Mining Weighted Rare Association Rules Using Sliding Window over Data Streams

Weimin Ouyang
Department of Computer Teaching
Shanghai University of Political Science and Law
Shanghai, China
oywm@shupl.edu.cn

***Abstract*—Rare association rules mining is an association rule which has low support and high confidence. In recent years, the problem of mining rare association rules has got quite a lot of attention, which has become a hot topic in data mining research. However, most of the research on mining rare association rules are confined to the static database environment, and treat each item with the same significance although different items may have different significance. In this paper, we propose an algorithm for mining weighted rare association rules over data streams with a sliding window. Experiments on the synthetic data stream show that the proposed algorithm is efficient and scalable.**

Keywords—rare association rules; weighted rare association rules; data streams; sliding window

I. INTRODUCTION

A data stream is an ordered sequence of elements which arrives one by one with positive real time intervals [1]. It is often refer to as streaming data. Different from data in traditional static datasets, a data stream is continuous, huge, fast changing, rapid and infinite. Many applications generate large amount of data streams in real time, such as sensor data generated from sensor networks, online transaction flows in retail chains, Web log and click-streams in Web applications, call records in telecommunications, etc [2].

Association rules mining is a most of important tasks in data mining research. A number of algorithms have been proposed to improve the running time for generating frequent itemsets and association rules since the problem was pointed out by R.Agrawal in 1993[3].

With the further research on the mining of frequent itemsets, it has been recognized that some infrequent itemsets can provide very useful insight view into the data set[4], and a new kind of knowledge discovery problems called as rare association rules has been proposed [5,6,7,8,9,10]. While association rules are discovered from frequent itemsets, rare association rules are discovered from rare itemsets. Frequent itemset reveals the information about the items which occurs frequently, and rare itemset unfolds the information about the items which occurs infrequently.

Rare association rules mining is an association rule which has low support and high confidence. In recent years, the problem of mining rare association rules has got quite a lot of attention, which has become a hot topic in data mining research. However, most of the research on mining rare association rules are confined to the static database environment, and treat each item with the same significance although different item may have different significance. In this paper, we propose an algorithm for mining weighted rare association rules over data streams with a sliding window.

The paper is organized as follows. The related work is described in section 2. The definitions for weighted rare association rules in data stream with a sliding window are given in section 3. In section 4, we describe the algorithm to find weighted rare association rules in data stream with sliding window, and Section 5 presents our experimental results. The conclusion and future works are made in the last section.

II. RELATED WORK

Recently, mining rare association rules has attracted numerous researches. However, most of all the current algorithms for mining rare association rules is designed for static database and can not treat each item with different significance. There are two different types of rare association rules mining approaches: level-wise and tree based. Current rare itemsets mining approaches which are based on level-wise exploration of the search space are similar to the Apriori algorithm.

MS-Apriori [5], Apriori-Inverse [6], Rarity [7], ARIMA [8] and AfRIM [9] are five algorithms that discover rare itemsets. They all use level-wise algorithm similar to Apriori, which has potentially expensive candidate generation and pruning steps. In addition, these algorithms attempt to identify all possible rare itemsets, and require a significant amount of execution time. Tsang et al. [10] proposed a RP-Tree algorithm to handle these issues. RP-Tree avoids the expensive itemset generation and pruning steps by using a tree data structure to find rare patterns. However RP-Tree algorithm still uses a multi-pass approach, which is not suitable in data stream environment. Up until now, to our best knowledge, there has been no research on rare pattern mining in data streams.

III. PROBLEM DEFINITIONS

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. A transaction $T = (tid, x_1, x_2, \ldots, x_n)$, $x_i \in I$, for $1 \le i \le n$, is a subset of I, while n is called the size of the transaction, and tid is the unique identifier of the transaction. A non-empty subset of I is called itemset. An itemset containing k items is called k-itemset.

A. Weighted Association Rule

In order to express the significance of items, a weight value is assigned for each item x denoted as ω(x). We arrange itemset according to weight of items in descending order. Let X be an itemset $\{x_1, x_2, \ldots, x_k\}$, where $w(x_1) \geq w(x_2) \geq \ldots \geq w(x_k)$. Using this itemset arrangement, we can keep the downward closure property hold which will be proved late.

