Detecting Changes in Rare Patterns from Data Streams

David Tse Jung Huang[1], Yun Sing Koh[1], Gillian Dobbie[1], and Russel Pears[2]

[1] Department of Computer Science, University of Auckland, New Zealand
{dtjh,ykoh,gill}@cs.auckland.ac.nz
[2] School of Computing and Mathematical Sciences, AUT University, New Zealand
rpears@aut.ac.nz

Abstract. Current drift detection techniques in data streams focus on finding changes in streams with labeled data intended for supervised machine learning methods. Up to now there has been no research that considers drift detection on item based data streams with unlabeled data intended for unsupervised association rule mining. In this paper we address and discuss the current issues in performing drift detection of rare patterns in data streams and present a working approach that enables the detection of rare pattern changes. We propose a novel measure, called the M measure, that facilitates pattern change detection and through our experiments we show that this measure can be used to detect changes in rare patterns in data streams efficiently and accurately.

Keywords: Data Stream, Drift Detection, Rare Pattern.

1 Introduction

Mining data streams for knowledge discovery, using techniques such as clustering, classification, and frequent pattern discovery, has become increasingly important. A data stream is an ordered sequence of instances that arrive at a high rate. This characteristic imposes additional constraints on the mining algorithms to be efficient enough to keep up with the fast rate of arrival and also requires an efficient memory usage as not all data instances can be stored in memory. Many techniques that find frequent patterns from data streams have been proposed, such as CPS-Tree [17] and FPStream [6]. Frequent patterns have been widely considered to be informative and useful but in some domains and scenarios rare patterns may be more interesting. Rare patterns are patterns that do not occur frequently and can sometimes be considered as exceptions. Rare patterns often represent irregular behaviors such as frauds. The detection of rare patterns can benefit a wide range of domains such as fraud detection in credit card transactions and auctions. The mining of rare patterns from data streams has been considered in some previous research, such as SRP-Tree [9].

An important characteristic of data streams is that changes in the underlying distribution can signal important changes in the data stream. Many drift detection techniques have been proposed to detect these changes. However,

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 437–448, 2014.

these techniques are designed with the focus of detecting drift in data streams that contain class labels which are intended for supervised machine learning methods such as classification. These drift detection techniques (*e.g.* ADWIN2 [3]) take in binary inputs that are derived from the error rates of a classifier run on the labeled data stream. Because these techniques are designed for use in labeled data streams, they cannot be applied directly onto unlabeled data streams that frequent and rare pattern mining techniques take in as input.

A naive method for detecting changes in patterns would be to mine the stream for a set of patterns at given intervals and then compare the sets of patterns. This is not the suitable approach, especially in the case of rare patterns, where it is often harder and more costly to discover rare patterns from data streams. A more enlightened scheme would be to apply drift detection techniques at an item level. Therefore, instead of running one instance of the drift detection technique (*e.g.* ADWIN2) on the stream, multiple instances of the technique is run on each separate item (or a subset of the items) found in the unlabeled stream. The binary inputs into ADWIN2 can be derived from the presence or absence of the item in a series of transactions where the binary input 1 would represent that the item occurs in the transaction and the binary 0 would represent the item did not occur in the transaction. For example, consider three transactions: $T_1 : \{a, b, c\}, T_2 : \{a\}, T_3 : \{a, b\}$. The binary inputs used for item a would be $\{111\}$ as item a appears in all three transactions and the input for item b would be $\{101\}$ as it does not occur in transaction T_2. Essentially this monitors the support change of the individual items in the stream and a detected change in this case would represent that the item is either occurring in more transactions or occurring in fewer transactions than it did previously. The issue with this method is that only pattern changes caused by support variations in items will be detected. If there is a change in pattern, but without an accompanying change in support of items, the change will not be detected. For example, consider 4 items $\{a, b, c, d\}$ where $\{a, b\}$ always occur together forming a pattern and $\{c, d\}$ always occur together forming another pattern. If the support of these items do not change but now item a occurs with item c and item b occurs with item d, then this change will not be picked up by simply monitoring the support of items using current drift detection techniques. In this paper, we describe this form of pattern change as a change in item association.

