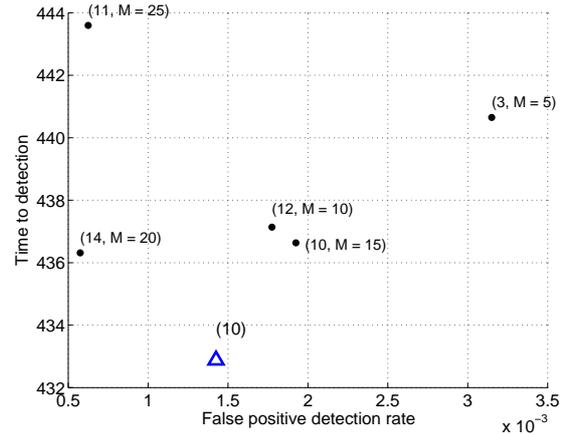
**Figure 5.** Single and ensemble detectors: **unlabelled** data. The numbers in brackets show in how many out of the 100 runs the respective detector has missed the change altogether.

All values of  $M = \{5, 10, 15, 20, 25\}$  and  $K = \{2, 3, 4\}$  were tried, resulting in 15 individual detectors. One hundred runs were carried out with each combination  $(M, K)$  with different data generated from the distributions in question. The dots in the figure correspond to the individual detectors, where the false positive rate and the true positive detection times were averaged across the 100 runs. An ensemble was created by pooling all 15 individual detectors. The ensemble signals a detection if at least two individual detectors signal detection. The star in the figure shows the ensemble detector. The number in brackets shows the number of runs, out of 100, where no change was detected after observation 400. Better detectors are those that have both small false positive rate and small time to detection  $\Delta_{\text{true}+}$ . The ensemble has the smallest time to detection. While its false detection rate is not the best, the ensemble missed only one change

## 4.2 Detection from labelled data

The strongest change indication is a sudden or gradual drop in the classification accuracy, consistent over a period of time. An ensemble detector can make use of different window sizes, thresholds or detections heuristics. To illustrate this point we continue the example shown in Figure 3, this time with the true class labels supplied immediately after classification. For the purposes of the illustration we chose the nearest mean classifier (NMC), in spite of the fact that it is not optimal for the problem. The same values of  $M$  were used as in the detection from unlabelled data. NMC was built on the first window  $W_1$  and updated with the new observations without any forgetting. The guessed label of observation  $i$  was obtained from the current version of NMC. The true label was received and the error was recorded (0 for no error, 1 for error). NMC was retrained using observation  $i$  and the true label. The running error was the error of NMC built on all the data from the start up to point  $i$ . The window of size  $M$  containing the error records of the past  $M$  observations was used to estimate the current error and compare it with the mean error hitherto. The 3 sigma threshold was used again. The error rate does not increase dramatically after observation 400 where the old

and the new distributions are swapped. Hence the change detection patterns were erratic, often not signalling a change throughout the whole run. The ensemble was constructed by pooling the detectors for the 5 different values of  $M$ . The ensemble signals a change at observation  $i$  if at least one of the detectors signals a change. Figure 6 shows the individual detector scores and the ensemble score for the labelled data case. The same two axes are used: false positive rate and the detection time (Note that Detection time =  $400 + \Delta_{\text{true}+}$ ). The numbers in brackets show in how many out of the 100 runs the respective detector has missed the change altogether. The ensemble detector again has the smallest time to detection. The two individual detectors with better false detection rate have a larger number of missed detections as well as higher time to detection.



**Figure 6.** Single and ensemble detectors: **labelled** data. The numbers in brackets show in how many out of the 100 runs the respective detector has missed the change altogether.

The control chart detection methods fall in the category of detectors from labelled data. Ensembles of control charts may be an interesting research direction. Based on the currently available theoretical fundament, desirable ensemble properties may turn out to be theoretically provable.

