A Martingale Framework for Concept Change Detection in Time-Varying Data Streams

Shen-Shyang Ho SHO@GMU.EDU
Department of Computer Science, George Mason University, 4400 University Drive, Fairfax, VA 22030 USA

Abstract

In a data streaming setting, data points are observed one by one. The concepts to be learned from the data points may change infinitely often as the data is streaming. In this paper, we extend the idea of testing exchangeability online (Vovk et al., 2003) to a martingale framework to detect concept changes in time-varying data streams. Two martingale tests are developed to detect concept changes using: (i) martingale values, a direct consequence of the Doob's Maximal Inequality, and (ii) the martingale difference, justified using the Hoeffding-Azuma Inequality. Under some assumptions, the second test theoretically has a lower probability than the first test of rejecting the null hypothesis, "no concept change in the data stream", when it is in fact correct. Experiments show that both martingale tests are effective in detecting concept changes in time-varying data streams simulated using two synthetic data sets and three benchmark data sets.

1. Introduction

A challenge in mining data streams is the detection of changes in the data-generating process. Recent research includes profiling and visualizing changes in data streams using velocity density estimation (Aggarwal, 2003), but reaches no conclusion on whether a change takes place. Fan et al. (2004) proposed change detection (active mining) based on error estimation of a model of the new data stream without knowing the true class labels. Kifer et al. (2004) proposed a change-detection method with statistical guarantees of the reliability of detected changes, however the method is impractical for high dimensional data streams. Besides detecting changes, change-adaptive methods for the so-called concept drift problem, based on a sliding window (instance selection) (Klinkenberg & Joachims, 2000; Widmer & Kubat, 1996), instance weighting (Klinkenberg, 2004), and ensemble learning (Chu et al., 2004; Kolter & Maloof, 2003; Wang et al., 2003), are also suggested.

The problem of detecting changes in sequential data was first studied by statisticians and mathematicians. In the online setting, data are observed one by one from a source. The disruption of "stochastic homogeneity" of the data might signal a change in the data-generating process, which would require decision-making to avoid possible losses. This problem is generally known as a "change-point detection". Methods of change detection first appeared in the Forties based on Wald's sequential analysis (Wald, 1947), and later, Page introduced the cumulative sum method (Page, 1957). These methods are parametric and work only for low-dimensional data streams.

An effective change-detecting algorithm requires that (i) the mean (or median) delay time between a true change point and its detection be minimal, (ii) the number of miss detections be minimal, and (iii) data streams be handled efficiently.

In this paper, we propose a martingale framework that effectively and efficiently detects concept changes in time-varying data streams. In this framework, when a new data point is observed, hypothesis testing using a martingale takes place to decide whether change occurs. Two tests are shown to be effective using this framework: testing exchangeability using (i) a martingale value (Vovk et al., 2003) and (ii) the martingale difference. The first test is a direct consequence of the Doob's Maximal Inequality. We provide detailed justification for the second test using the Hoeffding-Azuma Inequality. Under some assumptions, this second test has a much lower probability than the first test of rejecting the null hypothesis, "no concept change in the

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

data stream", when it is in fact correct. The efficiency of the martingale tests depends on the speed of the classifier used for the construction of the martingale.

Our martingale approach is an efficient, one-pass incremental algorithm that (i) does not require a sliding window on the data stream, (ii) does not require monitoring the performance of the base classifier as data points are streaming, and (iii) works well for high dimensional, multi-class data stream.

In Section 2, we review the concept of martingale and exchangeability. In Section 3, we describe and justify two tests using martingales. In Section 4, we examine both tests in time-varying data streams simulated using two synthetic data sets and three benchmark data sets.

2. Martingale and Exchangeability

Let $\{Z_i : 1 \leq i < \infty\}$ be a sequence of random variables. A finite sequence of random variables $Z_1, \cdots, Z_n$ is *exchangeable* if the joint distribution $p(Z_1, \cdots, Z_n)$ is invariant under any permutation of the indices of the random variables. A *martingale* is a sequence of random variables $\{M_i : 0 \leq i < \infty\}$ such that M_n is a measurable function of $Z_1, \cdots, Z_n$ for all $n = 0, 1, \cdots$ (in particular, M_0 is a constant value) and the conditional expectation of M_{n+1} given $M_0, \cdots, M_n$ is equal to M_n, i.e.