Motivated by the difficulties and the lack of methods for detecting rare pattern changes from unlabeled data streams, the aim of this paper is to address this problem by proposing a novel approach that enables such detection. We propose a novel M measure that enable the detection of both rare pattern changes caused by support change and item association change. The M measure consolidates the state of association of items into one numerical value and in our approach, instead of monitoring the supprot of items as described earlier, we monitor the M measure. Through our evaluations, we demonstrate that this overall approach is capable of detecting rare pattern changes.

There are several scenarios where detecting a change in item associations can be useful. For example, consider a stream of data for recording a series of

traceroutes where an item represents a host in the route. Through monitoring a subset of the items (hosts), the user can identify whether there are changes in the relationship of the hosts. A change in the relationship of the hosts could represent a major change in the routing behavior of the network and signal a possible congestion in the network. The prompt identification of these changes can allow the user to quickly respond to these situations.

The major contributions of this paper are as follows:

1. We present a new approach that enables the detection of rare pattern changes from unlabeled data streams. To the best of our knowledge there has been no previous work on this topic.
2. We propose a novel M measure that is a consolidated numerical measure and represents the state of item associations for an item at any point in the stream. Through monitoring this measure using techniques like the Page-Hinkley Test, changes in rare patterns can be detected in an efficient manner. The M measure enables the discovery of pattern changes relating to changes in item associations that was previously not possible.

The rest of the paper is as follows: in Section 2 we detail the current state of research in the areas of drift detection and pattern mining in data streams. Section 3 describes the preliminaries and definitions of the problem we address. In Section 4 we introduce our overall framework for solving the problem of drift detection of rare patterns and introduce our novel M measure. In Section 5 we present the experimental evaluations and analysis of our algorithm and lastly Section 6 concludes the paper.

2 Related Work

There has been intense research in the area of rare pattern mining. Most of the research was designed for a static database environment and can generally be divided into level-wise exploration or tree based approaches. Level-wise approaches are similar to the Apriori algorithm [2] developed by Agrawal. In the Apriori algorithm, k-itemsets (itemsets of cardinality k) are used to generate $k+1$-itemsets. These new $k+1$-itemsets are pruned using the downward closure property, which states that the superset of a non-frequent itemset cannot be frequent. Apriori terminates when there are no new $k+1$-itemsets remaining after pruning. MS-Apriori [13], ARIMA [16], AfRIM [1] and Apriori-Inverse [12] are algorithms that detect rare itemsets. They all use level-wise exploration similar to Apriori, which have candidate generation and pruning steps. The RP-Tree algorithm was proposed by Tsang et al. [18] as a tree based approach. RP-Tree avoids the expensive itemset generation and pruning steps of Apriori and uses a tree structure, based on FP-Tree [7], to find rare patterns. Both the Apriori and tree based approaches find rare patterns in static database environments. Recently the SRP-Tree algorithm [9], an adaptation of the RP-Tree algorithm, was developed to enable the capturing of rare patterns in a stream environment. There is currently no relevant work in rare pattern mining that also considers drifts in data streams.

Sebastiao and Gama [15] present a concise survey on drift detection methods. They point out that methods used fall into four basic categories: Statistical Process Control (SPC), Adaptive Windowing [5], Fixed Cumulative Windowing Schemes [10] and finally other classic statistical drift detection methods such as the Martingale frameworks [8], the Page-Hinkley Test [14], and support vector machines [11]. Gama et al. [5] adapted the SPC approach. They proposed the Drift Detection Method (DDM) based on the fact, that in each iteration an online classifier predicts the decision class of an example. That prediction can be either true or false, thus forming the binary input for the method. More recently Bifet et al. [3] proposed an adaptive windowing scheme called ADWIN that is based on the use of the Hoeffding bound to detect concept change. The ADWIN algorithm was shown to outperform the SPC approach and has the attractive property of providing rigorous guarantees on false positive and false negative rates. ADWIN maintains a window (W) of instances at a given time and compares the mean difference of any two subwindows (W_0 of older instances and W_1 of recent instances) from W. If the mean difference is statistically significant, then ADWIN removes all instances of W_0 considered to represent the old concept and only carries W_1 forward to the next test.

In addition, ADWIN has also been used in the IncTreeNat algorithm [4] that performs mining of frequent closed trees using streaming data which finds frequent closure patterns in closed trees. Currently there is no research that specifically looks at finding changes in rare patterns from data streams.

