A novel incremental approach for stream data mining

Hatim A. Aboalsamh
Computer Sciences, King saud Unversity, Saudi Arabia
Hatim@ccis.k.su.edu.sa http://faculty. ksu.edu.sa/Aboalsamh/

With the recent advances in data collection systems, different continuously generated data have been collected in a wide range of applications. According to that, stream data analysis is considered as a crucial component of strategic control over a broad variety of disciplines in business, science and engineering. Many examples could be considered as applications for stream data mining in areas such as stock markets, sensor networks, network traffics, and explicit time series. Stream data mining has recently received much interest from researches working in the field of data mining. Mining data streams is defined as extracting knowledge structures from continuous streams of information. Online analysis on data streams is needed where data values change frequently. With the increase of computational operations for mining data streams, finding efficient techniques is considered a challenging task. Unfortunately most of the existing data mining algorithms have been designed for stored data, and could not be considered on data streams. For on-line mining tasks, new techniques in areas such as classification, clustering, frequent itemsets mining, pattern matching, etc., should be presented. Also, suitable architectures should be proposed for handling new mining methods. In this paper, we propose a novel stream data mining technique based on the association mining approach, along with a suitable structure for generating continuous or transient rules. The proposed technique collects knowledge through dynamic windows, where computations are done in an incremental fashion. A hierarchical structure is designed for storing transient patterns. The experimental results show that the performance of the Incremental Approach for Stream Data Mining (IASDM) technique against the Closed Frequent Itemsets (CFI) -Stream technique, which is considered as one of the latest and accredited techniques in the area of stream data mining, is improving the process of stream data mining by almost 60%.

في الآونة الأخيرة ومع التقدم الحديث في نظم تجميع البيانات مما ادي الي وجودالعديد من البيانات المختلفة التي يتم تجميعهابشكل مستمر في مجموعة واسعة من التطبيقات. ووفقا لذلك فإن التنقيب في تيارات البيانات يعتبر عنصرا حاسما من عناصر استراتيجية السيطرة على مجموعة واسعة من التخصصات في مجال الأعمال التجارية والعلوم والهندسة. وهناك العديد من الأمثلة التي يمكن أن تعتبر تطبيقات للتنقيب في تيارات البيانات ومنها مجالات مثل أسواق الأسهم ، شبكات أجهزة الاستشعار ، والشبكات التحارية والسلاسل الزمنية. وفي الآونة الأخيرة لقي التنقيب في تيارات البيانات اهتماما كبيرا من البحوث العاملة في مجال التنقيب عن البيانات. يتم تعريف التنقيب في تيارات البيانات علي انها استخراج المعرفة من تيارات المعلومات المستمرة. وهناك احتياج شديد للتحليل الفوري لتيارات البيانات والتي حيث كثيرا ما تتغير. ومع العمليات الحسابية المتقدمة المرتبطة بتيارات البيانات فإن إيجاد أساليب فعالة للتنقيب يعتبر مهمة صعبة. ومما يؤسف له أن معظم الخوارزميات المتواجدة للتنقيب في البيانات قد صممت للبيانات المخزنة ولا يمكن استخدامها مع تيارات البيانات. ولعمليات التنقيب الفوري فلابد من تقديم تقنيات حديثة في مجالات مثل التصنيف ، والتجميع ، وفئات الاصناف المرتبطة ومطابقة الانماط ، وما إلى ذلك. كذلك ، ينبغي أن يتم تقديم تصاميم مناسبة للتعامل مع الأساليب الجديدة للتنقيب. في هذه الورقة العلمية فإننا نقترح طريقة حديثة للتنقيب في تيارات البيانات على أساس نهج التنقيب عن الارتباطات إلى جانب هيكل مناسب لتكوبن قواعد متواصلة أو عابرة. وفي الاسلوب المقترح يتم تجميع المعرفة من خلال نوافذ ديناميكية حيث تتم العمليات الحسابية في على نحو تدريجي. ويتم تصميم هيكل هرمي لتخزين الأنماط العابرة. وقد اوضحت نتائج التجارب أنه بمقارنة أداء طريقة IASDM المقترحة مع اداء طريقة CFI- التي تعتبر واحدة من أحدث الطرق المستخدمة والمعتمدة في مجال التنقيب في تيارات البيانات—وجد ان هناك تحسين في عمليات التنقيب في تيارات البيانات بنحو ٦٠ ٪.