$$E(M_{n+1}|M_1, \cdots, M_n) = M_n \tag{1}$$

Vovk et al. (2003) introduced the idea of testing exchangeability online using the martingale. After observing a new data point, a learner outputs a positive martingale value reflecting the strength of evidence found against the null hypothesis of data exchangeability. Consider a set of labeled examples $Z = \{z_1, \cdots, z_{n-1}\} = \{(x_1, y_1), \cdots, (x_{n-1}, y_{n-1})\}$ where x_i is an object and $y_i \in \{-1, 1\}$, its corresponding label, for $i = 1, 2, \cdots, n-1$. Assuming that a new labeled example, z_n, is observed, testing exchangeability for the sequence of examples $z_1, z_2, \cdots, z_n$ consists of two main steps (Vovk et al., 2003):

A. Extract a p-value p_n for the set $Z \cup \{z_n\}$ from the strangeness measure deduced from a classifier

The randomized p-value of the set $Z \cup \{z_n\}$ is define as

$$V(Z \cup \{z_n\}, \theta_n) = \frac{\#\{i : \alpha_i > \alpha_n\} + \theta_n \#\{i : \alpha_i = \alpha_n\}}{n} \tag{2}$$

where α_i is the strangeness measure for z_i, $i = 1, 2, \cdots, n$ and θ_n is randomly chosen from $[0, 1]$. The strangeness measure is a way of scoring how a data point is different from the rest. Each data point z_i is assigned a strangeness value α_i based on the classifier used (e.g. support vector machine (SVM), nearest neighbor rule, and decision tree). In our work, the SVM is used to compute the strangeness measure, which can be either the Lagrange multipliers or the distances from the hyperplane for the examples in $Z \cup \{z_n\}$.

The p-values $p_1, p_2, \cdots$ output by the randomized p-value function V are distributed uniformly in $[0, 1]$, provided that the input examples $z_1, z_2, \cdots$ are generated by an exchangeable probability distribution in the input space (Vovk et al., 2003). This property of output p-values no longer holds when the exchangeability condition is not satisfied (see Section 3).

B. Construct the randomized power martingale

A family of martingales, indexed by $\epsilon \in [0, 1]$, and referred to as the *randomized power martingale*, is defined as

$$M_n^{(\epsilon)} = \prod_{i=1}^{n} \left(\epsilon p_i^{\epsilon - 1}\right) \tag{3}$$

where the p_is are the p-values output by the randomized p-value function V, with the initial martingale $M_0^{(\epsilon)} = 1$. We note that $M_n^{(\epsilon)} = \epsilon p_n^{\epsilon-1} M_{n-1}^{(\epsilon)}$. Hence, it is not necessary to stored the previous p-values. In our experiments, we use $\epsilon = 0.92$, which is within the desirable range where the martingale value is more sensitive to a violation of the exchangeability condition (Vovk et al., 2003).

When $\theta_n = 1$, the p-value function V is deterministic, the martingale constructed is also deterministic. We use this deterministic martingale in our justification for the second test in Section 3.3.

3. Testing for Change Detection

Intuitively, we assume that a sequence of data points with a concept change consists of concatenating two data segments, S_1 and S_2, such that the concepts of S_1 and S_2 are C_1 and C_2 respectively and $C_1 \neq C_2$. Switching a data point z_i from S_2 to a position in S_1 will make the data point stands out in S_1. The exchangeability condition is, therefore, violated. Exchangeability is a necessary condition for a conceptually stable data stream. The absence of exchangeability would suggest concept changes.

When a concept change occurs, the p-values output from the randomized p-value function (2) become skewed and the p-value distribution is no longer uniform. By the Kolmogorov-Smirnov Test (KS-Test) [1], the p-values are shown not to be distributed uniformly after the concept changes. The null hypothesis "the p-values output by (2) are uniformly distributed" is rejected at significance level $\alpha = 0.05$, after sufficient number of data points are observed (see the example in Figure 1). The skewed p-value distribution plays an important role in our martingale test for change detection as small p-values inflate the martingale values. We note that an immediate detection of a true change is practically impossible. Hence, a short delay time between a change and its detection is highly desirable.

