International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-7, December 2013

Mining Regular Pattern Over Dynamic Data Stream Using Bit Stream Sequence

Vijay Kumar Verma, Kanak Saxena

Abstract :-In recent years, data streams have become an increasingly important area of research for the computer science, database and data mining communities. Data streams are ordered and potentially unbounded sequences of data points created by a typically nonstationary generation process. Common data mining tasks associated with data streams include clustering, classification and frequent pattern mining[1]. Recently, temporal regularity in occurrence behavior of a pattern was treated as an emerging area in several applications A pattern is said to be regular in a data stream, if its occurrence behavior is not more than the user given regularity threshold. Although there has been some efforts done in finding regular patterns over stream data. In this paper we develop a new method called Mining regular Pattern using Bit Stream Sequence with sliding window to generate the complete set of regular pattern over a data stream at a user given regularity threshold. Experimental results show that highly efficiency in terms of execution and memory consumption and also in number of candidate scan

Keywords: Data mining, Data stream, pattern mining, regular pattern, temporal regularity. Sliding window.

I. INTRODUCTION

A data stream is a continuous, unbounded, and timely ordered sequence of data elements generated at a rapid rate. Unlike traditional static databases, stream data, in general, has additional processing requirements; i.e., each data element should be examined at most once and processed as fast as possible with the limitation of available memory[2,3]. Even though mining user-interest based patterns from data stream has become a challenging issue, interests in online stream mining for discovering such patterns dramatically increased. Therefore, a pattern is called frequent if its occurrence frequency (i.e., support) in the database exceeds the user-given support threshold. However, the occurrence frequency may not always represent the significance of a pattern. The other important criterion for identifying the interestingness of a pattern might be the shape of occurrence i.e., whether the pattern occurs periodically, irregularly, or mostly in a specific time interval.Traditional frequent pattern mining techniques fail to uncover such regular patterns because they focus only on the high frequency patterns. Patterns with temporal regularity can be discovered in a wide range of applications where users might be interested on the occurrence behavior (regularity) of patterns rather than just the occurring frequency. For example, in a retail chain data, some products may be sold more regularly than other products.

Manuscript published on 30 December 2013.
*****Correspondence Author(s)
Vijay Kumar Verma, Dept of Computer. Science and Engg. Lord Krishna College of Technology Indore M.P., India.
Dr. Kanak Saxena, Professor and Head Dept of Comp. Application Samrat Ashok Technological Institute Vidisha M.P., India.

Thus, even though both of the products are sold frequently over the entire selling history or for a specific time period (e.g., for a year), the products still need to be managed independently. That is, it is necessary to identify a set of items that are sold together at a regular interval for a specified time period.[4,5]

II. BASIC CONCEPTS

Let I = {i1, i2, . . . , in} be the set of items. A set X = {ij, . . . ,….. ik} □ I, where j ≤ k and j, k □ [1, n] is called a pattern (or an itemset). A transaction
t = (tid, Y) is a couple where tid is a transaction-id and Y is a patter or an itemset. If X□□ Y, which means that t contains X or X occurs in t. Let size(t)
be the size of t, i.e., the number of items in Y.[6,7]

2.1 Data Stream

A data stream DS can be defined as infinite sequence of transactions, i.e., DS = [t1, t2, …, tm, …], i□□ [1, m] where ti is the ith arrived transaction. A window W can be referred to as a set of all transactions between the ith and jth arrival of transactions, where j > i and the size of W is |W|= j – i, i.e., the number of transactions between ith and jth arrival of transactions. Let each slide of window introduce and expire slide_size, 1 ≤ slide_size ≥|W|, transactions into and from the current window. If X occurs in tj, j □ [1, |W|], such transactions-id is denoted as tj X, j □ [1, |W|]. Therefore TwX = { tjX, . . . tkX }, j, k□□ [1, |W|] and j ≤ k is the set of all transaction-ids where X occurs in the current window W

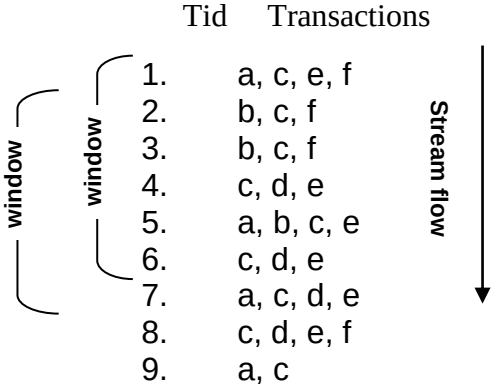

Table 1 Transactional Database

Where Window size |W| =8
For example figure 1 represents a data stream which contains transaction-id i.e., tid, and itemset i.e., transaction with respect to tid. Now consider the window size may be 8 i.e., the size of the window |W| = 8.