Keywords: Stream data mining, Association rule mining, Incremental data mining, Dynamic data mining, Frequent closed itemsets

1. Introduction

With the tremendous increase of data collected through mobile and non-mobile computing and networking environments, and the emergence of new applications such as, medical monitoring applications and network traffic monitoring, a new research area has forced itself on researchers' interests. Many characteristics of data streams that have not been addressed in traditional data mining research should be noted, such as, variable data rates, mobility, disconnections and environmental changes. To perform data stream mining in distributed computing environments, applications need to monitor context changes and react to them in order to continue the required tasks.

Stream data mining has the same conceptual definition of the traditional data mining, which is the process of discovering potentially valuable patterns, associations, trends, sequences and dependencies in data streams [2-5, 13]. A data stream is an ordered sequence of transactions that arrive in a timely order. Data streams are different from traditional static data. Data streams have the following three characteristics:

- Data streams are frequent, continuous, and unbounded.
- Size of data streams is large and with an open end.
- Data distribution in streams usually changes with time.

Analysis on stream data include discovering trends (or patterns) in data stream sequences such as telephone records, network traffics monitoring, web click-stream analysis, highway traffic congestion analysis, market basket data mining, etc. In business domain, examples include web site access patterns, improving e-commerce advertising, fraud detection, product analysis, and customer segmentation. These problems present two main research issues:

- Finding new algorithms that suit the requirements of online applications.
- Building systems that efficiently support such algorithms in an integrated and extensible manner.

Stream patterns are classified as either systematic patterns, where patterns can be determined at specific points of time, or non-systematic patterns. Many researchers have been working on different algorithms on stream analysis. The main line of research in stream data mining is identifying or describing patterns of observed stream data. Once a pattern is established, we can interpret and integrate it with other data. The identified pattern can be extrapolated to predict future events.

The rest of this paper is organized as follows: in section 2, we give related works in the area of stream data mining. In section 3, we give the problem definition. The incremental stream data mining technique is proposed in section 4. In section 5, we give the analysis and performance study of the proposed approach. The paper is concluded in section 6.

2. Related work

In the literature, various approaches have been proposed to address the problem of mining data streams. Most of those techniques have proposed approaches that use one scan over the entire data stream. Two main methodologies are used in stream data mining, sliding window models [3-5, 8], and no sliding window models [2, 7, 12]. In sliding window methodology, different models have been proposed to work on recent arrived data. In no sliding window methodology, most of the introduced algorithms have been working over the whole history of data streams to generate frequent itemsets.

In [4], the Moment algorithm was introduced to mine closed frequent itemsets over data stream sliding windows. A selected set of itemsets is maintained. That set includes infrequent gateway nodes, unpromising gateway nodes, intermediate nodes, and closed nodes. For each node, the itemset, its type, support and sum of the ids of the transactions in which the itemset occurs (tid_sum) are stored. The selected itemsets work as the boundary between closed frequent itemsets and the other itemsets. When a transaction arrives, it tests the closed frequent itemsets stored in a hash table with its support and tid_sum information to decide its node type, and the associated node information is modified. The closed itemsets are checked and excluded from the other types of nodes stored in the data structure. One of the drawbacks of the proposed approach is that, when the minimum support value is small, it uses much memory for storing more information other than the current closed frequent itemsets.

In [7], CFI-Stream algorithm directly computes closed itemsets, online and incrementally, without support information. Only closed itemsets are maintained in their data structure. For each arriving new transaction, the algorithm performs closure checking. The associated closed itemsets and their support information are updated. The current closed frequent itemsets can be output in real time based on any user's specified thresholds.