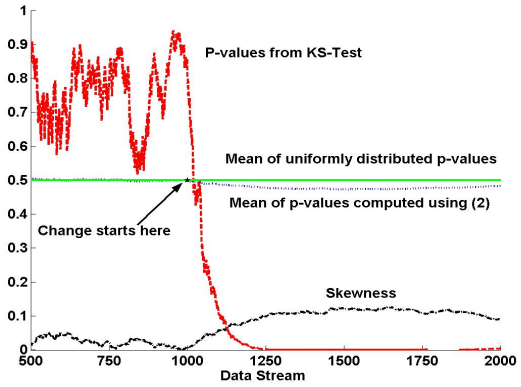

Figure 1. The 10-dimensional data points simulated using the normally distributed clusters data generator (see Section 4.1.2) are observed one by one from the $1st$ to the $2000th$ data point with concept change starting at the $1001th$ data point. The reader should not confuse the p-values from the KS-Test and the p-values computed from (2).

3.1. Martingale Framework for Detecting Changes

In the martingale framework, when a new data point is observed, hypothesis testing takes place to decide whether a concept change occurs in the data stream. The decision is based on whether the exchangeability condition is violated, which, in turn, is based on the martingale value.

Two hypothesis tests based on the martingale (3) are proposed based on the Doob's Maximal Inequality and the Hoeffding-Azuma Inequality respectively. Consider the simple null hypothesis H_0 : "no concept change in the data stream" against the alternative H_1 : "concept change occurs in the data stream". The test continues to operate as long as

Martingale Test 1 (MT1):

$$0 < M_n^{(\epsilon)} < \lambda \tag{4}$$

where λ is a positive number. One rejects the null hypothesis when $M_n^{(\epsilon)} \geq \lambda$.

OR

Martingale Test 2 (MT2):

$$0 < |M_n^{(\epsilon)} - M_{n-1}^{(\epsilon)}| < t \tag{5}$$

where t is a positive number. One rejects the null hypothesis when $|M_n^{(\epsilon)} - M_{n-1}^{(\epsilon)}| \geq t$.

3.2. Justification for Martingale Test 1 (MT1)

Assuming that $\{M_k : 0 \leq k < \infty\}$ is a nonnegative martingale, the Doob's Maximal Inequality (Steele, 2001) states that for any $\lambda > 0$ and $0 \leq n < \infty$,

$$\lambda P\left(\max_{k \leq n} M_k \geq \lambda\right) \leq E(M_n) \tag{6}$$

Hence, if $E(M_n) = E(M_1) = 1$, then

$$P\left(\max_{k \leq n} M_k \geq \lambda\right) \leq \frac{1}{\lambda} \tag{7}$$

This inequality means that it is unlikely for any M_k to have a high value. One rejects the null hypothesis when the martingale value is greater than λ. But there is a risk of announcing a change detection when there is no change. The amount of risk one is willing to take will determine what λ value to use.

3.3. Justification for Martingale Test 2 (MT2)

Theorem 1 (Hoeffding-Azuma Inequality) *Let $c_1, \cdots, c_m$ be constants and let $Y_1, \cdots, Y_m$ be a martingale difference sequence with $|Y_k| \leq c_k$, for each k. Then for any $t \geq 0$,*

$$P\left(\left|\sum_{k=1}^{m} Y_k\right| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2\sum c_k^2}\right) \tag{8}$$

To use this probability bound to justify our hypothesis test, we need the martingale difference to be bounded, i.e. $|Y_i| = |M_i - M_{i-1}| \leq K$ such that M_i and M_{i-1}

[1] Kifer et al. (2004) proposed using Kolmogorov-Smirnov Test (KS-Test) for detecting changes using two sliding windows and a discrepancy measure which was tested only on 1D data stream.

are two arbitrary consecutive martingale values and $K \in \mathbf{R}^+$. This bounded difference condition states that the process does not make big jumps. Moreover, it is unlikely that the process wanders far from its initial point. Hence, before using (8) to construct the probability upper bound to justify MT2, we need to show that the difference between two consecutive power martingale values is bounded for some fixed ϵ. As mentioned earlier in Section 2, we use the deterministic power martingale in our proof. We set $\theta_n = 1$, for $n \in \mathbf{Z}^+$ in the randomized p-value function (2). An output p-values p_n is a multiple of $\frac{1}{n}$ between $\frac{1}{n}$ and 1.

The martingale difference is

$$d_n = \prod_{i=1}^{n-1} \left(\epsilon p_i^{\epsilon-1}\right) \left(\epsilon p_n^{\epsilon-1} - 1\right) \quad (9)$$

For $p_n = \frac{u}{n}, 1 \leq u \leq n$, if

$$p_n < \exp\left(\frac{\log \epsilon}{1-\epsilon}\right) \quad (10)$$

we have $d_n > 0$; otherwise $d_n < 0$. The most negative d_n occurs when $p_n = 1$ and the most positive d_n occurs when $p_n = \frac{1}{n}$. This most positive value is higher than the most negative value and, therefore, $p_n = \frac{1}{n}$ will be used in the bounded difference condition.

