On the Online Classification of Data Streams Using Weak Estimators

Hanane Tavasoli[1], B. John Oommen[1*], and Anis Yazidi[2]

[1] Department of Computer Science, Oslo and Akershus University College of Applied Sciences, Norway

[2] School of Computer Science, Carleton University, Ottawa, Canada

Abstract. In this paper, we propose a novel *online* classifier for complex data streams which are generated from non-stationary stochastic properties. Instead of using a single training model and counters to keep important data statistics, the introduced online classifier scheme provides a real-time self-adjusting learning model. The learning model utilizes the multiplication-based update algorithm of the Stochastic Learning Weak Estimator (SLWE) at each time instant as a new labeled instance arrives. In this way, the data statistics are updated every time a new element is inserted, without requiring that we have to rebuild its model when changes occur in the data distributions. Finally, and most importantly, the model operates with the understanding that the correct classes of previously-classified patterns become available at a later juncture subsequent to some time instances, thus requiring us to update the training set and the training model.

The results obtained from rigorous empirical analysis on multinomial distributions, is remarkable. Indeed, it demonstrates the applicability of our method on synthetic datasets, and proves the advantages of the introduced scheme.

Keywords : *Weak Estimators, Learning Automata, Non-Stationary Environments, Classification in Data Streams*

1 Introduction

In the past few years, due to the advances in computer hardware technology, large amounts of data have been generated and collected and are stored permanently from different sources. Some the applications that generate data streams are financial tickers, log records or click-streams in web tracking and personalization, data feeds from sensor applications and call detail records in telecommunications. Analyzing these huge amounts of data has been one of the most important challenges in the field of Machine Learning (ML) and Pattern Recognition (PR). Traditionally, ML methods are assumed to deal with static data

* *Chancellor's Professor*; *Fellow: IEEE* and *Fellow: IAPR*. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway.

stored in memory, which can be read several times. On the contrary, streaming data grows at an unlimited rate and arrives continuously in a single-pass manner that can be read only once. Further, there are space and time restrictions in analyzing streaming data. Consequently, one needs methods that are "automatically adapted" to update the training models based on the information gathered over the past observations whenever a change in the data is detected.

Mining streaming data is constrained by limited resources of time and memory. Since the source of data generates a potentially unlimited amount of information, loading all the generated items into the memory and achieving off-line mining is no longer possible. Besides, in non-stationary environments, the source of data may change over time, which leads to variations in the underlying data distributions. Thus, with respect to this dynamic nature of the data, the previous data model discovered from the past data items may become irrelevant or even have a negative impact on the modeling of the new data streams that become available to the system.

A vast body of research has been performed on the mining of data streams to develop techniques for computing fundamental functions with limited time and memory, and it has usually involved the sliding-window approaches or incremental methods. In most cases, these approaches require some *a priori* assumption about the data distribution or need to invoke hypothesis testing strategies to detect the changes in the properties of data.

In this article we will study classification problems in non-stationary environments (NSE), where sequential patterns are arriving and being processed in the form of a data stream that was potentially generated from different sources with different statistical distributions. The classification of the data streams is closely related to the estimation of the parameters of the time varying distribution, and the associated algorithms must be able to detect the source changes and to estimate the new parameters whenever a *switch* occurs in the incoming data stream.

Apart from the "traditional" classification problem involving *unique and distinct* training and testing phases, this paper pioneers the concept when these phases are not so clearly well-defined. Rather, we consider the fascinating phenomenon in which the testing patterns can subsequently be considered as training patterns, once their true class identities are known. Thus, the model operates with the understanding that the correct classes of previously-classified patterns become available at subsequent time instances (after some time has lapsed), thus requiring us to update the training set and the training model. This renders the whole PR problem intriguing.

Finally, to render the problem more complex, we consider the case where the classes' stochastic properties potentially vary with time as more instances become available. In this perspective, with regard to the training, we will argue that using "strong" estimators that converge with probability of 1 is inefficient for tracking the statistics of the data distributions in non-stationary environments. However, "weak" estimator approaches are able to rapidly unlearn what they have learned and adapt the learning model to new observations.

This feature of "weak" estimators makes these approaches the most effective methods for estimation in non-stationary environments. In this work, we will employ a particular family of weak estimators, referred to as Stochastic Learning Weak Estimation (SLWE) methods [5], for classification in non-stationary environments. The SLWE has been successfully used to solve *two*-class classification problems by Oommen and Rueda [5] by applying it on non-stationary *one*-dimensional datasets. In this article we will study the performance of the SLWE with more complex classification schemes.