The weighted support of an itemset $X = \{ x_1, x_2, \ldots, x_k \}$ in which $w(x_1) \geq w(x_2) \geq \ldots \geq w(x_k)$ is defined as $w\sup(X) = \max_{i=1}^{k} w(x_i) \times \sup(X)$ .The downward closure property of frequent itemset will hold true.

Let $X = \{ x_1, x_2, \ldots, x_{k-1} \}$ and $Y = \{ y_1, y_2, \ldots, y_{k-1}\}$, k>=2. if $x_i=y_i$, i =1,2,3,…, k-2, and $w(x_{k-1})>w(y_{k-1})$, the X and Y is called as joinable.

If $X = \{ x_1, x_2, \ldots, x_{k-1} \}$ and $Y = \{ y_1, y_2, \ldots, y_{k-1}\}$(k>=2) is joinable denoted as $X \bullet Y = \{ x_1, x_2, \ldots, x_{k-1}, y_{k-1}\}$.

Theorem 1: Let itemset $X = \{ x_1, x_2, \ldots, x_k \}$ and $Y = \{ x_1, x_2, \ldots, x_{k-2}, x_{k-1} \}$, $Z = \{ x_1, x_2, \ldots, x_{k-2}, x_k \}$ and $w(x_1) \geq w(x_2) \geq \ldots \geq w(x_k)$. If X is a weighted frequent itemset, then Y and Z will be both weighted frequent itemset.

Theorem 2: For k>2, any weighted frequent k-itemset X can be obtained by joining two weighted frequent (k-1)-itemset.

Limited to the paper length, the proof for the above two theorems have been omitted here.

Definition 1: Given a transaction database, and a user-defined minimum support threshold *s*, The form (X, Y) is a weighted association rule, if and only if $wsup(X \cup Y) \geq s$ and $wconf(X, Y) = wsup(X \cup Y) / wsup(X) \geq c$.

B. Weighted Rare Association Rule in Data Stream

Definition 2: A transaction data stream TDS = $T_1, T_2, \ldots, T_N$ is a continuous sequence of transactions, where N is the tid of latest incoming transaction T_N.

A transaction-sensitive sliding window in the transaction data stream is a window that slides forward for every transaction. The window at each slide has a fixed number, w, of transactions, and w is called the size of the window. The current transaction-sensitive sliding window is denoted as $TransSW_{N-w+1} = [T_{N-w+1}, T_{N-w+2}, \ldots, T_N]$, where $N-w+1$ is the index of the first transaction of the current window SW.

Definition 3: The **support** of an itemset *X* in *SW*, denoted as $\mathbf{sup}(X)^{SW}$, is the number of transactions in *SW* containing *X* as a *subset*.

Definition 4: The **weighted support** of an itemset *X* in *SW*, denoted as $\mathbf{wsup}(X)^{SW}$, which is defined as follows:

$$w\sup(X)^{SW} = \max_{i=1}^{k} w(x_i) \times \sup(X)^{SW}$$

Definition 5: An itemset *X* is called a **frequent** if $\mathbf{wsup}(X)^{SW} \geq s \cdot w$, where *s* is a user-defined minimum support threshold, c is a user-defined minimum confidence. The value $s \cdot w$ is called the **frequent threshold** of *SW*.

Definition 6: Given a transaction-sensitive sliding window *SW*, and a user-defined maximum support threshold maxs, a user-defined minimum support threshold mins, and is a user-defined minimum confidence c, the rule X->Y is a weighted rare association rules in window *SW*, if and only if and only if $wsup(X \cup Y)^{SW} < maxs \cdot w$, $wsup(X \cup Y)^{SW} \geq mins \cdot w$ and $wconf(X, Y)^{SW} \geq c$.