When $m = 1$, the Hoeffding-Azuma Inequality (8) becomes

$$P(|Y_1| \geq t) \leq 2\exp\left(-\frac{t^2}{2c_1^2}\right) \quad (11)$$

and hence, for any n,

$$P(|M_n^{(\epsilon)} - M_{n-1}^{(\epsilon)}| \geq t) \leq 2\exp\left(-\frac{t^2}{2\left(\epsilon\left(\frac{1}{n}\right)^{\epsilon-1} - 1\right)^2 \left(M_{n-1}^{(\epsilon)}\right)^2}\right) \quad (12)$$

Assuming that every testing step is a new testing step based on a new martingale sequence, we set the previous martingale value $M_{n-1}^{(\epsilon)} = M_0^{(\epsilon)} = 1$ on the right-hand side of the inequality (12). Hence, we have

$$P(|M_n^{(\epsilon)} - M_{n-1}^{(\epsilon)}| \geq t) \leq 2\exp\left(-\frac{t^2}{2\left(\epsilon\left(\frac{1}{n}\right)^{\epsilon-1} - 1\right)^2}\right) \quad (13)$$

If we only consider $M_n^{(\epsilon)} > M_{n-1}^{(\epsilon)}$, the upper bound is less than the right-hand side of (13). Like MT1, one selects t according to the risk one is willing to take.

However, the probability upper bound (13) for MT2 also depends on n, the number of data points used. As n increases, the upper bound also increases. The probability of rejecting the null hypothesis when it is correct increases. To maintain a much better probability bound for larger n, t can be increased (see Figure 2) at the expense of a higher delay time (see Section 4.2).

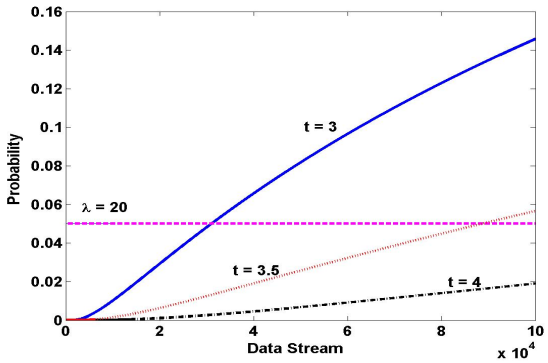

Figure 2. Comparison of the upper bound of the probability of the martingale difference for some t values and $\epsilon = 0.92$, and the fixed probability upper bound for the martingale value when $\lambda = 20$ on a data stream consisting of 100000 data points. To have an upper bound for MT1 that matches the upper bound for a particular t value (say, $3 - 4$) for a small n (< 5000), λ has to be very large.

From Figure 2, one observes that if a sliding window is not used for MT2, the classifier used to extract the p-value should dynamically remove old data points from its memory when the upper bound exceeds a predefined value. In our experiments, we use a "pseudo-adaptive" approach for the window size . Our window starts from the previous detected point and increases in size until the next change point is detected, as long as the probability upper bound does not exceed a fixed value we specified for a particular chosen t. Otherwise, we remove the earliest data point from the memory. We note that in our experiments the interval between two true change points is small ($< 2,000$) and the performance of MT2 is not affected by the upper bound (13) as n increases.

4. Experiments

Experiments are performed to show that the two tests are effective in detecting concept changes in time-varying data streams simulated using two synthetic data sets and three benchmark data sets. The five different simulated data streams are described in Section 4.1.

We examine the performance of both tests based on the retrieval performance indicators, recall and precision, and the delay time for change detections for various λ and t values on two time-varying data streams simulated using the two synthetic data sets. The retrieval performance indicators are defined in our context as:

$$\text{Precision} = \frac{\text{Number of Correct Detections}}{\text{Number of Detections}}$$

$$\text{Recall} = \frac{\text{Number of Correct Detections}}{\text{Number of True Changes}}$$

Precision is the probability that a detection is actually correct, i.e. detecting a true change. Recall is the probability that a change detection system recognizes a true change.

The delay time for a detected change is the number of time units from the true change point to the detected change point, if any. We also show that both martingale tests are feasible on high dimensional (i) numerical, (ii) categorical, and (iii) multi-class data streams.