3 Preliminaries

In this section we present the preliminaries of drift detection in Section 3.1 and formally define rare patterns and itemsets in Section 3.2.

3.1 Drift Detection

Let us frame the problem of drift detection and analysis more formally. Let $S_1 = (x_1, x_2, ..., x_m)$ and $S_2 = (x_{m+1}, ..., x_n)$ with $0 < m < n$ represent two samples of instances from a stream with population means μ_1 and μ_2 respectively. The drift detection problem can be expressed as testing the null hypothesis H_0 that $\mu_1 = \mu_2$, *i.e.* the two samples are drawn from the same distribution against the alternate hypothesis H_1 that they are drawn from different distributions with $\mu_1 \neq \mu_2$. In practice the underlying data distribution is unknown and a test statistic based on sample means is constructed by the drift detector. A false negative occurs if the null hypothesis is accepted incorrectly when a change has occurred. On the other hand if the drift detector accepts H_1 when no change has occurred in the data distribution then a false positive is said to have occurred. Since the population mean of the underlying distribution is unknown, sample means need to be used to perform the above hypothesis tests. The hypothesis tests can be restated as the following. We accept hypothesis H_1 whenever $Pr(|\hat{\mu_1} - \hat{\mu_2}|) \geq \epsilon) > \delta$, where $\delta \in (0, 1)$ and is a parameter that

controls the maximum allowable false positive rate, while ϵ is a function of δ and is the test statistic used to model the difference between the sample means.

In all drift detection algorithms, a detection delay is inevitable but should be minimized. A detection delay can be expressed as the distance between $\hat{c}$ and $n^{'}$, where $\hat{c}$ is the true drift point and $n^{'}$ is the instance at which change is actually detected. Thus detection delay is determined by $\left(n^{'} - (\hat{c}+1)\right)$. Detection delay is used as one of our evaluation measures in this research.

3.2 Rare Patterns and Itemsets

Let $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ be a set of literals, called items. A set $X = \{i_l, \ldots, i_m\} \subseteq I$ and $l, m \in [1, n]$, is called an itemset, or a k-itemset if it contains k items. A transaction $t = (tid, Y)$ is a tuple where tid is an identifier and Y is a pattern. An association rule is an implication $X \rightarrow Y$ such that $X \cup Y \subseteq \mathcal{I}$ and $X \cap Y = \emptyset$. X is the antecedent and Y is the consequent of the rule. The *support* of $X \rightarrow Y$ is the proportion of transactions that contain $X \cup Y$. The *confidence* of $X \rightarrow Y$ is the proportion of transactions containing X that also contain Y.

We adopt the rare itemsets concept from Tsang et al. [18]. We consider an itemset to be rare when its support is below a threshold, the minimum frequent support (minFreqSupp) threshold. We also define a noise filter threshold to prune out the noise called the minimum rare support (minRareSupp) threshold.

Definition 1. *An itemset X is a* rare itemset *in a window $\mathcal{W}$ iff*

$$sup_{\mathcal{W}}(X) \leq minFreqSupp \text{ and } sup_{\mathcal{W}}(X) > minRareSupp$$

However not all rare itemsets that fulfill these properties are interesting so we only consider *rare-item itemsets* in this paper.

Rare item itemsets refer to itemsets which are a combination of only rare items and itemsets that consist of both rare and frequent items.

Definition 2. *An itemset X is a* rare-item itemset *iff X is a* rare itemset ***and***

$$\exists x \in X, sup_{\mathcal{W}}(x) \leq minFreqSupp \text{ and } sup_{\mathcal{W}}(x) > minRareSupp$$

4 Our Algorithm

In this section we discuss our RPDD (Rare Pattern Drift Detector) algorithm for detecting changes in rare patterns. This approach is designed to find rare pattern changes from unlabeled transactional data streams intended for unsupervised association rule mining. Overall the framework has two components. One of the components is the processing of stream data and we detail it in Section 4.1. In Section 4.2 we detail the actual drift detection component of the algorithm. Here we introduce our novel M measure, a consolidated numerical measure that represents the state of item associations, and discuss drift detection techniques used in this paper that monitors the M measure to detect changes.