## 5 CONCLUSIONS

The true potential of ensembles for change detection lies in the ability of different detection modes and information sources to sense different types and magnitudes of change. The purpose of this study was not to offer answers but rather to open new questions. The idea of constructing an ensemble for change detection was suggested and illustrated using both labelled and unlabelled data. The detection ensembles could be homogeneous (using the same detection methods with different parameters) as well as heterogeneous (detection based on labelled and unlabelled data, feature relevance, and more). The detector may be used in conjunction with the classifier. For example, the detector may propose how long after the detection the old classifier can be more useful than an undertrained new classifier. With gradual changes, the detector can be used to relate the magnitude of the change with the size of a variable training window for the classifier. Various problem-specific heuristics can be used to combine the individual detectors. Control chart methods come with a solid body of theoretical results that may be used in designing the ensemble detectors.

One possibility which was not discussed here is to couple detectors with the members of the classifier ensemble responsible for the labelling of the streaming data. Thus various changes can be handled by different ensemble members, and a single overall change need not be explicitly signalled.

## REFERENCES

- [1] D. W. Aha, D. Kibler, and M. K. Albert, 'Instance-based learning algorithms', *Machine Learning*, **6**, 37–66, (1991).
- [2] M. Baena-García, J. Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, 'Early drift detection method', in *Fourth International Workshop on Knowledge Discovery from Data Streams*, pp. 77–86, (2006).
- [3] A. Blum, 'Empirical support for Winnow and weighted-majority based algorithms: results on a calendar scheduling domain', in *Proc. 12th International Conference on Machine Learning*, pp. 64–72, San Francisco, CA., (1995). Morgan Kaufmann.
- [4] S. J. Delany, P. Cunningham, and A. Tsybmal, 'A comparison of ensemble and case-based maintenance techniques for handling concept drift in spam filtering', Technical Report TCD-CS-2005-19, Trinity College Dublin, (2005).
- [5] W. Fan, S. J. Stolfo, and J. Zhang, 'Application of adaboost for distributed, scalable and on-line learning', in *Proc KDD-99*, San Diego, CA, (1999). ACM Press.
- [6] T. Fawcett and F. J. Provost, 'Adaptive fraud detection', *Data Mining and Knowledge Discovery*, **1**(3), 291–316, (1997).
- [7] G. Folino, C. Pizzuti, and G. Spezzano, 'Mining distributed evolving data streams using fractal GP ensembles', in *Proceedings of EuroGP*, volume LNCS 4445, pp. 160–169. Springer, (2007).
- [8] G. Forman, 'Tackling concept drift by temporal inductive transfer', Technical Report HPL-2006-20(R.1), HP Laboratories Palo Alto, (June 2006).
- [9] M.M. Gaber and P.S. Yu, 'Classification of changes in evolving data streams using online clustering result deviation', in *Third International Workshop on Knowledge Discovery in Data Streams*, Pittsburgh PA, USA, (2006).
- [10] J. Gama, Pedro Medas, G. Castillo, and P. Rodrigues, 'Learning with drift detection', in *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence*, volume 3171 of *Lecture Notes in Computer Science*, pp. 286–295. Springer Verlag, (2004).
- [11] V. Ganti, J. Gehrke, and R. Ramakrishnan, 'Mining data streams under block evolution', *ACM SIGKDD Explorations Newsletter*, **3**, 1–10, (2002).
- [12] D.J. Hand, 'Classifier technology and the illusion of progress (with discussion)', *Statistical Science*, **21**, 1–34, (2006).
- [13] M. Harries and K. Horn, 'Detecting concept drift in financial time series prediction using symbolic machine learning', in *Eighth Australian Joint Conference on Artificial Intelligence*, pp. 91–98, Singapore, (1995). World Scientific Publishing.
- [14] V.J. Hodge and J. Austin, 'A survey of outlier detection methodologies', *Artificial Intelligence Review*, **22**(2), 85–126, (2004).
- [15] G. Hulten, L. Spencer, and P. Domingos, 'Mining time-changing data streams', in *In Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 97–106. ACM Press, (2001).
- [16] P. Juszczak, *Learning to recognise. A study on one-class classification and active learning*, Ph.D. dissertation, Delft University of Technology, 2006.