Let the first window W1 handles the transactions of data stream from tid-1 to tid-8. Similarly windows W2 contain next 8 transactions. We want to find out the periods for each itemset which are given in the data stream to get the regular patterns which are less than or equal to the user given regularity threshold. After finding W1 regular patterns generate second window W2 and repeat the same procedure to find out latest regular patterns from the data stream

2.2 A period of X in W

Let tX j+1 and tj X , j □ [1, (|W|-1)], be two consecutive transaction-ids in Tw X. the number of transactions between tX j+1 and tj X is defined as a period of X, say pX (i.e., pX = tX j+1 – tj X, j□□ [1, (|W|-1)]). For the simplicity of period computation, a "null" transaction with no item is considered at the beginning of W, i.e., tf = 0 (null), where tf represents the tid of the first transaction to be considered. Similarly, tl, the tid of the last transaction to be considered, is the tid of the |W|th transaction in the window, i.e., tl = t |W|. For instance, the stream data in table 1, consider the window is composed of eight transactions (i.e., tid = 1 to tid = 8 make the first window, say W1). Then set of transactions in W1 where pattern (b, c) appears in (2, 3, 5). Therefore, the periods for (b, c) are {(2 – tf) = 2, (3 – 2) = 1, (5 – 3) = 2 and (tl – 5) = 3}, where tf = 0 and tl = 8.[7,8,]

2.3 Regularity of a pattern X in W

Let in a TwX, PwX be the set of all periods of X in W i.e., PwX = { p1X . . . psX}, where s is the total number of periods of X in W. Then, the regularity of X in W can be denoted as regw(X) = Max (p1X, ..., psX}. For example, in DS of table 1 regw (b, c) = 3, since Pw1 {b, c} = Max (2, 1, 2, 3) = 3. Therefore a pattern is called a regular pattern in W if its regularity in W must not more than a user given maximum regularity threshold called max_reg λ, with $1 \leq \lambda \leq |W|$. The regularity threshold is given as the percentage of window size. Therefore the regular patterns in W satisfy the downward closure property. i.e., if a pattern is found to be regular, then all of its non-empty subsets will be regular. Accordingly, if a pattern is not regular, then none of its supersets can be regular. Given DS, |W|, and max_reg, finding the complete set of regular patterns in W, Rw that have regularity of not greater than the max_reg value is the problem of mining regular patterns in data stream.[10,12]

III. RELATED WORK

In data mining, one of the most important techniques is Association rule mining. It was first introduced by Agarwal et al. It extracts frequent patterns, correlations, associations among sets of items in databases. The main drawback with the classical Apriori algorithm is that it needs repeated scans to generate candidate set. After that Frequent pattern tree and FP-growth algorithm is introduced by Han et al. to mine frequent patterns without candidate generation.

In 2005 M.G. Elfeky, W.G. Aref, A.K. Elmagarmid proposed "Periodicity detection in time series databases are also closely related with Regular patterns. Periodic pattern mining in time-series data focuses on the cyclic behavior of patterns either in whole or some part of time-series. Although periodic pattern mining is closely related with our work, it cannot be applied directly to mine regular patterns from a data stream because it process with either time-series or sequential data[11,13].

In 2006 S.K. Tanbeer, C.F. Ahmed, B.-S. Jeong, Y.-K. Lee "Sliding Window-based Frequent Pattern Mining over Data Streams. Information Sciences". [14]

In 2008 Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, Young-Koo Lee proposed "Efficient Frequent Pattern Mining over Data Streams " .They introduce prefix-tree structure CPS-tree which uses dynamic tree restructuring mechanism in data stream and efficiently finds recent frequent patterns from high-speed data stream with a single-pass.[15,16]

In 2009 Hua-Fu Li *, Suh-Yin Lee proposed " Mining frequent itemsets over data streams using efficient window sliding techniques" In this paper, they propose an efficient single-pass algorithm, called MFI-TransSW, for mining the set of frequent itemsets over data streams with a transaction-sensitive sliding window. An effective representation of items is developed to enhance the performance [16]

In 2010 Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, and Byeong-Soo Jeong proposed "Mining Regular Patterns in Data Streams" new concept of mining interesting patterns (called regular patterns) that occur with a temporal regularity in high-speed data streams. they proposed a novel tree structure, RPS-tree, to capture the stream content in memory efficient manner and to enable regular pattern mining from it. To obtain the fast and interesting results RPS-tree can be updated efficiently for the current content of the stream.[15]

IV. PROPOSED METHODS

Consider a simple transactional data base given table one suppose minimum support threshold is. We transform of transactions into bit stream sequence. Now we generate frequent pattern for CSW1 which consists of eight transactions form T1 to T8, transforming Bit stream sequences are

Table 2(with period of item and regularity value)

One Itemset	Bit Stream sequence	A period of itemset X	**R**
a	10001010	1,4,2	4
b	01101000	2,1,2,3	3
c	11111111	1,1,1,1,1,1,1,1	1
d	00010111	4,2,1,1	4
e	10011111	1,3,1,1,1,1	3
f	11100001	1,1,1,5	5

Suppose maximum regularity threshold called max_reg λ is 3. Items {b, c, e} are the only item which are regular and item a,d,f are not regular because they have the regularity value greater than the given regularity threshold value.