4.1 Stream Processing and Item Selection

As transactions from the stream are fed into the algorithm, they are processed and maintained using a sliding window W of size $|W|$. A list of item frequencies is recorded while the stream is processed. We use the minFreqSupp and minRareSupp defined in the preliminaries to identify the rare items. A rare item is an item with a support value between the minFreqSupp and minRareSupp. All the rare items found are forwarded to the drift detection component where individual item tracking is performed.

An example is given Table 1. We set $minFreqSupp = 3$ and $minRareSupp = 1$. Based on the thresholds, items b and c are selected as rare items for tracking from this window of transactions. Although the selection of items for tracking in this example is based on the measures of support, this does not have to be the case. The user can specify other ways of selecting items to be tracked or use a specific list of items that are of interest based on domain knowledge.

4.2 Drift Detection Using M Measure

The selected rare items from the stream processing component will be individually tracked and monitored. Essentially, each selected item will have a separate process that monitors the item associations it has with other items in the stream using our M measure.

For each selected item, a separate local item frequency list is maintained that keeps track of the frequency of occurrence of other items with the selected item. For example, consider the previous example reproduced below:

Table 1. Transactions

tid	transaction
1	a b c
2	a b
3	a c
4	a h j

Table 2. Global Freq List

item	freq
a	4
b	2
c	2
h	1
j	1

Table 3. Local Freq List for item c

item	freq
a	2
b	1

Since items b and c were selected as tracking items, in this phase, these two items are separately monitored. For each monitored item, a local frequency list is maintained that consists of item associations of the monitored item with other items in the stream. For example, for item c, the local frequency list would consist of the items that occur with item c in the same transaction and their respective co-occurrence frequencies. The local frequency list for item c is shown in Table 3. These local frequency lists for each item will be used by our novel M measure to detect changes in item associations.

The M measure is based on the absolute percentage difference in support of an item. The M measure is a consolidated measure that represents the level of change in the item associations in an item at a given time t.

The M measure for an item x at time t is given by the following formula:

$$M(x)_t = \frac{1}{|X|} \times \sum_{i \in X}^{|X|} \frac{|Supp(i)_t - Supp(i)_{t-1}|}{Supp(i)_{t-1} + c}$$

where X is the set of items that occurs with item x, $Supp(i)$ is the support of i in the local frequency list, and c is the Laplacian coefficient.

The Laplacian coefficient c should be set at a low value such as 0.01 and is introduced to avoid cases of undefined values. In most cases c can be set equal to the minimum rare support threshold specified by the user earlier. The lower bound of the M measure is 0 and is reached when $Supp(i)_t - Supp(i)_{t-1}$ is equal to 0 for all $i \in X$. The upper bound of the M measure is $1/c$ and is reached when $Supp(i)_t = 1$ and $Supp(i)_{t-1} = 0$ for all $i \in X$.

The M measure is calculated for each transaction and for each tracked item individually. Then, we monitor the M measures across time t by applying existing drift detection methods. A fluctuation in the M measure for an item would represent a rare pattern change.

The usage of the M measure is crucial in achieving the goal of detecting changes in item associations and patterns because if users choose to monitor only the support of items, then pattern changes related to change in item associations would not be detected. The M measure is specifically useful for finding rare item association changes because of its property of using percentage difference in support. Since we consider rare items and rare items are characterized by having a lower support value, a small fluctuation in support of these values would result in a much higher percentage difference. For example, if a rare item changes in support from 0.1 to 0.05, the percentage difference is 100%, whereas, if a frequent item goes through the same support change 0.90 to 0.85, the percentage difference is then 6%. The M measure, based on percentage difference, allows us to monitor the changes in rare items more closely and enables the detection of drifts that results from a smaller magnitude of change that would otherwise be missed using pure support based strategies.

In this paper we use two different drift detection techniques to monitor our M measure and find changes in the item associations: the Page-Hinkley Test and the Hoeffding bound with Fixed Size Flushing Window. These two techniques operate on numerical inputs as required by our M measure. We cannot use ADWIN2 as the drift detection technique here because ADWIN2 takes in binary input values and our M measure is not a binary measure.