In [8], a data estimation technique using association rule mining on stream data is proposed. The closed frequent itemsets approach CARM is used as a basis for the proposed technique. In CARM, the relationships between sensors are discovered and used for missing data compensation. It discovers the relationships between two or more sensors when same or different values are recorded. The generated association rules provide complete and non-redundant information.

In this paper, we introduce a novel incremental stream data mining technique. In this technique, we use history of frequent closed itemsets for generating new frequent closed itemsets. A sliding window is used to capture the continuity of data streams. Two classes of closed itemsets are recorded. The first class represents frequent closed itemsets whose supports exceed a support threshold. The other class is used to keep history of promising frequent closed itemsets that could be turned into frequent closed itemsets. A non frequent closed itemset is considered promising frequent closed itemset based on some parameter α, *a>1*. The proposed approach favors those itemsets in the current window that are frequent closed The proposed technique works in real time, the same way the CFI and CARM work.

3. Problem definition

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n elements, called items. A subset $X \subset I$ is called an itemset. A k-subset is called a k-itemset. Each transaction t is a set of items in I. Given a set of transactions *T*, the support of an itemset X is the percentage of transactions in *T* that contain X.

The concept of a closed itemset is based on the following. Let *T* be a subset of transactions in data stream *D*, and *X* be a subset of *I*, $X \subset I$. Consider the following two functions, *f* and *g*, where

$f(T) = \{i \in I \,/\, \forall t \in T, i \in t\}$ and

$g(X) = \{t \in D \,/\, \forall i \in X, i \in t\}$, or

$$f(T) = \bigcap_{\forall t \in T} f(t) \text{ and } g(X) = \bigcap_{\forall x \in X} g(x)$$

$f(T)$ returns the set of common items in all transactions *tεT*, and $g(X)$ returns the set of transactions that have *X* as part of all of them.

The support of an itemset *X*, *Sup*(*X*), is the number of transactions in which *X* occurs as a subset, i.e., Sup(*X*) = $|g(X)|$. An itemset *X* is frequent, if *Sup*(*X*) ≥ minsup. Let $Ch(I) \to h(I)$ be the closure operator, defined as $C(X) = f(g(X))$. A frequent itemset *X* is called closed if and only if *C*(*X*)=*X*, where *X* is subset of I. Alternatively, a frequent itemset *X* is closed if there exists no proper superset *Y*, Y $\subset X$, with *Sup* (*X*) = *Sup* (*Y*).

Definition 1: An itemset *X* is called closed if and only if $C(X) = f(g(X)) = f.(g(X)) = X$ where the composite function $C = f \bullet g$ is called a Galois operator or a closure operator.

Definition 2: An itemset *X* is called a local frequent closed if *X* is frequent closed itemset in the current sliding window w_{last}.

Definition 3: An itemset *X* is called a global frequent closed if *X* is frequent closed itemset in $w_{global} = \bigcup_{i=0}^{i=last} w_i$, where w_0 is the first window that captures the history of *X*, and w_{last} is the current sliding window.

In our proposed technique, we use the current sliding window w_{last} that contains recent transactions arriving to the system, to get local frequent closed itemsets. Count of frequent closed itemset *X* depends on the current count in the current window w_{last}, and the history information recorded for this itemset in previous windows w_i's, where windows w_i's could have different lengths. All itemsets *X's* in the current set of frequent closed itemsets F_i, should have global support and/or local support greater than or equal to the minimum support threshold; minsup. X should be dropped off F_i as soon as its global support and local support get below *minsup*. If *X* appears again in a subsequent window, we do not have any history to tell us what its global support is. In our technique, instead of purging the history of *X* as soon as it is turned m_{last} infrequent, we use a compromised formula for X that keeps its history. Itemsets that satisfy the compromised formula are called promising itemsets.

Definition 4: let *X* be an infrequent itemset; $X \notin F_i$, where *Sup* (*X*) < *minsup. If Sup* (*X*) ≥ $\frac{minsup}{\alpha}, \alpha > 1$, then we keep the history of *X* and keep it in the promising cloaed itemsets F_i^p and continue counting its history as long as its support, *Supt* (*X*) ≥ $\frac{minsup}{\alpha}, \alpha$ is called the promising factor.