IV. Mining Weighted Rare Association Rules in Data Stream with Sliding Window

According to the definitions of rare association rules in last section, we propose an algorithm to discover rare association rules in data stream called MWRAR-SW (Mining weighted rare Association Rules in a Sliding Window). In the proposed algorithm, for each item X in the current sliding window SW, we construct a bit-sequence with w bits denoted as Bit(X). If an item X is in i-th transaction of the current window SW, i-th bit of Bit(X) is set to be 1; otherwise, it is set to be 0. The process is called bit-sequence transform.

For example, an example is shown in Table 1, and window size is 4.

TABLE 1: DATA STREAM TRANSACTION AND WEIGHT VALUES

Tid	items	item	weight
Tid1	bd	a	0.8
Tid2	bcd	b	0.4
Tid3	be	c	0.2
Tid4	bde	d	0.6
Tid5	abd	e	0.3

The first sliding window SW1 consists of four transactions: <Tid1, (bd)>, <Tid2, (bcd) >, <Tid3, (be) > and <Tid4, (bde)>. Because item a does not appear in any transaction of window SW1, the bit-sequence of a, Bit(a), is 0000. Similarly, Bit(b) = 1111, Bit(c) = 0100, Bit(d) = 1101, and Bit(e) = 0011.

The algorithm MWRAR-SW is described as follows:

Algorithm MWRAR-SW

Input: TDS (a transaction data stream), maximum support threshold: maxs ; minimum support threshold: mins ; minimum confidence threshold: c; the user-specified sliding window size w.

Output: Set of rare association rules: WRAR ;

Begin

SW = Null; /* Window SW consists of w transactions */

Repeat:

For each incoming transaction Ti in SW do

If SW = Full then

Do bitwise-shift on bit-sequences of all items in SW;

Else

For each item X in Ti do

Do bit-sequence transform(X);

EndIf;

For each bit-sequence Bit(X) in SW do

If sup(X) = 0 then Drop X from SW;

EndRepeat

/* the following is the rare itemsets generation phase. */

SR= ∅;

R1 = {rare 1-itemsets};

For (k=2; $R_{k-1} \neq$ Null; k++) do {

Ck = Candidate_Gen(R_{k-1});

Do bitwise AND to find the supports of C_k;

R_k = { c ∈C_k | wsup(c)^{SW} <maxs·w ∧ wsup(c)^{SW} ≥mins·w };

SR = SR ∪ R_k;

}

/* the following is the rare association rules generation phase. */

WRAR = ∅;

For each itemset i in SR Do {

For any X∪Y=i and X ∩Y = ∅ Do {

If wconf(X->Y) ^{SW}≧c

Then WRAR = WRAR ∪ { X->Y };

}

}

End

The proposed MWRAR-SW algorithm consists of four phases, window initialization phase, window sliding phase, and weighted rare itemsets generation phase and rare association rules generation phase.

(1) Window Initialization Phase

The phase is processed when the number of transactions come into the current window so far is less than or equal to a user-predefined sliding window size w. In this phase, each item in the new incoming transaction is transformed into its bit-sequence representation. Before this phase, for each item X in I, the bit-sequence Bit(X) is initialized with 0.

For example, in Table 1, the first sliding window SW1 contains four transactions: Tid1, Tid2, Tid3 and Tid4. The bit-sequences of items of SW1 in the window initialization phase are shown in Table 2.

TABLE 2: BIT-SEQUENCES OF ITEMS OF SW1

Tid	Items	bit-sequence transformation in SW1
Tid1	(*bd*)	Bit(a)=000, Bit(b)=1000, Bit(c)=0000,Bit(d)=1000,Bit(e)=0000
Tid2	(*bcd*)	Bit(a)=0000, Bit(b)=1100, Bit(c)=0100,Bit(d)=1100,Bit(e)=0000
Tid3	(be)	Bit(a)=0000, Bit(b)=1110, Bit(c)=0100,Bit(d)=1100,Bit(e)=0010
Tid4	(bde)	Bit(a)=0000, Bit(b)=1111, Bit(c)=0100,Bit(d)=1101,Bit(e)=0011

(2) Window sliding phase

The phase is activated after the number of transactions in the sliding window SW is w. Before a new incoming transaction is appended to the sliding window, the oldest transaction is removed from the window.