Page-Hinkley Test. As described by Sebastiao in [15], the Page-Hinkley Test is a sequential analysis technique and designed to detect change in the average of a Gaussian signal. The test monitors a cumulative variable U_T which represents the cumulative difference between observed values and the cumulative mean of the samples. U_T is given by the formula: $U_T = \sum_{t=1}^{T} (x_t - \mu_T - \delta)$ where $\mu_T = \sum_{t=1}^{T} x_t \times \frac{1}{T}$, the cumulative mean at T, and δ is the magnitude threshold.

To detect drifts, it calculates the minimum value of U_t, $m_T = min(U_t, t = 1 \ldots T)$ and the difference between U_T and m_T is monitored at each time t.

A change is signaled when $U_T - m_T > \lambda$ where λ is the detection threshold. The detection threshold, λ, controls the balance between false positives, true positive rates and detection delay of the test. Setting a high λ value will lower false positives but will increase detection delay and might miss drifts, whereas setting a low λ value will increase false positives but will decrease detection delay and increase true positive rates. In practice it is often difficult to find the optimal λ value as the setting of the threshold can vary widely depending on the input.

Hoeffding Bound with Fixed Size Flushing Window. As the Page-Hinkley Test requires the setting of thresholds that can sometimes require an extensive amount of trial and error experimentation, we also use the statistical Hoeffding bound which requires only the setting of one δ parameter while providing statistical guarantees on error bounds. The Hoeffding bound provides an upper bound on the probability that the sum of the random variables deviates from its expected value. The Hoeffding bound states that, with probability $1 - \delta$, the true mean of a random variable r is at least $\bar{r} - \epsilon$ when the mean is estimated over t samples, where R is the range of r.

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2t}}$$

The Hoeffding bound is applied with a Fixed Size Flushing Window (FSFW) in order to perform drift detection. The FSFW consists of a window W with size $|W|$. W is split into two separate blocks of equal size (B_L and B_R). The window is filled as sample values arrive at each time t. When the window is full, the oldest instance is dropped as new instances arrive. At each time t, Hoeffding bound is applied to check for the difference in the mean between the two blocks within the window W. If $|\mu_{B_L} - \mu_{B_R}| > \epsilon$ then a drift is signaled and samples in B_L are flushed and replaced with the samples in B_R. For the purpose of optimization, the FSFW is coded with a circular array to eliminate the cost of shifting items from B_R to B_L. The window size $|W|$ in this case should be set to a reasonably large value to allow a statistically sound comparison of two samples (*e.g.* 1000).

5 Experimental Evaluation

In this section we present the experimental results and evaluations we performed on our RPDD algorithm for finding changes in rare patterns from data streams. Specifically we compare the results of using our M measure in drift detection monitored by the Page-Hinkley Test (PHT) against using M measure in the Hoeffding Bound with Fixed Size Flushing Window. The performance measures used are: True Positives, False Positives, Detection Delay, Execution Time, and Memory. The algorithms are coded in Java and run on an Intel Core i5-2400 CPU @ 3.10 GHz with 8GB of RAM running Windows 7 x64. The execution time reported excludes I/O costs.

In the first experiment we compare the number of false positives obtained by applying the M measure with PHT and Hoeffding bound based drift detection

methods. The experiment is set up by generating item transaction streams using a modified IBM Quest Market Basket Data Generator. We control one rare item in the stream by forcing it to undergo a pattern drift at mid-point of the stream (e.g. If R is the rare item then an example of the pattern drift can be: $\{A, B, C, R\}$ to $\{X, Y, Z, R\}$). The controlled item has varying support values (0.05 - 0.2) across the stream and the stream size is 1M. All experiments are run 100 times and the results shown are average values across the 100 runs.

The results for the average number of false positives found are shown in Table 4. For the Page-Hinkley Test experiments we used a constant δ threshold of 0.001 while varying the λ threshold. The Laplacian coefficient c is set at 0.01 for all experiments. The Hoeffding bound is tested using δ values of 0.05, 0.1, and 0.3. The window size $|W|$ is set at 1000 for all experiments.

Overall we observe that PHT produces more false positives than Hoeffding bound. The number of false positives for PHT decreases as λ increases, as expected. The number of false positives for $\lambda = 200$ is extremely high showing that the λ value is too low and that the technique is raising most points as drifts. As we increase the λ value, we see that the number of false positives decreases significantly reaching a more acceptable state. The number of false positives for Hoeffding bound slightly increases as δ increases, also as expected.