In the experiments, a fast adiabatic incremental SVM (Cauwenberghs & Poggio, 2000), using the Gaussian kernel and $C = 10$, is used to deduce the strangeness measure for the data points. A necessary condition for both tests to work well is that the classifier must have a reasonable classification accuracy. At a fixed ϵ, the performance of the two tests depend on the λ or t. Experimental results are reported in Section 4.2.

4.1. Simulated Data Stream Descriptions

In this subsection, we describe how the five data streams with concept changes are simulated by (i) using rotating hyperplane (Hulten et al., 2001) (Section 4.1.1), (ii) using the normally distributed clusters data generator (NDC) (Musicant, 1998) (Section 4.1.2), (iii) combining ringnorm and twonorm data sets (Breiman, 1996) (Section 4.1.3), (iv) modifying UCI nursery data set (Blake & Merz, 1998) (Section 4.1.4), and (v) modifying the USPS handwritten digits data set (LeCun et al., 1989) (Section 4.1.5).

4.1.1. Simulated Data Stream using Rotating Hyperplane

A data stream is simulated by using a rotating hyperplane to generate a sequence of 100,000 data points consisting of changes occurring at points $(1,000 \times i)+1$, for $i = 1, 2, \cdots, 99$. First we randomly generate 1,000 data points with each component's values in the closed interval $[-1, 1]$. These data points are labeled positive and negative based on the following equation:

$$\sum_{i=1}^{m} w_i x_i = \begin{cases} < c & : \text{ negative} \\ \geq c & : \text{ positive} \end{cases} \quad (14)$$

where c is an arbitrary fixed constant, x_i is the component of a data point, x, and the fixed components, w_i, of a weight vector are randomly generated between -1 and 1. Similarly, the next 1,000 random data points are labeled using (14) with a new randomly generated fixed weight vector. This process continues until we get a data stream consisting of 100 segments of 1,000 data points each. Noise is added by randomly switching the class labels of $p\%$ of the data points. In our experiment, $p = 5$ and $m = 10$.

4.1.2. Simulated Data Stream using the normally distributed clusters data generator (NDC)

Linearly non-separable binary-class data streams of 100,000 data points consisting of changes occurring at points $(1,000 \times i) + 1$, for $i = 1, 2, \cdots, 99$ is simulated using the NDC in $\mathbf{R}^{10}$ with randomly generated cluster means and variances. The values for each dimension are scaled to range in $[-1, 1]$. The generating process for the data stream is similar to that used for the rotating hyperplane data stream described in Section 4.1.1.

4.1.3. Numerical High Dimensional Datasets: Ringnorm and Twonorm

We combined the *ringnorm (RN)* (two normal distribution, one within the other) and *twonorm (TN)* (two overlapping normal distribution) data sets to form a new binary-class data stream of 20 numerical attributes consisting of $14,800$ data points. The 7,400 data points from the RN are partitioned into 8 subsets with the first 7 subsets $(RN_i, i = 1, \cdots, 7)$ consisting of 1,000 data points each and RN_8 consisting of 400 data points. Similarly, the 7,400 data points from TN are also partitioned into 8 subsets with the first 7 subsets $(TN_i, i = 1, \cdots, 7)$ consisting of 1,000 data points each and the TN_8 consisting of 400 data points.

The new data stream is a sequence of data points arranged as follows: $\{RN_1; TN_1; \cdots RN_7; TN_7; RN_8; TN_8\}$ with 15 changes at data points $1000i + 1$ for $i = 1, \cdots, 14$, and 14,401.

4.1.4. Categorical High Dimensional Dataset: Nursery benchmark

We modified the nursery data set, which consists of 12,960 data points in 5 classes with 8 nominal attributes, to form a new binary-class data stream.

Segment	Digit 1	Digit 2	Digit 3	Total	Change Point
1	597 (0)	502 (1)	731 (2)	1830	1831
2	597 (0)	658 (3)	652 (4)	1907	3738
3	503 (1)	556 (5)	664 (6)	1723	5461
4	645 (7)	542 (8)	644 (9)	1831	-

Table 1. **Three-Digit Data Stream**: TR (D): TR is the number of data points and D is the true digit class of the data points.

First, we combined three classes (not recommended, recommended, and highly recommended) into a single class consisting of 4,650 data points labeled as negative examples. The set RN is formed by randomly selecting 4,000 out of the 4,650 data points. The "priority" class contains 4,266 data points that are labeled as positive examples. We randomly selected 4,000 out of the 4,266 data points to form the set PP. The "special priority" class, which contains 4,044 data points, is split into two subsets consisting of 2,000 data points each, a set (SPP) of positive examples, and a set (SPN) of negative examples. The other 44 data points are removed.