1.1 Contributions of the paper

The main contributions of the paper are the following:

- We have pioneered an *online* classification scheme that is composed of three phases. In the first phase, the model learns from the available labeled samples. In the second phase, the learned model predicts the class label of the unlabeled instances currently observed. In the third phase, after knowing the true class label of these recently-classified instances, the classification model is adjusted in an *online* manner.
- Most of the data stream mining approaches have involved building an initial model from a sliding window of recently-observed instances and thereafter, refining the learning model periodically or whenever its performance degrades based on the current window of observed data. We present a novel framework to deal with concept and distribution drift over data streams in non-stationary environments, which is more efficient and provides more accurate results. We emphasize that this non-stationarity could even be *abrupt*.
- Our classifier scheme provides a real-time self-adjusting learning model, utilizing the multiplication-based update algorithm of the SLWE at each time instance, as new labeled instances arrive. Instead of using a single training model and maintaining counters to keep important data statistics, we have used a technique to replace these frequency counters by data estimators. In this way, the data statistics are updated every time a new element is observed, without needing to rebuild its model when a change in the distributions is detected.
- Extensive experimental results that we have obtained, for multi-dimensional distributions, demonstrate the efficiency of the proposed classification schemes in achieving a good performance for data streams involving non-stationary distributions under different scenarios of concept drift, and the new model of computation in which the training and testing samples are not completely dichotomized.

1.2 Organization of the paper

In Section 2, we proceed with discussing the issues and challenges encountered when one learns from data streams and provide a brief explanation about the theoretical properties of the SLWE. In Section 3, we present the details of the design and implementation of the online classifier where the samples that were testing samples at any given time instant can, at a subsequent juncture, be considered as training data. We then explain how this solution can be used to perform online classification, and present the new experimental framework for concept drift in Section 4. This section also contains the experimental results we have obtained from rigorous testing. Section 5 concludes the paper.

2 Literature Review

Estimation theory is a fundamental subject that is central to the fields of Pattern Recognition (PR) and data mining. The majority of problems in PR require the estimation of the unknown parameters that characterize the underlying data distributions.

2.1 Learning Methods for Data Streams in NSE

In general, most algorithms in the data stream mining literature have one or more of the following modules: a Memory module, an Estimator module, and a Change Detector module [1]. The Memory module is a component that stores summaries of all the sample data and attempts to characterize the *current* data distribution. Data in non-stationary environments can be handled by three different approaches, namely, by using partial memory, by window-based approaches and by instance-based methods. The term "partial memory" refers to the case when only a part of the information pertaining to the training samples are stored and used regularly in the training. In window-based approaches, data is presented as "chunks", and finally, in instance-based methods, the data is processed upon its arrival. In fact, the Memory module determines the forgetting strategy used by the mining algorithm operating in the dynamic environments.

The Estimator module uses the information contained in the Memory or only the observed information to estimate the desired statistics of the time varying streamed data. The Change Detector module involves the techniques or mechanisms utilized for detecting explicit drifts and changes, and provides an "alarm" signal whenever a change is detected based on the estimator's outputs.

Apart from the above schemes, many other incremental approaches have been proposed that infer change points during estimation, and use the new data to adapt the learning model trained from historical streaming data. The learning model in incremental approaches is adapted to the most recently received instances of the streaming data. Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of training examples available at time $t = 1 \ldots n$. An incremental approach produces a sequence of hypothesis $\{\ldots, H_{i-1}, H_i, \ldots\}$ from the training sequence, where each hypothesis, H_i, is derived from the previous hypothesis, H_{i-1}, and the example x_i. In general, in order to detect concept changes in these types of approaches, some characteristics of the data stream (e.g., performance measures, data distribution, properties of data, or an appropriate statistical function) are monitored over time. When the parameters switch during the monitoring process, the algorithm should be able to adapt the model to these changes.

We now briefly review some *other* schemes used for learning in non-stationary environments. The review here will not be exhaustive because the methods explained can be considered to be the basis for other modified approaches.

FLORA Widmer and Kubat [6], presented the FLORA family of algorithms as one of the first supervised incremental learning systems for a data stream. The initial FLORA algorithm used a fixed-size sliding window scheme. At each time step, the elements in the training window were used to incrementally update the learning model. The updating of the model involved two processes: an incremental *learning* process that updated the concept description based on the new data, and an incremental *forgetting* process that discarded the out-of-date (or stale) data.