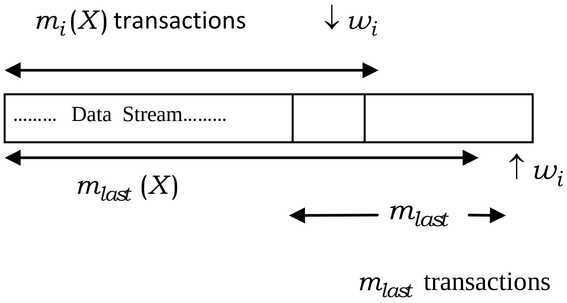
In our proposed technique, we keep the history of only those itemsets in F_i and F_i^p. All itemsets in F_i^p were previously in F_k, $k < i$.

4. The incremental stream data mining technique

Through the rest of this paper, we use the following terminologies.

For window w_i, *i=1,2,…, last*

X	frequent closed itemset,
local.freq(*X*)	count of *X* in w_{last},
local.freq(*X*)=	*local. freq*$_1$ (*X*) + *local. freq*$_2$ (*X*), where
local.freq$_1$(*X*)	count of *X* in the area between the beginning of w_{last} and end of w_{last-1},
local.freq$_2$(*X*)	count of *X* in the area between, the end of w_{last} and end of w_{last-1}
global.freq$_i$(*X*)	count of *X* in the area between the starting point of counting *X* till end of w_i ,
m_i	number of transactions in window w_i ,
m_i (*X*)	number of transactions since the starting count of *X* till end of window w_i.,
F	Set of global frequent closed Itemsets,
F_i	Set of local frequent closed itemset at window w_i.,
F_i^p	Set of promising global frequent closed itemset at window w_i.
α	The promising factor,
Sup$_{local}$ *(x)*	$= \frac{local.freq(X)}{m_i}$, and
Sup$^i_{glocal}$ *(x)*	$= \frac{global.freq_i(X)}{m_i(X)}$.



Lemma 1: If $X \in F$ and $X \notin F_{i-1}$ and $\exists Y \in F_{i-1}$ or $Y \in F_{i-1}^P$ such that $X \subset Y$ and $\nexists Z \in F_{i-1}$ and $\nexists Z \in F_{i-1}^p$ where $X \subset Z \subset Y$ then *global.freq*$_i$(*X*) = *global.freq*$_{i-1}$ *(Y)* + *local.freq*$_2$ (*X*)

Proof: if $X \in F_{i-1}$, then

global.freq$_i$(*X*) =
gobal.freq$_{i-1}$(*X*) +*local. freq*$_2$ (X). (1)

If $X \notin F_{i-1}$, and there exists an itemsest $Y \in F_{i-1}$ *or* $Y \in F_{i-1}^p$ such that $X \subset Y$ and $\nexists Z \in F_{i-1}$ and $\nexists Z \in F^p_{i-1}$ where $X \subset Z \subset Y$ then support of *X* is calculated as

global.freq$_{i-1}$(*X*) +*global. freq*$_{i-1}$ (*Y*). (2)

From eqs. (1 and 2), then:

global.freq$_{i-1}$(*X*) =*global.freq*$_{i-1}$ (*Y*)+
+ *local.freq*$_2$ (X)

Q.E.D.

Corollary 1: In our approach, if X ∈ F and $X \notin F_{i-1}$ and X $\notin F_{i-1}^p$ and $\nexists Y \in F_{i-1}$ and $\nexists Y \in F^p_{i-1}$ such that X⊂Y then *global.freq*$_i$(*X*) = loca.freq(X). In this case, the values of m_i(*X*) is set to m_i.