New subsets of data points are constructed as follows:

- Set A_i: 500 negative examples from RN and 500 positive examples from PP.
- Set B_i: 500 negative examples from SPN and 500 positive examples from PP.
- Set C_i: 500 negative examples from RN and 500 positive examples from SPP.

The data stream S is constructed as follows: $\{A_1; B_1; C_1; A_2; B_2; C_2; A_3; B_3; C_3; A_4; B_4; C_4\}$ consisting of 12,000 examples with 11 change points.

4.1.5. Multi-class High Dimensional Data: three-digit data stream from USPS handwritten digits data set.

The USPS handwritten digits data set, which consists of 10 classes of dimension 256 and includes 7,291 data points, is modified to form a data stream as follows. There are four different data segments. Each segment draws from a fixed set of three different digits in a random fashion. The three-digit sets change from one segment to the next. The composition of the data stream and ground truth for the change points are summarized in Table 1. We note that the change points do not appear at fixed intervals. The one-against-the-rest multi-class SVM is used to extract p-values.

For the three-digit data stream, three one-against-the-rest SVM are used. Hence, three martingale values are computed at each point to detect change (see Figure 7). When one of the martingale values is greater than λ (or t), change is detected.

4.2. Results

Figure 3 and 4 show the recall, precision, and delay time of the two martingale tests on the data streams simulated using the rotating hyperplane and NDC respectively. As can be seen from Figure 3 and 4 (first row), the recall is consistently greater than 0.95 on both simulated data streams for various λ and t values. Both tests recognize concept changes with high probability.

As λ or t increases, one observes that the precision increases. As λ increases from 4 to 100, the upper bound (7) becomes tighter, decreasing from 0.25 to 0.01, for MT1. This corresponds to the precision increasing from 0.82 to 1 (see Figure 3), decreasing the false alarm rate. On the other hand, as t increases from 1.5 to 5, precision increases from 0.88 to 1. The upper bound (13) for MT2 is consistently small as long as the data stream used for computing the martingale is "short" (e.g. at $n = 1,000$, when $t = 1.5$, the upper bound is 0.0868 and when $t = 5$, the upper bound is 1.44×10^{-15}). This is a plausible explanation for MT2 having a higher precision than MT1. A similar trend also appears in simulated data streams using the NDC (see Figure 4). To this end, it seems that for high recall and precision, a large λ or t should be used.

Figure 3 and 4 (second row) reveal, unsurprisingly, that a higher precision (using higher λ or t) comes at the expense of a higher mean (or median) delay time for both tests. The mean (or median) delay time for the two tests do not differ significantly. With a box-plot on the delay time, one can observe that the delay time distribution skews toward large values (i.e. small values are packed tightly together and large values stretch out and cover a wider range), independent of the λ or t value. The delay time is very likely to be less than the mean delay time.

In real applications, λ or t must be chosen to minimize losses (or cost) due to delay time, missed detections, and false alarms.

Figure 5, 6, and 7 show the feasibility of MT1 and MT2 on high dimensional (i) numerical (combining ringnorm and twonorm data sets), (ii) categorical (modified UCI nursery data set), and (iii) multi-class (modified USPS handwritten digit data set) data streams, respectively. From the figures, one observes that for MT2, when changes are detected, there are more variations in the martingale values. To this end, one sees

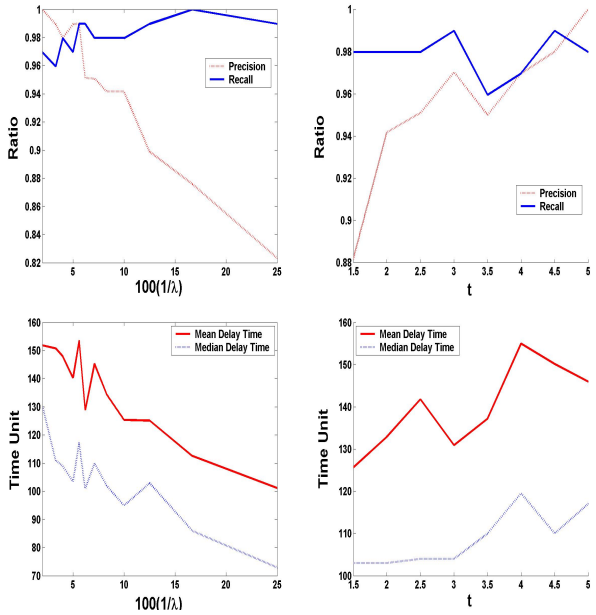

Figure 3. Simulated data streams using the rotating hyperplane. Left Column: MT1 (with λ^{-1} scaled by a factor of 100 for easier visualization of the probability); Right Column: MT2. First Row: Precision and Recall; Middle Row: Mean and Median Delay time for various λ and t values.