The initial FLORA system did not perform well on large and complex data domains. Thus, FLORA2 was developed to solve the problem of working with a fixed window size, by using a heuristic approach to adjust the window size dynamically. Further improvements of the FLORA were presented to deal with recurring concepts (FLORA3) and noisy data (FLORA4).

Statistical Process Control (SPC) The SPC was presented by Gama *et al.* [4] for change detection in the context of data streams. The principle motivating the detection of concept drift using the SPC is to trace the probability of the error rate for the streamed observations. While monitoring the errors, the SPC provides three possible states, namely, "in control", "out of control" and "warning" to define a state when a warning has to be given, and when levels of changes appear in the stream. When the error rate is lower than the first (lower) defined threshold, the system is said to be in an "in control" state, and the current model is updated considering the arriving data. When the error exceeds that threshold, the system enters the "warning" state. In the "warning" state, the system stores the corresponding time as the warning time, t_w, and buffers the incoming data that appears subsequent to t_w. In the "warning" mode, if the error rate drops below the lower threshold, the "warning" mode is canceled and the warning time is reset. However, in case of an increasing error rate that reaches the second threshold, a concept change is declared and the learning model is retrained from the buffered data that appeared after t_w.

ADWIN Bifet and Gavalda [2,3] proposed an adaptive sliding window scheme named ADWIN for change detection and for estimating statistics from the data stream. It was shown that the ADWIN algorithm outperforms the SPC approach and that it has the ability to provide rigorous guarantees on false positive and false negative rates. The initial version of ADWIN keeps a variable-length sliding window, W, of the most recent instances by considering the hypothesis that there is no change in the average value inside the window. To achieve this, the distributions of the sub-windows of the W window are compared using the Hoeffding bound, and whenever there is a significant difference, the algorithm removes all instances of the older sub-windows and only keeps the new concepts for the next step. Thus, a change is reliably detected whenever the window shrinks, and the average over the existing window can be considered as an estimate of the current average in the data stream.

2.2 Stochastic Learning Weak Estimator (SLWE)

Using the principles of stochastic learning, Oommen and Reuda [5] proposed a strategy to solve the problem of estimating the parameters of a binomial or multinomial distribution efficiently in non-stationary environments. This method is referred to as the SLWE, where the convergence of the estimate is "weak", i.e., with respect to the first and second moments. Unlike the traditional MLE and the Bayesian estimators, which demonstrate strong convergence, the SLWE converges fairly quickly to the true value, and it is able to just as quickly "un-learn" the learning model trained from the historical data in order to adapt to the new data.

The SLWE is an estimator method that estimates the parameters of a binomial/multinomial distribution when the underlying distribution is non-stationary. In non-stationary environments, the SLWE updates the estimate of the distribution's probabilities at each time-instant based on the new observations. The updating, however, is achieved by a *multiplicative* rule. To formally introduce the SLWE, let X be a random variable of a multinomial[3] distribution, which can take the values from the set $\{`1', \ldots, `r'\}$ with the probability of S, where $S = [s_1, \ldots, s_r]^T$ and $\sum_{i=1}^{r} s_i = 1$. In the other words: $X = `i'$ with probability s_i.

Consider $x(n)$ as a concrete realization of X at time 'n'. In order to estimate the vector S, the SLWE maintains a running estimate $P(n) = [p_1(n), p_2(n), \ldots, p_r(n)]^T$ of vector S, where $p_i(n)$ is the estimation of s_i at time 'n', for $i = 1, \ldots, r$. The value of $p_i(n)$ is updated with respect to the coming data at each time instance, where Eqs. (1) and (2) show the updating rules:

$$p_i(n+1) \leftarrow p_i + (1-\lambda)\sum_{j \neq i} p_j \quad \text{when } x(n) = i \tag{1}$$

$$\leftarrow \lambda p_i \quad \text{when } x(n) \neq i. \tag{2}$$

Similar to the binomial case, the authors of [5] explicitly derived the dependence of $\mathrm{E}[P(n+1)]$ on $\mathrm{E}[P(n)]$, demonstrating the ergodic nature of the Markov matrix. The paper[4] also derived two explicit results concerning the convergence of the expected vector $P(.)$ to S, and the rate of convergence based on the learning parameter, λ.

Theorem 1. *Consider $P(n)$, the estimate of the multinomial distribution S at time 'n', which is obtained by Eqs. (1) and (2). Then, $E[P(\infty)] = S$.* □

[3] The case of estimating binomial distributions is a particular case of multinomial distributions where $r = 2$.