From lemma 1 and corollary 1, the global frequencies of frequent closed itemsets can be calculated as follows:

- If X$\in$ F and $\exists X \in F_{i-1} or X \in F_{i-1}^{p}$, then

global.freq$_i$ (X) = *global.freq*$_{i-1}$ (X) + *local.freq*$_2$(X),
$m_i(X) = m_{i-1}(X) + m_i$, and $X \rightarrow F_i$
Frequencies of X ancestors may need to be adjusted accordingly.

- if $X \in F$ and $\exists$ F_{i-1} or Y $\notin F_{i-1}^{p}$, such that $X \subset Y$ and $\nexists Z \in F_{i-1}$ and $\nexists Z$ F_{i-1}^{p} where $X \subset Y$ then

global.freq$_i$(X) = *global.freq*$_{i-1}$ (Y) + *local.freq*$_2$(X),
$m_i(X) = m_{i-1}(Y) + m_i$, and X $\rightarrow F_i$

- if $X \in F$ and $\nexists$ Y $\in F_{i-1}$ and $\nexists$ Y$\in F_{i-1}{}^{p}$ such that $X \subset Y$ then *global.freq*$_i$ (X) = *local.freq* (X),

$m_i(X) = m_i$, and X $\rightarrow F_i$

Frequencies of X ancestors may need to be adjusted accordingly.

- if $X \in F$ and $\exists Y \in F_{i-1}$ $\exists$ F_{i-1} or Y$\in F_{i-1}{}^{p}$, X $\subseteq$ Y, *global.freqi*(X) = *global.freq*$_{i-1}$ (Y), + *local.freq*$_2$(X)

  if $\dfrac{global.freq_i(X)}{m_i(X)} \geq minsup$ then

  else if $\dfrac{global.freq_i(X)}{m_i(X)} \geq \dfrac{\min\sup}{\alpha}$ then

  $X \rightarrow F_i^p$ and $m_i(X) = m_{i-1}(X) + m_i$,

Our technique does not depend on the data structure used. The data could be stored in a range of data structures from heap structures to tree structures. In our implementation, we use a tree structure, where global frequent closed itemsets and promising frequent closed itemsets are stored as the global frequent closed itemsets tree and the promising frequent closed itemsets tree, respectively. During working on the current window, a temporary tree is maintained, and the temporary tree is merged to the global frequent closed itemsets tree and the promising global frequent closed itemsets tree. At each node in the global frequent closed itemsets tree and the promising global frequent closed itemsets tree, we keep the following attributes:

- count of X in the area between the beginning of w_{i-1}, *local.freq*$_1$ (X)
- count of X in the area between the end of w_i and end of w_{i-1}, *local.freq*$_2$ (X)
- Global frequency, *global.freqi*(X)
- Number of transactions since the starting count of X, *mi*(X)

The main outlines of the proposed technique are given below.

The Incremental Stream Data Mining Technique

1- For window w_1
Generate F_1, the set of closed frequent itemsets in w_1.
F1 = {X | ($X \in F$ and $Sup_{local}(X) \geq minsup$)}
2- For i=2,3,..., repeat the following:
a. Slide to window w_i
b. Generate F_i and F_i^p

$$F_i = (X \big| X \in F \vee (X \notin F \Lambda (X \in F_{i-1} \vee X \in F_{i-1}^{p} \big\rangle \wedge$$
$$SupglobaliX \geq minsup)\}$$

$$F_i^p = \{X \big| XF\Lambda(X \in F_{i-1} \vee X \in F_{i-1}^{p}) \wedge ,$$
$$\frac{\min\sup}{\alpha} \leq Sup^{i}{}_{global}(X) < \min\sup\}, \alpha > 1\,.$$