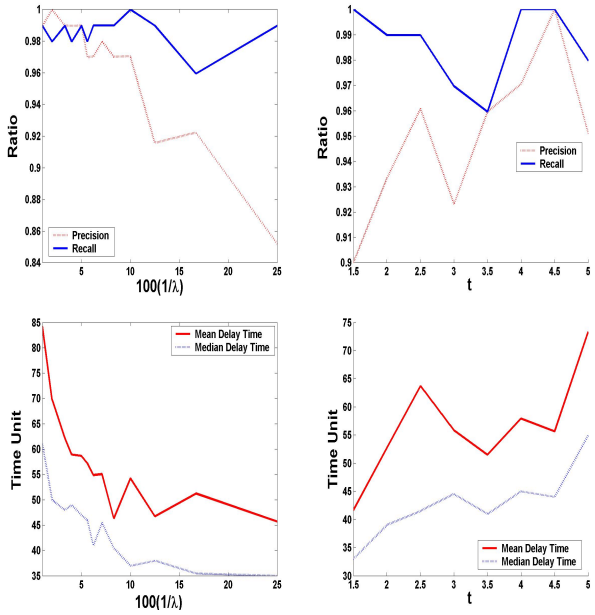

Figure 4. Simulated data streams using the NDC data generator. Left Column: MT1; Right Column: MT2. (Explanation: See Caption for Figure 3.)

that a change is detected by either of the tests when the martingale value deviates from its initial value, $M_0 = 1$.

5. Conclusion

In this paper, we describe a martingale framework for detecting concept changes in time-varying data streams based on the violation of exchangeability condition. Two tests using martingales to detect changes are used to demonstrate this framework. One test using the martingale value (MT1) for change detection is easily justified using the Doob's Maximal Inequality. The other test, based on the martingale difference (MT2), is justified using the Hoeffding-Azuma Inequality. Under some assumptions, MT2 theoretically has a much lower probability than MT1 of rejecting the null hypothesis "no concept change in the data stream" when it is in fact correct. Our experiments show that both martingale tests detect concept changes with high probability. Precision increases with the increase of λ or t values, but at the expense of a higher mean (or median) delay time. Experiments also show the effectiveness of the two tests for concept change detection on high-dimensional (i) numerical, (ii) categorical, and (iii) multi-class data streams.

Acknowledgments

The author thanks the reviewers for useful comments, Alex Gammerman and Vladimir Vovk for the manuscript of (Vovk et al., 2005) and useful discussions, and Harry Wechsler for guidance and discussions.

References

Aggarwal, C. C. (2003). A framework for change diagnosis of data streams. *Proc. ACM SIGMOD Int. Conf. on Management of Data* (pp. 575–586). ACM.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.

Breiman, L. (1996). *Bias, variance, and arcing classifiers* (Technical Report 460). Statistics Department, University of California.

Cauwenberghs, G., & Poggio, T. (2000). Incremental support vector machine learning. *Advances in Neural Information Processing Systems 13* (pp. 409–415). MIT Press.

Chu, F., Wang, Y., & Zaniolo, C. (2004). An adaptive learning approach for noisy data streams. *Proc. 4th IEEE Int. Conf. on Data Mining* (pp. 351–354). IEEE Computer Society.

Fan, W., Huang, Y.-A., Wang, H., & Yu, P. S. (2004). Active mining of data streams. *Proc. 4th SIAM Int. Conf. on Data Mining*. SIAM.