[4] The proofs of the theorems are omitted in the interest of brevity.

Theorem 2. *Consider $P(n)$, the estimate of the multinomial distribution S at time 'n', which is obtained by Eqs. (1) and (2). The expected value of P at time 'n + 1' is related to the expectation of $P(n)$ as $E[P(n+1)] = \mathbf{M}^T E[P(n)]$, where $\mathbf{M}$ is a Markov matrix. Further, every off-diagonal term of the stochastic matrix, $\mathbf{M}$, has the* same *multiplicative factor, $(1-\lambda)$, and the final solution of this vector* difference equation *is independent of λ.* □

Theorem 3. *Consider $P(n)$, the estimate of the multinomial distribution S at time 'n', which is obtained by Eqs. (1) and (2). Then, all the non-unity eigenvalues of $\mathbf{M}$ are exactly λ, and therefore the convergence rate of P is fully determined by λ.* □

Theoretically, since the derived results are asymptotic, they are valid only as $n \to \infty$. However, in practice, by choosing λ from the interval $[0.9, 0.99]$, the convergence happens after a relatively small value of 'n'. Indeed, if λ is as "small" as 0.9, the variation from the asymptotic value will be in the order of 10^{-50} after 50 iterations. In other words, the SLWE will provide good results even if the distribution parameters change after 50 steps. The experimental results in [5] demonstrated a good performance achieved by using the SLWE in dynamic environments.

3 Online Classification Using SLWE

In traditional ML learning and particularly supervised learning, the training phase is performed in an *offline* manner, i.e., the training set is used to learn the stochastic properties of each class. Subsequently, the learned model is deployed and used to classify unlabeled data instances that appeared in the form of data streams.

In many real life applications, it is not possible to analyze the stochastic model of the classes in an *offline* manner because of their dynamic natures. In fact, *offline* classifiers assume that the entire set of training samples can be accessed. However, in many real life applications, the entire training set is not available either because it arrives gradually or because it is not feasible to store it so as to infer the model of each class. Consequently, one is forced to constantly make the classifier update the learning model using the newly-arriving training samples.

We present a novel *online* classifier scheme, that is able to update the learned model using a single instance at a time. Our goal is to predict the source of the arriving instances as accurately as possible, with the added complexity that the testing patterns can subsequently be considered as training patterns. To achieve this, we first define the general structure of the *Online* classifier, and then provide some experimental results on synthetic multinomial datasets in the next section.

Online classifiers deal with data streams, in which the labeled and unlabeled samples are mixed. Therefore, the training, testing and deploying phases of the *online* classifiers are interleaved as they are applied to these types of data streams. This fascinating avenue is our domain, and we have investigated the performance of SLWE-based classifiers to this new scheme.

Devising a classifier that deals with the data streams generated from non-stationary sources poses new challenges, since the probability distribution of each class might change even as new instances arrive. An important characteristic of our model for *online* learning is that the actual source of the data is discovered shortly after the prediction is made, which can then be used to update the learned model. In other words, our *online* algorithm includes three steps, which are described in Algorithm 1. First, the algorithm receives a data element. Using it and the currently-learned model, the classifier predicts the source of that element. Finally, the algorithm receives the true class of the data, which is then used to update and refine the classification model.

In order to perform the *online* classification of the instances, we need to obtain the *a posteriori* probability of each class. Analogous to the previous classification models, we assign a label to the new unlabeled data element by comparing the obtained *a posteriori* probabilities and the estimated probability from the unlabeled test stream. Finally, after receiving the true label of the instance, the *a posteriori* probabilities are updated using the algorithm explained in Eqs. (1) and (2).

In this classification model the training phase and the testing phase were performed simultaneously, and so the problem can be described as follows. We are given the stream of unlabeled samples generated from different sources arriving in the form of a (Periodically Switching Environment) PSE, in which, after every T time instances, the data distribution and the source of the data might change. In this case, in addition to the switching of the source of the data elements, the probability distribution of each source also possibly changes at random time instances. The aim of the classification is to predict the source of the elements arriving at each time step by using the information in the detected data distribution, and also the information of current model of each class. In the *online* classification model, shortly after the prediction is made, the actual class label of the instance is discovered, which can be utilized to update the classification model to be used by the SLWE updating algorithm.

The process is formalized in Algorithm 1.