5. Analysis and performance study

We have run our experiments on a 2.4 GHz machine, with 4 GB of RAM and running windows Vista. The databases used have been generated synthetically, to evaluate the performance of the algorithms over a range of data parameters. The range of database sizes is varied between 10,000 and 100,000 transactions, and fixed window size of 500 transactions. The average transaction length is varied between 5 and 10 items, and the average length of the maximum pattern is between 4 and 6. We have used the FP-growth algorithm as the basic stream data mining algorithm. The results have been evaluated against the CFI-Stream algorithm [7], which is one of the latest algorithms designed for mining frequent closed sequences. We have measured the mining time for the different database sizes on various minimum support values ranges between 0.2% and 2%, and α ranges between 1 and 2. In figs. 1-9, we have studied the differences between our technique and the CFI-Stream technique for different minimum support thresholds, database sizes, and promising factor values on runtime. In Figs. 1-9, although the use of α =1.25, 1.5 and 2, increases the processing time because of the extra load of handling promising frequent closed itemsets, but we still find that there is a significant difference in the runtime between the two techniques. The improvement of runtime is almost 40%. The difference in runtime is due to the increasing cost of handling and maintaining CFI's proposed tree structure. While in our tree structure we just handle the merging of the temporary tree and the global frequent closed itemsets tree and the promising global frequent closed itemsets tree. Regarding memory usage, we have measured the memory needed for different minimum support values range between 2 and 0.2, and have compared the results to the CFI-Stream technique. As shown in fig. 10, the comparison has been in favor of the CFI-Stream algorithm by the ratio of on average 16.5%. Although, CFI-Stream technique uses more nodes than those used in IASDM, but the increase of memory is due the memory needed for the two tree structures we have used. The results have been measured for database size =50,000 transactions and α=1.5.

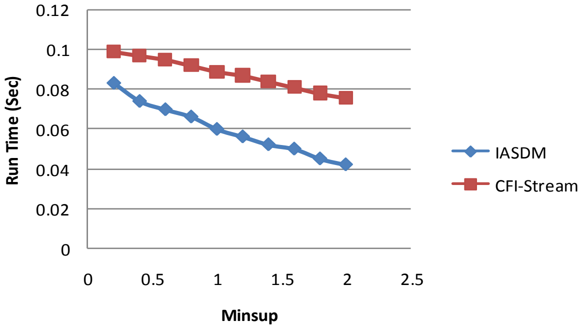


Fig. 1. Data Base Size=10,000, α =1.25.

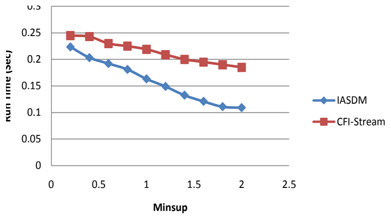


Fig. 2. Data Base Size=50,000, α =1.25.

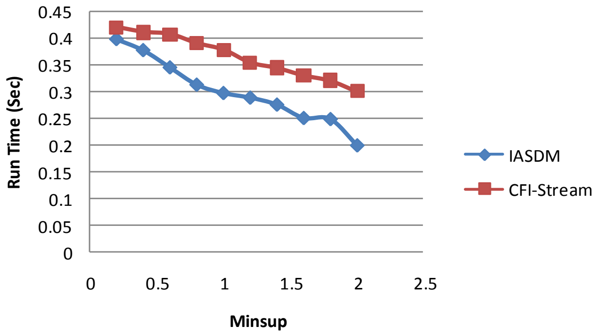


Fig. 3. Data Base Size=100,000, α =1.25.