Table 4. False positives

	PHT					Hoeffding Bound		
Support	$\lambda = 200$	$\lambda = 400$	$\lambda = 600$	$\lambda = 800$	$\lambda = 1000$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.3$
0.05	977.58	36.98	13.13	10.2	8.64	2.29	2.64	3.20
0.1	1884.42	58.38	13.57	10.51	8.95	2.26	2.70	3.38
0.15	2803.16	84.91	13.99	10.74	9.11	2.36	2.75	3.34
0.2	3760.67	116.01	14.15	10.73	9.14	2.34	2.60	3.30

In the second experiment we compare the true positive rates of the techniques. The experiment is set up similar to the false positives experiments except instead of varying the support level of the controlled item, in the true positive experiment we varied the percentage of pattern change. In these experiments the support of the controlled item is set at 0.1. The percentage of pattern change represents the magnitude of pattern change with 0% representing no pattern change and 100% representing a complete pattern change in the controlled item. The percentage of pattern change can be viewed as the change in the items that the controlled item is associated with. It is distinctly different from a change where there is an increase or decrease in the support value of the patterns.

Overall the true positive rates of both techniques at detecting the drift is 1.00 with the exception of Hoeffding bound with the conditions of 25% pattern change and $\delta = 0.05$. This is likely due to the compounded effects of a smaller magnitude of drift produced by the 25% pattern change and a smaller δ value which produces a tighter bound.

In the third experiment we experiment on the detection delay of the two techniques. The experiment is set up exactly the same as the true positive experiments. The detection delay is the distance between $\hat{c}$ and n', where $\hat{c}$ is the

Table 5. True positive rates

	PHT					Hoeffding Bound		
% Change	$\lambda = 200$	$\lambda = 400$	$\lambda = 600$	$\lambda = 800$	$\lambda = 1000$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.3$
25%	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
50%	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
75%	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
100%	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 6. PHT: Detection delay

% Change	$\lambda = 200$	$\lambda = 400$	$\lambda = 600$	$\lambda = 800$	$\lambda = 1000$
25%	186.26±(100.86)	626.61±(427.56)	1159.79±(453.03)	1429.05±(508.83)	1693.81±(551.30)
50%	160.83±(89.44)	477.29±(341.77)	949.30±(407.01)	1214.16±(442.61)	1412.18±(506.27)
75%	152.27±(76.47)	398.64±(276.37)	844.03±(315.43)	1109.62±(370.61)	1303.90±(434.68)
100%	122.95±(70.69)	367.04±(292.02)	785.38±(321.16)	1025.64±(377.68)	1294.21±(430.63)

Table 7. Hoeffding Bound: Detection delay

% Change	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.3$
25%	3338.80±(459.53)	3028.56±(427.60)	2396.52±(346.82)
50%	3083.74±(471.49)	2765.02±(422.96)	2137.10±(338.95)
75%	2897.53±(412.59)	2585.15±(365.43)	1963.74±(282.92)
100%	2863.67±(388.49)	2548.06±(352.91)	1915.35±(279.10)

true drift point and n' is the instance at which change is signaled. Thus, delay is determined by $\left(n^{'} - (\hat{c} + 1)\right)$ as described in Section 3.

We observe that as the % change of the pattern is increased, the delay is reduced. This observation meets expectation as a higher % change represents a higher magnitude of change and would result in an earlier detection. PHT generally has a lower detection delay but this is at the cost of a higher false rate.

In the last set of experiments we experiment on the execution time and memory use of tracking a various number of items for drift detection. Since the execution time and memory use is heavily reliant on the number of items selected for tracking and the compounded effects of tracking multiple items, we experimented with a varying number of tracked items. Table 8 shows the results. The execution time is reported in ms and the memory is reported in bytes.