4 Experimental Results

In this section, we present the results of this classifier on synthetic data. To assess the efficiency of the SLWE-based *online* classifier, we applied it for multinomial randomly generated data streams. Our results demonstrate the applicability of our method on synthetic datasets, and proves the advantages of the introduced scheme. We also classified and compared the data streams' elements by following the traditional MLE with a sliding window, whose size is also selected randomly.

Algorithm 1 Online Classification Algorithm

$X \leftarrow$ data stream for classification
$\hat{S} \leftarrow$ initialize *posterior* probabilities for each class
while there exists an instance $x \in X$ **do**
  Step 1. Receiving data:
  The model receives the unlabeled sample
  for all dimensions d of x **do**
    $p_i(n) \leftarrow$ Estimate the probability p_i using the SLWE
  end for
  Step 2. Prediction:
  $P(n) \leftarrow \{p_1(n), p_2(n), \ldots, p_d(n)\}$
  $\hat{\omega} \leftarrow \arg_i \min KL(\hat{S}_i || P(n))$
  Step 3. Updating the model:
  After some delay, t_d, the true category of the instance x is received
  $\omega \leftarrow$ true class of x
  Update *posterior* probabilities $\hat{S}$ using ω and the SLWE
end while

4.1 Multinomial Data Stream

In this section, we report the results for simulations performed for *multinomial* data streams with *two* different non-stationary categories. Here, the classification problem was defined as follows. We are given a stream of unlabeled multinomially distributed random d-dimensional vectors, which take on the values from the set $\{1, \ldots, r\}$, and which are generated from *two* different periodically switching sources (classes), say, S_1 and S_2. Each class was characterized with probability values, S_{i1}, S_{i2}, which demonstrate the probability of the value 'i', where $i \in \{1, \ldots, r\}$.

The multinomial data stream classification started with the estimation of the *a priori* probability of each possible value of 'i', in all the 'd' dimensions, for each class 'j' from the available labeled instances, which we refer to as $\hat{S}_{ij}$. To assign a label to the newly arriving unlabeled element, the SLWE estimated the probabilities of each possible value 'i', in all the 'd' dimensions, from the unlabeled instances, which we refer to as $P_i(n)$. Thereafter, these probabilities were used to predict the class label that a new instance belonged to class 'j', with the probability vector of the $\hat{S}_j = \{\hat{S}_{1j}, \hat{S}_{2j}, \ldots, \hat{S}_r\}$, that had the minimum distance to the estimated probability of $P = \{P_1(n), P_2(n), \ldots, P_r(n)\}$, was chosen as the label of the observed element. The distances between the learned SLWE probabilities, $P_i(n)$, and the SLWE estimation during training, $\hat{S}_{ij}$ were again computed using the KL divergence measure, using Eq. (3).

$$KL(U||V) = \sum_i u_i \log_2 \frac{u_i}{v_i}. \tag{3}$$

Thereafter, after some delay, t_d, at time $n + t_d$ the algorithm received the true class of the n^{th} instance and used it to refine and update the true class probabilities. The true value of the category for the n^{th} instance was read and added to the previously trained model by updating the probability of the corresponding class based on the updating algorithm in Eqs. (1) and (2).

The classification procedure explained above was performed on multinomial data streams generated from two different classes where the probability of the distributions of each class switched four times. For the results which we report, each element of the data stream could take any of the four different values, namely 1, 2, 3 or 4. The specific values of S_{i1} and S_{i2}, were changed and set to random values four times at random time instances, which were assumed to be unknown to the classifiers. The results are shown in Table 1, and again, the uniform superiority of the SLWE over the MLEW is noticeable. For example, when T=100, the MLEW-based classifier yielded an accuracy of only 0.7443, but the corresponding accuracy of the SLWE-based classifier was 0.8012. We also notice that the results of the classification in periodic environments with a varying T chosen randomly from $[50, 150]$ were also similar to the fixed $T = 100$ case, as the classifier achieved the accuracy of 0.8092 and 0.8012 in the first and second environments, respectively. The results also show that the SLWE-based algorithm handles the concept drift and provides satisfactory performance.

T	MLEW	SLWE
50	0.6786	0.7582
100	0.7443	0.8012
150	0.7476	0.8153
200	0.7595	0.8197
250	0.7514	0.8282
300	0.7509	0.8367
350	0.7587	0.8322
400	0.7598	0.8354
450	0.7635	0.8344
500	0.7574	0.8387
Random $T \in (50, 150)$	0.7523	0.8092

Table 1. The ensemble results for 100 simulations obtained from testing multinomial classifiers which used the SLWE (with $\lambda = 0.9$) and the MLEW for classifying one-dimensional data streams generated by two non-stationary different sources.