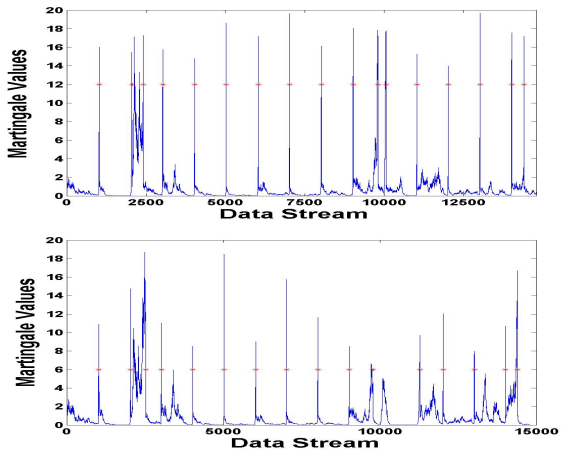

Figure 5. Simulated data streams using the Ringnorm and Twonorm data sets: The martingale values of the data stream. $*$ represent detected change points. Top Graph: MT1 ($\lambda = 20$), mean (median) delay time is 30.93 (26) with 2 false alarms. Bottom Graph: MT2 ($t = 3.5$), a miss detection at 10001. The mean (Median) delay time is 42.57 (24.5).

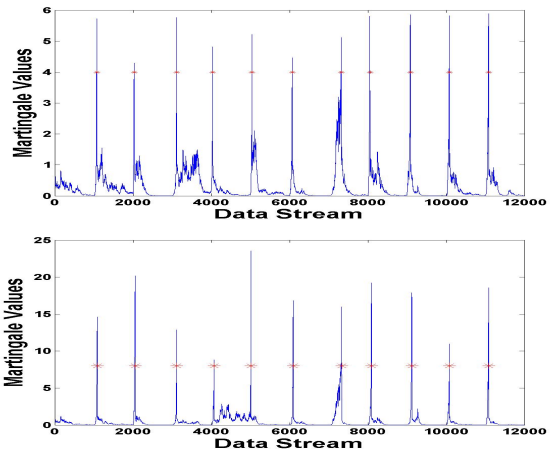

Figure 6. Simulated data streams using the UCI nursery dataset: The martingale values of the data stream. $*$ represent detected change points. Top Graph: MT1 ($\lambda = 6$), the mean (median) delay time is 91.27 (81). Bottom Graph: MT2 ($t = 3.5$), the mean (median) delay time is 107.64 (92).

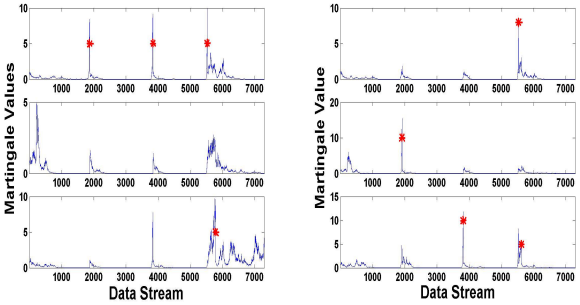

Figure 7. Simulated three-digit data stream using the USPS handwritten digit data set: The martingale values of the data stream. $*$ represent detected change points. Left Graph: MT1 ($\lambda = 10$), the delay time are 45, 99 and 62. There is one false alarm . Right Graph: MT2 ($t = 2.5$), the delay time are 88, 81 and 73. There is one false alarm.

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. *Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 97–106). ACM.

Kifer, D., Ben-David, S., & Gehrke, J. (2004). Detecting change in data streams. *Proc. 13th Int. Conf. on Very Large Data Bases* (pp. 180–191). Morgan Kaufmann.

Klinkenberg, R. (2004). Learning drifting concepts: examples selection vs example weighting. *Intelligent Data Analysis, Special Issue on Incremental Learning Systems capable of dealing with concept drift*, *8*, 281–300.

Klinkenberg, R., & Joachims, T. (2000). Detecting concept drift with support vector machines. *Proc. 17th Int. Conf. on Machine Learning* (pp. 487–494). Morgan Kaufmann.

Kolter, J. Z., & Maloof, M. A. (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. *ICDM* (pp. 123–130). IEEE Computer Society.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. J. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*, 541–551.

Musicant, D. R. (1998). *Normally distributed clustered datasets.* Computer Sciences Department, University of Wisconsin, Madison, http://www.cs.wisc.edu/dmi/svm/ndc/.

Page, E. S. (1957). On problem in which a change in a parameter occurs at an unknown point. *Biometrika*, *44*, 248–252.

Steele, M. (2001). *Stochastic calculus and financial applications.* Springer Verlag.

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world.* Springer.

Vovk, V., Nouretdinov, I., & Gammerman, A. (2003). Testing exchangeability on-line. *Proc. 20th Int. Conf. on Machine Learning* (pp. 768–775). AAAI Press.

Wald, A. (1947). *Sequential analysis.* Wiley, N. Y.

Wang, H., Fan, W., Yu, P. S., & Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. *Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 226–235). ACM.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, *23*, 69–101.