For removing the oldest transaction, a simple method is used in the proposed algorithm. Since the MWRAR-SW algorithm use bit-sequence representation, we can uses the bitwise left shift operation to remove the oldest transaction from the current sliding window.

After sliding the window phase, an effective pruning method, called Item-Prune, is used to improve the memory usage. The pruning method is that an item X in the current sliding window is dropped if and only if wsup(X)^{SW} = 0.

For example, in Figure 1, before the fifth transaction <Tid5, (abd)> is processed, the first transaction Tid1 must be removed from the current window using bitwise left shift on the set of items. Hence, Bit(a) is modified from 0000 to 0001. Similarly, Bit(b)= 1111,Bit(c)= 1000, Bit(d)= 1011, and Bit(e)= 0110. Then, the new transaction <T5, (abd)> is processed by bit-sequence transform. The result is shown in Table 3.

TABLE 3: BIT-SEQUENCES OF ITEMS IN WINDOW SLIDING PHASE OF SW2

Window-id	Transactions	Bit-Sequences of items
SW_2	<Tid2, (bcd) >	Bit(a)=1000, Bit(b)=1111 Bit(c)=1000,Bit(d)=1011 Bit(e)=0110
	<*Tid3*, (*be*) >	
	<*Tid4, (bde)*>	
	<*Tid5, (abd)*>	

(3) Weighted rare itemsets generation phase

In this phase, MWRAR-SW algorithm uses a level-wise method to generate the set of candidate itemsets C_k from the frequent itemsets R_{k-1} according to the Apriori. The step is called Candidate_Gen. Then, the proposed algorithm uses the bitwise AND operation to count the support of these candidates to find the rare k-itemsets R_k. The process is stopped until no new candidates are generated.

For instance, consider the bit-sequences of SW_2 in Figure 3, and let maximum support threshold maxs to be 0.8, the minimum support threshold mins to be 0.2. Hence, an itemset X is rare if wsup(X)^{SW} <0.8*4 = 3.2 and wsup(X)^{SW} ≥ 0.2*4 = 0.8. In the following, we describe the step of rare itemset generation of $TransSW_2$.

Firstly, MWRAR-SW algorithm find out rare 1-itemset R_1={(d),(e)},which is shown as Table 4, then generates three candidate 2-itemsets C_2={(de)} by combining rare 1-itemsets R_1={(d),(e)}, which is shown as Table 5.

After using bitwise AND operations to count the supports of these candidates, R_2={(de)}, which is shown as Table 6, because the Bit(de) =0001, sup(de) = 1.

TABLE 4: RARE 1-ITEMSET IN SW2

Window-id	R_1	bit-sequence	support	weight
SW_2	a	1000	1	0.8
	d	1101	3	0.6

TABLE 5: CANDIDATE RARE 2-ITEMSET

Window-id	C_2	bit-sequence	support	weight
SW_2	ad	1000	1	0.8

TABLE 6: RARE 2-ITEMSET IN SW2

Window-id	R_2	bit-sequence	support	weight
SW_2	ad	1000	1	0.8

(4) Weighted rare association rules generation phase

In this phase, MWRAR-SW generates weighted rare association rules as follows. For each weighted rare itemset R, choose any two subsets X and Y of the itemset, which satisfy $X \in R$ and $Y \in R$ and $X \cap Y = \phi$. For example, consider rare 2-itemset (ad). The possible rare rules with the itemset are a->d and d->a. To check the interestingness of these rules, compare it with the minimum confidence threshold c. A rule is interesting if confidence of the rule is greater than the 0.8.

V. EXPERIMENT

In this section, we evaluate the performance of our proposed algorithm for mining indirect temporal sequential patterns. The computation environments are i5-3470, 4G RAM, Windows 7 operating system. The algorithm is implemented with C++. The synthetic experiment data set is generated by Assocgen [4].