The experiment explained above was repeated on different 2-class multinomial datasets with different dimensionalities. These sets were generated randomly based on random vectors with different random distribution probabilities involving 2, 3 and 4 dimensions and each element could take on four different values. The results obtained are shown in Tables 3-2, from which we see that classification using the SLWE was uniformly superior to classification using the MLEW. For example, for the 2-dimensional data, when $T = 250$, the MLE-based classifier resulted in the accuracy of 0.7544 and the SLWE achieved significantly better results with the accuracy of 0.9706. Here the accuracy of the classifier,

T	MLEW	SLWE
50	0.6906	0.8847
100	0.7535	0.9402
150	0.7557	0.9592
200	0.7532	0.9669
250	0.7550	0.9730
300	0.7531	0.9760
350	0.7522	0.9785
400	0.7460	0.9818
450	0.7572	0.9817
500	0.7562	0.9839
Random $T \in (50, 150)$	0.7526	0.9512

Table 2. The ensemble results for 100 simulations obtained from testing multinomial classifiers which used the SLWE (with $\lambda = 0.9$) and the MLEW for classifying 4-dimensional data streams generated by two non-stationary different sources.

increased with the dimensionality of the datasets as the classifiers could process the data more efficiently. For example, in the case of $T = 150$ the SLWE-based *online* classifier resulted in the average accuracy of 0.9181 over several different two-dimensional datasets, while with more useful information in 4-dimensional datasets, it yielded better results with the accuracy of 0.9569.

T	MLEW	SLWE
50	0.6913	0.8560
100	0.7402	0.9082
150	0.7463	0.9333
200	0.7480	0.9393
250	0.7478	0.9430
300	0.7397	0.9474
350	0.7486	0.9463
400	0.7525	0.9535
450	0.7554	0.9505
500	0.7518	0.9559
Random $T \in (50, 150)$	0.7436	0.9241

Table 3. The ensemble results for 100 simulations obtained from testing multinomial classifiers which used the SLWE (with $\lambda = 0.9$) and the MLEW for classifying 2-dimensional data streams generated by two non-stationary different sources.

5 Conclusion

In this paper we have considered the problem of classification when these phases of training and testing are not so clearly well-defined, i.e., where the testing patterns can subsequently be considered as training patterns. This paradigm of classification is further complicated because we have assumed that the class-conditional distributions of the classes are time-varying or non-stationary. Here, we consider the scenario when the patterns arrive sequentially in the form of a data stream with potentially time-varying probabilities that change over time for each class. The proposed *online* classification algorithm was used to perform the training and the testing simultaneously in three phases. In the first phase, the algorithm received a new unlabeled instance. After this, the scheme assigned a label to it based on the distributions' estimated probabilities using the SLWE. Finally, after a few time instances, the algorithm received the actual class of the instance and used it to update the training model by invoking the SLWE updating algorithm. In this way the training and testing phases are almost intertwined. Thereafter, the classification model was adjusted to the new available instances in an *online* manner.

References

1. A. Bifet. *Adaptive Learning and Mining for Data Streams and Frequent Patterns*. PhD thesis, Departament de Llenguatges i Sistemes Informatics, Universitat Politcnica de Catalunya, Barcelona Area, Spain, 2009.
2. A. Bifet and R. Gavalda. Kalman Filters and Adaptive Windows for Learning in Data Streams. In L. Todorovski and N. Lavrac, editors, *Proceedings of the 9th Discovery Science*, volume 4265, pages 29–40, 2006.
3. A. Bifet and R. Gavalda. Learning From Time-changing Data With Adaptive Windowing. In *Proceedings SIAM International Conference on Data Mining*, volume 8, pages 443–448, 2007.
4. J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with Drift Detection. In A. Bazzan and S. Labidi, editors, *Advances in Artificial Intelligence SBIA 2004*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer Berlin Heidelberg, 2004.
5. B. J. Oommen and L. Rueda. Stochastic Learning-based Weak Estimation of Multinomial Random Variables and Its Applications to Pattern Recognition in Non-stationary Environments. *Pattern Recognition*, 39(3):328–341, 2006.
6. G. Widmer and M. Kubat. Learning in the Presence of Concept Drift and Hidden Contexts. In *Machine Learning*, volume 23, pages 69–101, 1996.