We observe that the execution time and memory use of the algorithm for both PHT and Hoeffding bound increases in a linear fashion as the number of tracked

Table 8. Execution Time and Memory Use

	PHT		Hoeffding Bound	
# Items	Time (ms)	Memory (bytes)	Time (ms)	Memory (bytes)
1	9115±(3758)	586459±(13044)	11618±(5279)	595867±(13014)
2	14780±(6991)	636018±(14317)	21151±(10664)	654649±(14318)
3	20130±(9803)	685151±(15978)	29584±(15102)	712996±(15978)
4	25703±(12312)	733791±(18547)	36870±(18235)	770836±(18548)
5	32209±(14533)	783120±(20315)	47386±(22117)	829373±(20317)
10	61859±(24527)	1026566±(29311)	92722±(37962)	1118854±(29317)
25	130840±(34240)	1751869±(50487)	198476±(51754)	1982639±(50508)
50	250506±(37330)	2936653±(66849)	417208±(64545)	3398090±(66880)

items increase. Overall PHT executes faster and uses less memory due to the extra costs of maintaining the window in the Hoeffding bound technique with Fixed Size Flushing Window. The standard costs of processing and maintaining a sliding window in the stream processing component constitutes the base execution time and memory cost across the two techniques.

6 Conclusion and Future Work

In this paper we proposed a new approach that deals with the problem of detecting rare pattern changes in unlabeled data streams. Our approach uses a novel measure, the M measure, that is a consolidated numerical value which represents the status of item associations in a stream for a given item at a given time. Our experimentation showed that the use of the M measure in conjunction with drift detection techniques enabled the detection of changes in rare patterns that are otherwise undetectable using standard support based detection approaches.

Our future work includes developing a drift detection technique that is optimized with the aim of detecting rare patterns drifts. Even though drift detection techniques such as the Page-Hinkley test works relatively well in this scenario, it requires the setting of the λ parameter and the overall detection scheme is not optimized to the proposed M measure. We also want to adapt the M measure to detect drift in frequent patterns and investigate the possibility of combining and adapting other frequent pattern mining mechanisms such as the CPS-Tree with M measure.

References

1. Adda, M., Wu, L., Feng, Y.: Rare itemset mining. In: Proceedings of the Sixth International Conference on Machine Learning and Applications, ICMLA 2007, pp. 73–80. IEEE Computer Society, Washington, DC (2007)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Santiago, Chile, pp. 487–499 (1994)
3. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: SIAM International Conference on Data Mining (2007)
4. Bifet, A., Gavaldà, R.: Mining adaptively frequent closed unlabeled rooted trees in data streams. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 34–42. ACM, New York (2008), http://doi.acm.org.ezproxy.auckland.ac.nz/10.1145/1401890.1401900
5. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
6. Giannella, C., Han, J., Pei, J., Yan, X., Yu, P.S.: Mining frequent patterns in data streams at multiple time granularities. Next Generation Data Mining 212, 191–212 (2003)

7. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000, pp. 1–12. ACM, New York (2000)
8. Ho, S.S., Wechsler, H.: A Martingale framework for detecting changes in data streams by testing exchangeability. IEEE Trans. on Pattern Analysis and Machine Intelligence 32(12), 2113–2127 (2010)
9. Huang, D., Koh, Y.S., Dobbie, G.: Rare pattern mining on data streams. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 303–314. Springer, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-32584-7_25
10. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the Thirtieth International Conference on VLDB, vol. 30, pp. 180–191. VLDB Endowment (2004)
11. Klinkenberg, R., Joachims, T.: Detecting concept drift with support vector machines. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML), pp. 487–494 (2000)
12. Koh, Y.S., Rountree, N.: Finding sporadic rules using apriori-inverse. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 97–106. Springer, Heidelberg (2005)
13. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 337–341 (1999)
14. Page, E.S.: Continuous inspection schemes. Biometrika 41(1/2), 100–115 (1954), http://www.jstor.org/stable/2333009
15. Sebastiao, R., Gama, J.: A study on change detection methods. In: 4th Portuguese Conf. on Artificial Intelligence, Lisbon (2009)
16. Szathmary, L., Napoli, A., Valtchev, P.: Towards rare itemset mining. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007, vol. 01, pp. 305–312. IEEE Computer Society, Washington, DC (2007)
17. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S., Lee, Y.K.: Efficient frequent pattern mining over data streams. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 1447–1448. ACM, New York (2008)
18. Tsang, S., Koh, Y.S., Dobbie, G.: RP-Tree: Rare pattern tree mining. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 277–288. Springer, Heidelberg (2011)