Table 3(with period of item and regularity value)

two Itemset	Bit Stream sequence	A period of itemset X	R
(b, c)	01101000	2,1,1,3	3
(b, e)	00001000	5	5
(c ,e)	10011111	1,3,1,1,1,1	3

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

For two itemset we can see from table 3 only (b, c) and (c,e) are regular item set because they have the less regularity threshold value with the given minimum regularity threshold. For three item set we perform logical AND operation on regular two item set .

Three item set	Bit stream sequence	A period of itemset X	R
(b,c,e)	00001000	5,3	5

Table 4 (with period of item and regularity value)

So it is clear that in window1 the regular pattern are { (b), (c), (e), (b , c), (c, e).

In window2 there are eight transactions from T2 to T9. We have to consider only the item of transaction T9.We are using left shift bit operation

To add bit 1 to the bit stream sequence of the an existing item otherwise we add 0

One Itemset	Bit Stream sequence	A period of itemset X	R
a	00010101	4,2,2	4
b	11010000	1,1,2,4	4
c	11111111	1,1,1,1,1,1,1,1	1
d	00101110	3,2,1,1,1	3
e	00111110	3,1,1,1,1,1	3
f	11000010	1,1,5,1	5

Table 5 (with period of item and regularity value)

Form table 5 it is clear that the item c, d, e are only one regular item set in window1 because they have regularity threshold less than the given minimum regularity threshold value

Itemset	Bit Stream sequence	A period of itemset X	R
c	11111111	1,1,1,1,1,1,1,1	1
d	00101110	3,2,1,1,1	3
e	00111110	3,1,1,1,1,1	3

Table 6 (with period of item and regularity value)

For three regular itemset we perform logical AND operation on two regular itemset.

Itemset	Bit Stream sequence	A period of itemset X	R
c, d	00101110	3,2,1,1,1	3
c, e	00111110	3,4,5,6,7	7
d, e	00101110	3,2,1,1,1	3

Table 7 (with period of item and regularity value)

Form table 7 it is clear that itemset(c, d) and (d, e) are regular item set in window 2 because they have regularity threshold value less the are equal to the given minimum regularity threshold value.

Now we can find out three regular item set from two regular item set by performing logical AND operation.

Table 8 (with period of item and regularity value)

Itemset	Bit Stream sequence	A period of itemset X	R
c, d ,e	101110	3,2,1,1,1	3

So regular pattern in window 2 are
{ (c) , (d), (e),(c, d), (c, e),(c, d, e)}

5. PROPOSED ALGORITHMS

Input: DS, minimum regularity threshold
Output: Complete set of regular patterns
Procedure:
1. For each window W of size in DS
2. Convert Tw into bit stream sequence
3. For each item i in X where X Í Tw
4. If (Find R(i) > minimum regularity threshold) 5. Delete i itemsets
6. Else
7. {
8. Result ← Result È(i)
9. For each item inext in X
10. {
11. If(Find R (inext) > minimum regularity threshold)
12 }
13 Delete inext
14 Else
15 Result ← Result U (i, next)
16 Do “AND ” operation till all regular i itemsets found
17 }
18 update window w(i, j)

VI. PERFORMANCE EVALUATIONS

The proposed algorithm is applied to the retail data set available in frequent itemset mining. The data set contains the retail market basket data from an anonymous Apurti Super market store dataset repository. Data collected is over approximately 5 months duration. The total amount of receipts being collected equals 10,000 The elements of the retail data set are looked up in sequence to simulate the environment of a data stream. The algorithm is implemented in on a 3i GHz Pentium(R) PC machine with 2 GB memory running on Windows XP. The initial execution starts with sliding window size being set equal to 100 .The value for the minimum support threshold is set to 0.3. The following graph in figure 3 illustrates the variations made to the sliding window size based on the adaptation factor. A higher value for the adaptation factor indicates greater consumption of computational resources as such a reduction will be made to the sliding window size. Conversely the sliding window size is increased for a lower value of adaptation factor.

VII. CONCLUSIONS

In this paper, we propose an efficient single-pass algorithm, for mining the regular pattern data dynamic data streams by efficiently removed aged transaction from current sliding window.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

We use an effective bit-stream sequence representation of items to enhance the performance of proposed algorithm. The proposed method is also efficient in term of memory problem of finding frequent items in a continuous stream of items. Proposed algorithm we proposed algorithm is efficient in term of execution time number of candidate generation and memory as compared to previous algorithms. In future we can enhance this algorithm to find out time interval pattern temporal pattern.