The synthetic data stream, denoted as T5I4D1000K, of size 1 million transactions (D1000K) has an average transaction size of 5 items (T5) with average maximal frequent itemset size of 4 items (I4). In the experiments, the transactions of T5I4D1000K are looked up in sequence to simulate the environment of an online data stream.

Figure 1 shows the processing time of window initialization phase under different window sizes from 20,000 (200K) transactions to 100,000 (1,000K) transactions. Figure 2 shows the total time of window sliding time and weighted association rules mining time at each 100K transactions using various window sizes from 200K transactions to 1000K transactions. As shown in Figure 1 and 2, MWAR-SW algorithm is efficient and scalable.

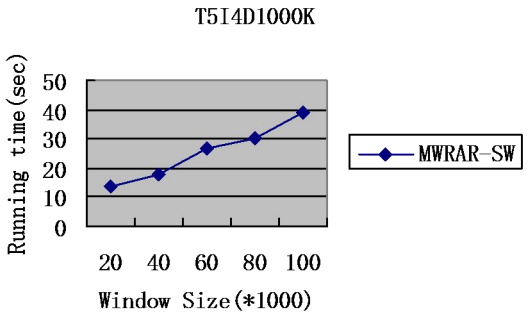

Figure 1: Running time in window initialization phases of algorithm MWRAR-SW under different window size.

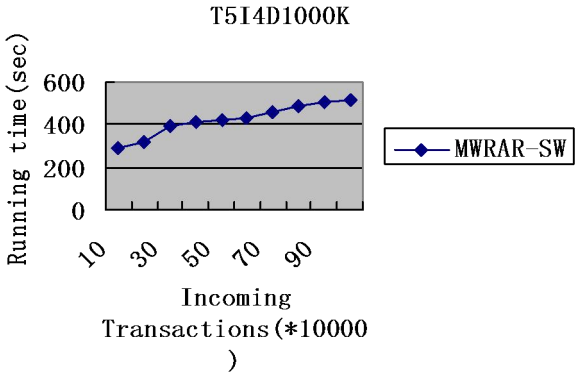

Figure 2: Running time including window sliding time and weighted association rules generation time of algorithm MWRAR-SW under different window size.

VI. CONCLUSIONS

In this paper, we proposed an efficient one-pass algorithm, called MWRAR-SW, for mining weighted association rules using sliding window over online data streams. Experiments show that the proposed algorithm is efficient and scalable.

REFERENCES

[1] N. Jiang, L. Gruenwald, "Research Issues in Data Stream Association Rule Mining", In SIGMOD Record, Vol. 35, No.1, pp.14-19, 2006.

[2] Babcock B, Babu S, Datar M, et al., "Models and issues in data stream systems", In Proc. of the 21th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.Wisconsin: Madison, 2002, pp. 1-16.

[3] Agrawal R, Swami A,Imielinski T, "Mining association rules between sets of items in large databases", In the Proc. of 1993 ACM International Conference on management of data,vol.22, pp.207-216.

[4] P.N.Tan and V.Kumar, "Indirect Association: Mining Higher Order Dependences in Data", In the Proc. Of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lyon, France, pp632-737, 2000.

[5] 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.

[6] Koh Y.S, Rountree N, "Finding Sporadic Rules Using Apriori-Inverse", In the Proc. of PAKDD 2005, vol. 3518, pp.97-106, 2005.

[7] Szathmary L, Napoli A, Valtchev P, "Towards rare itemset mining", In the Proc. of the 19th IEEE International Conference on Tools with Artificial Intelligence, vol. 1, pp. 305-312, 2007.

[8] Torino L, Sibelius G, Barolo C, "A fast algorithm for mining rare itemsets". In the Proc. of the Ninth International Conference on Intelligent Systems Design and Applications, pp.1149-1155, 2009.

[9] Adda M, Wu L, Feng Y, "Rare itemset mining", In the Proc. of the Sixth International Conference on Machine Learning and Applications, pp.73-80, 2007.

[10] Tsang S, Koh Y.S, Dobbie G, "RP-Tree: Rare Pattern Tree Mining", In the Proc. of DaWaK 2011, vol. 6862, pp. 277-288, 2011.