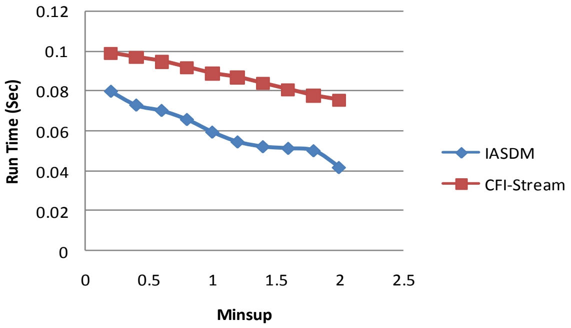


Fig. 4. Data Base Size=10,000, α =1.5

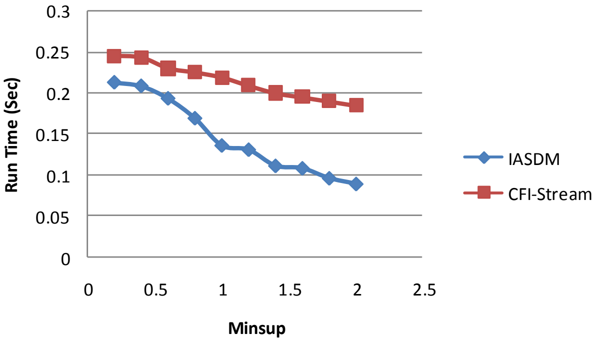


Fig. 5. Data Base Size=50,000, α =1.5

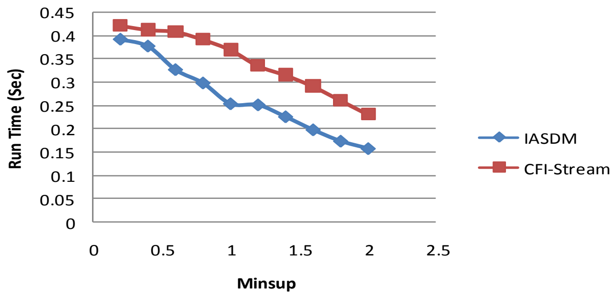


Fig. 6. Data Base Size=100,000, α =1.5.

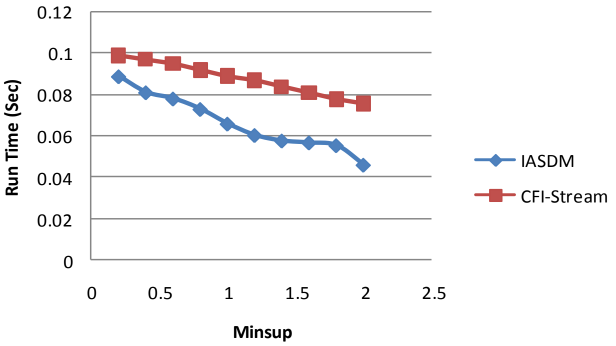


Fig. 7. Data Base Size=10,000, α =2.