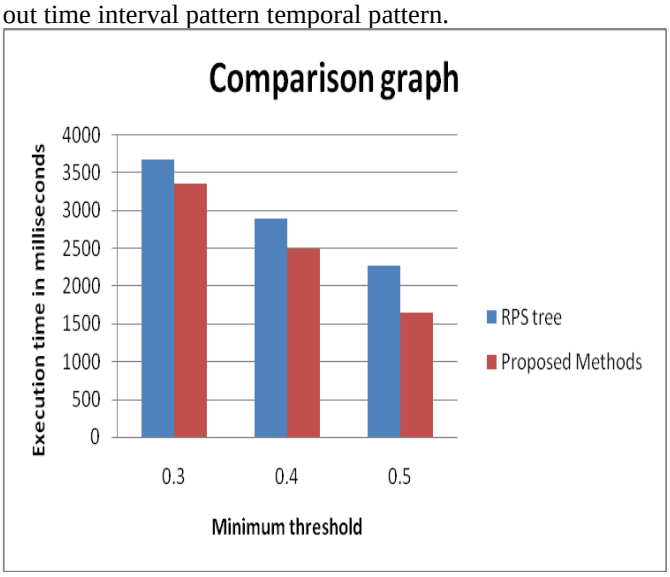

Figure1 comparison between RPS tree and proposed method

From the graph it is clear that proposed methods is more efficient in term of execution time as compared to RPS tree. More threshold need less execution time.

REFERENCES

1. S.K. Tanbeer, C.F. Ahmed, B.-S. Jeong, Y.-K. Lee "Sliding Window-based Frequent Pattern Mining over Data Streams. Information Sciences", 179, 2006, pp. 3843-3865
2. C.K.-S. Leung, , Q.I. Khan "DSTree: A Tree Structure for the mining of Frequent Sets from Data Streams." In: ICDM, 2006, pp. 928-932.
3. H.-F. Li, S.-Y. Lee "Mining Frequent Itemsets over Data Streams Using Efficient Window Sliding Techniques." Expert Systems with Applications 36, 2009, pp. 1466-1477
4. J. Han, J. Pie, Y. Yin "Mining Frequent Patterns without candidate generation", In Proc. ACM SIGMOD international Conference on management of Data, 2000, pp. 1-12.
5. R. Agarwal, and R. Srikanth, "Fast algorithms for mining association rules in Large databases", In Proc. 1994 Int. Conf. Very Large Databases VLDBA'94, Santiago, Chile, Sept. 1994, pp. 487- 499.
6. S. K. Tanbeer, C. F. Ahmed, B.S. Jeong, and Y.K. Lee, "Mining Regular Patterns in Transactional Databases", IEICE Trans. On Information Systems, E91-D, 11, 2008, pp. 2568- 2577.
7. S.K. Tanbeer, C.F. Ahmed, B.S. Jeong. "Mining regular patterns in data streams." In: DASFAA. Volume 5981 of LNCS., Springer 2010, pp. 399-413
8. J. Han, M. Kamber, "Data Mining: Concepts and Techniques", 2nd ed. An Imprint of Elsevier, Morgan Kaufmann publishers, 2006, pp. 468-489.
9. G. Yi-ming, W. Zhi-jun, "A Vertical format algorithm for mining frequent item sets", IEEE Transactions, pp. 11-13, 2010
10. M. J. Zaki, K. Gouda. "Fast Vertical Mining using Diffsets", SIGKDD '03, Copyright 2003 ACM 1-58113-737-0/03/0008, August' 24 – 27, 2003
11. G. Vijay Kumar, M. Sreedevi, NVS. Pavan Kumar. "Mining Regular Patterns in Transactional Databases using vertical Format", International Journal of Advanced Research in Computer Science, vol. 2, pp. 581-583, Sep-Oct 2011.
12. M.G. Elfeky, W.G. Aref, A.K. Elmagarmid "Periodicity detection in time series databases." IEEE Transactions on Knowledge and Data Engineering 17(7), pp. 875-887 2005
13. G. Lee, W. Yang, J-M Lee. "A Parallel algorithm for mining partial periodic patterns." Information Society 176, pp. 2006, pp.3591-3609
14. B. Ozden, S. Ramaswamy, A. Silberschatz. "Cyclic Association Rules." In.: 14th International conference on Data Engineering, 1998, pp. 412-421.
15. Frequent Itemset Mining Dataset Repository http://fimi.cs.helsinki.fi/data/ and UCI machine learning repository (University or California).
16. Agrawal, R., Srikant, R.: Fast algorithms for Mining Association Rules in Large Databases. In: VLDB, pp. 487–499 (1994)

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)