- [17] M. G. Kelly, D. J. Hand, and N. M. Adams, 'The impact of changing populations on classifier performance', in *KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 367–371, (1999).
- [18] R. Klinkenberg, 'Using labeled and unlabeled data to learn drifting concepts', in *Workshop notes of the IJCAI-01 Workshop on Learning from Temporal and Spatial Data*, pp. 16–24, Menlo Park, CA, USA, (2001). IJCAI, AAAI Press.
- [19] R. Klinkenberg and T. Joachims, 'Detecting concept drift with support vector machines', in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pp. 487–494, San Francisco, CA, USA, (2000). Morgan Kaufmann.
- [20] R. Klinkenberg and I. Renz, 'Adaptive information filtering: Learn-
- ing in the presence of concept drifts', in *AAAI-98/ICML-98 workshop Learning for Text Categorization*, Menlo Park, CA, (1998).
- [21] J. Z. Kolter and M. A. Maloof, 'Dynamic weighted majority: A new ensemble method for tracking concept drift', in *Proc 3rd International IEEE Conference on Data Mining*, pp. 123–130, Los Alamitos, CA, (2003). IEEE Press.
- [22] L. I. Kuncheva, 'Classifier ensembles for changing environments', in *5th International Workshop on Multiple Classifier Systems, MCS 04, LNCS*, volume 3077, pp. 1–15. Springer-Verlag, (2004).
- [23] M.M. Lazarescu and S. Venkatesh, 'Using selective memory to track concept drift effectively', in *Intelligent Systems and Control*, volume 388, Salzburg, Austria, (2003). ACTA Press.
- [24] N. Littlestone, 'Learning quickly when irrelevant attributes abound: A new linear threshold algorithm', *Machine Learning*, **2**, 285–318, (1988).
- [25] J. Ma and S. Perkins, 'Online novelty detection on temporal sequences', in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 613–618. ACM Press, (2003).
- [26] L. M. Manevitz and M. Yousef, 'One-class svms for document classification', *Journal of Machine Learning Research*, **2**, 139–154, (2001).
- [27] M. Markou and S. Singh, 'Novelty detection: A review. Part I: Statistical approaches', *Signal Processing*, **83**(12), 2481–2521, (2003).
- [28] C. Mesterharm, 'Tracking linear-threshold concepts with winnow', *Journal of Machine Learning Research*, **4**, 819–838, (2003).
- [29] E. S. Page, 'Continuous inspection schemes', *Biometrika*, **41**, 100–114, (1954).
- [30] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, 'Learn++: An incremental learning algorithm for supervised neural networks', *IEEE Transactions on Systems, Man and Cybernetics*, **31**(4), 497–508, (2001).
- [31] M. R. Reynolds Jr and Z. G. Stoumbos, 'The SPRT chart for monitoring a proportion', *IIE Transactions*, **30**, 545–561, (1998).
- [32] M. R. Reynolds Jr and Z. G. Stoumbos, 'A general approach to modeling CUSUM charts for a proportion', *IIE Transactions*, **32**, 515–535, (2000).
- [33] M. Salganicoff, 'Density-adaptive learning and forgetting', in *Proceedings of the 10th International Conference on Machine Learning*, pp. 276–283, (1993).
- [34] K. O. Stanley, 'Learning concept drift with a committee of decision trees', Technical Report AI-03-302, Computer Science Department, University of Texas-Austin., (2003).
- [35] W. N. Street and Y. S. Kim, 'A streaming ensemble algorithm (SEA) for large-scale classification', in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377–382. ACM Press, (2001).
- [36] J. Takeuchi and K. Yamanishi, 'A unifying framework for detecting outliers and change points from time series data', *IEEE Transactions on Knowledge and Data Engineering*, **18**(4), 482–492, (2006).
- [37] D.M. Tax and R.P.W. Duin, 'Combining one-class classifiers', in *Proc. of the 2nd Int Workshop on MCS*, volume LNCS 2096, pp. 299–308. Springer, (2001).
- [38] D.M.J. Tax, *One-class classification: Concept-learning in the absence of counter-examples*, Ph.D. dissertation, Delft University of Technology, 2001.
- [39] H. Wang, W. Fan, P. S. Yu, and J. Han, 'Mining concept drifting data streams using ensemble classifiers', in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–235. ACM Press, (2003).
- [40] G. Widmer, 'Tracking context changes through meta-learning', *Machine Learning*, **27**(3), 259–286, (1997).
- [41] G. Widmer and M. Kubat, 'Learning in the presence of concept drift and hidden contexts', *Machine Learning*, **23**, 69–101, (1996).