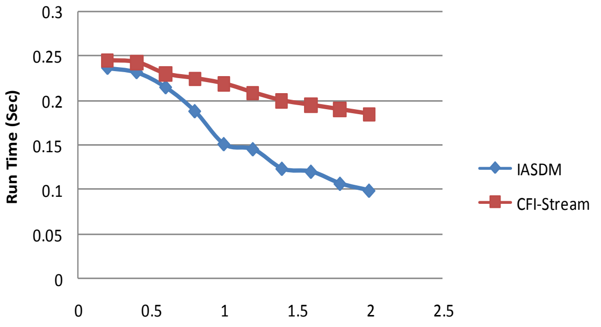


Fig. 8. Data Base Size=50,000, α =2

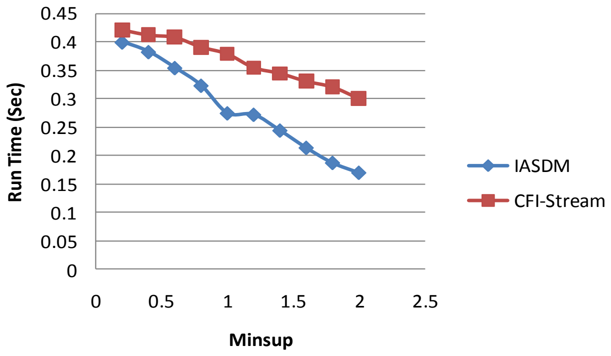


Fig. 9. Data Base Size=100,000, α =2

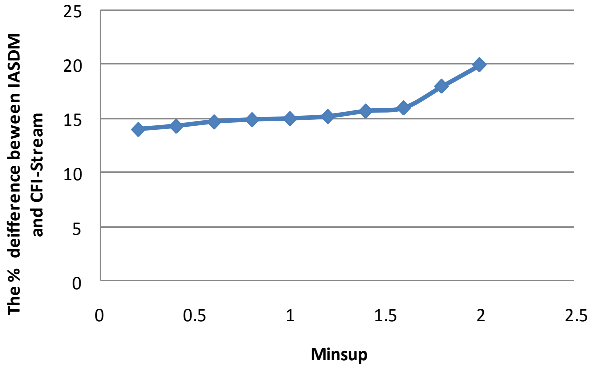


Fig. 10. Data Base Size=50,000, α =1.5

6. Conclusions

In this paper, we have proposed a novel stream data mining technique based on the association mining approach, along with a suitable structure, for generating continuous or transient rules. The proposed technique collects knowledge through sliding windows, where computations are done in an incremental fashion. A hierarchical structure has been designed for storing generated transient patterns. Two classes of closed itemsets have been formed. The first class represents frequent closed itemsets that have supports exceeding a certain threshold. The other class is used to keep the history of promising frequent closed itemsets that could be turned into frequent closed itemsets. The decision of whether an infrequent closed itemset *X* is promising or not is based on the value of some parameter α. The proposed approach favors those itemsets that are frequent closed in the current window. The proposed technique works in real time, the way the CFI [7] and CARM [8] work. The experimental results have shown that the performance of the IASDM technique against the CFI-Stream technique, which is considered as one of the latest and accredited techniques in the area of stream data mining, has improved the runtime needed for mining stream data by a factor close to 60%.

References

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules;

Int'l Conf. on Very Large Databases", September (1994).

[2] J.H. Chang, W.S. Lee and A. Zhou, "Finding Recent Frequent Itemsets Adaptively Over Online Data Streams", ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, August (2003).

[3] J.H. Chang and W.S. Lee, "A Sliding Window Method For Finding Recently Frequent Itemsets Over Online Data Streams", Journal of Information Science and Engineering; July (2004).

[4] Y. Chi, H. Wang, P.S. Yu and R.R. Muntz; Moment: Maintaining Closed Frequent Itemsets Over A Stream Sliding Window; Int'l Conf. on Data Mining; November (2004).

[5] C. Giannella, J. Han, J. Pei, X. Yan and P.S. Yu, "Mining Frequent Patterns in Data Streams at Multiple Time Granularities", Data Mining: Next Generation Challenges and Future Directions, AAAI/MIT (2003).

[6] S. Guha, N. Koudas and K. Shim, "Data Streams and Histograms" ACM Symposium on Theory of Computing (2001).

[7] N. Jiang and L. Gruenwald, "CFI-Stream: Mining Closed Frequent Itemsets in Data Streams," ACM International Conference on Knowledge and Data Discovery (KDD), August (2006).

[8] N. Jiang and L. Gruenwald, "Estimating Missing Data in Data Streams," the 12th International Conference on Database Systems for Advanced Applications, April (2007).

[9] H. Li, S. Lee and M. Shan, "An Efficient Algorithm for Mining Frequent Itemsets Over the Entire History of Data Streams", Int'l Workshop on Knowledge Discovery in Data Streams, Sept. (2004).

[10] C. Lin, D. Chiu, Y. Wu and A.L.P. Chen; Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window, SIAM Int'l Conf. on Data Mining; April (2005).

[11] C. Lucchese, S. Orlando and R. Perego, "Fast and Memory Efficient Mining of Frequent Closed Itemsets" Knowledge and Data Engineering, IEEE Transactions, January (2006)

[12] G.S. Manku and Motwani, "Approximate Frequency Counts Over Data Streams", Int'l Conf. on Very Large Databases (2002).

[13] J. Pei, J. Han, and R. Mao, "Closet: An Efficient Algorithm for Mining Frequent Closed Itemsets", ACM SIGMOD International Workshop on Data Mining and Knowledge Discovery, May (2000).

[14] J. Pei, J. Han and J. Wang, "Closet+: Searching for the Best Strategies for Mining Frequent Closed Itemsets", ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, August (2003).

[15] M.J. Zaki and C.J. Hsiao, "Charm: An Efficient Algorithm for Closed Itemsets Mining", SIAM Int'l Conf. on Data Mining; April (2002).

Received January 3, 2009
Accepted March 22, 2009