International Core Journal of Engineering Volume 8 Issue 2, 2022
ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

Research on Association Rules Mining for Data Stream

Qingfeng Li*, Wengfeng Peng

Hunan University of Technology and Business, Changsha, Hunan, 410205, China

Abstract

As data stream becomes common in many modern systems, data stream mining has gained its importance in recent years. Since traditional static data mining techniques is not sufficient for analyzing this type of new data, developing new mining algorithms becomes an emergent need. In this paper, we apply Frequent Tree Interpolation Method (FTIM) to mine association rules. It differs from traditional FP-TREE algorithms in two aspects. First, FTLM uses ascending instead of descending order to create frequent tree. As a result, the searching method for mining frequent itemsets evolves from the conditional search in FP-TREE to unconditional search, which saves computation time. Second, interpolation is applied to construct the frequent tree. If an existing item is included in the new input transaction, FTIM inserts the transaction into the frequent tree of this item, rather than to create a new branch. This decreases the branches of the frequent tree, and reduce the memory space required for constructing the data stream frequent tree. Experiments show that the FTIM algorithm outperforms the traditional FP-TREE algorithm in both speed and scalability.

Keywords

Data Stream; Association Rules; FTIM Algorithm; FP-TREE Algorithm.

1. Introduction

Association rules mining was first proposed by Agrawal et al. [1], and it has been one of the most basic and important issues in the field of data mining. Association rules have been widely used in additional express delivery, catalog design, up-sales, retail shelf design, and accurate positioning of potential customers based on purchase patterns. The databases in these applications are extremely large, so how to efficiently extract rules from such large databases is one of the technical difficulties of data mining. At present, many research results are based on the Apriori algorithm or its derivative algorithms. This type of algorithm can achieve better performance when a huge set of candidate items is generated. However, as we have found in the research of commercial risk management data mining, Apriori-based algorithms has a few disadvantages when the mining task has a large number of strong patterns, long patterns, or a low threshold. The disadvantages are as follows: (1) The algorithm will inevitably consume a lot of time when processing a large-scale candidate item set; (2) The algorithm must repeat canning multiple times of the database, and perform pattern matching analysis of candidate item sets, which will inevitably consume massive storage space. To address the above two problems, Han et al. [4] proposed an association rule mining algorithm FP-growth based on frequent pattern tree (FP-TREE). Theory analysis and experiments show that this algorithm is better than Apriori algorithm, but the algorithm cannot update, maintain and manage the association rules that have been mined, which severely limits the application of this algorithm.

In recent years, data streams have widely existed in many fields such as the financial industry, e-commerce network transactions, wireless sensor networks, and computer network monitoring, which has brought about a research boom in data stream mining. Since traditional static data mining technology is unable to adapt to this new data form, data mining for data streams has become an

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

urgent need in these fields. A data stream is a sequence of items that continue to appear over time. Compared with traditional static data, the data stream is continuous and potentially borderless, and usually flows in at a high speed. This has brought new challenges to data collection and data mining in terms of storage space, processor, and energy supply of the computer.

Unlike traditional static data, data flow has many new features: (1) elements in the data flow arrive online and in real time; (2) data elements arrive continuously, which cannot be controlled by the application system, and the arrival of data is unpredictable; (3) The total amount of data is huge and unlimited; the limited memory space cannot store all the data; (4) The sequence of items that cannot control the data stream comes in sequence, and these sequence of items come randomly in the form of a stream; (5) Single pass of the scanning data. Once the data is processed, there are two results: stored or discarded.

The characteristics of the data stream require that the analysis and processing of the data be instantaneous or online. The data stream mining algorithm cannot scan the database multiple times like traditional data mining. Data storage also replaces the original storage method that stores everything in the database before analysis. Instead, it requires data mining in a limited memory space to obtain knowledge or rules. Therefore, the traditional data association rule mining algorithm can no longer adapt to the data flow.

To summarize, the challenges faced by streaming data association rules mining are as follows: (1) For online data streams, there is not enough space to store all streaming data. Compressed storage space is necessary for association rule mining; (2) Due to the continuous, borderless, and high-speed characteristics of the data stream, the association rule mining algorithm on the data stream is neither allowed to repeatedly scan the entire database, nor to scan the database as soon as there is an update in the database; (3) Because of the data stream, data distribution characteristics of the data are constantly changing, and the method of association rule mining must adapt to its changing data distribution. Otherwise, it will easily cause the problem of concept migration; (4) Due to the high-speed characteristics of online data streams, they need to be processed as much as possible. The speed of the stream mining algorithm must be faster than the speed of data arrival, otherwise the accuracy of the mining results needs to be sacrificed to improve processing efficiency, such as data approximation, sampling, and load shedding techniques; (5) Due to the contradiction between unlimited streaming data and limited resources, it is necessary to improve the processing efficiency. A mining mechanism that adapts to limited resources, such as memory space and energy consumption, is needed, without decreasing the mining accuracy.

Therefore, there are two key problems to be solved in the study of streaming data association rules: (1) Mining all frequent item sets in the data stream in memory at a faster speed than the data stream input; (2) The relationship between transactions is complex and diverse, and is often not a simple probabilistic relationship. The problem lies in how to use the multi-parameter method to discover the true relationship between transaction sets.

2. Background

Let I={i1, i2,…,in} be a collection of items, and suppose that the task-related database D is a collection of database transactions, where each transaction T is a collection of items, such that $T \in I$. Every transaction has an identifier called TID. Let A be an itemset, and transaction T contains A if and only if $A \in T$. The association rule is an implication of the form $A \rightarrow B$, where $A \in T$, $B \in T$, and $A \cap B = \Phi$. The rule $A \rightarrow B$ is established in transaction D, with support S, where S is the percentage of transactions in D that contain $A \cup B$ (that is, both A and B); it is the probability $P(A \cup B)$. The rule $A \rightarrow B$ has a confidence level C in the transaction set D. If the transaction that contains A in D also contains B, the percentage is C; this is the conditional probability P(B/A). That is:

$$\text{Support}(A \rightarrow B) = P(A \cup B),$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

$$confidence(A \rightarrow B)=P(A \cup B).$$

A rule that meets both the minimum support threshold (min_sup) and the minimum confidence threshold (min_conf) is called a strong rule. If the itemset satisfies the minimum support, it is called frequent itemset (frequent itemset).

We study the mining of data stream association rules as a two-step process: (1) Use the Frequent Tree Interpolation Method (FTIM) to find all frequent item sets, for instantaneous, flowing, and unlimited data streams. FTIM is an improvement on the traditional frequent tree pattern algorithm (FP-tree), which can save memory storage space and reduce frequent itemset mining time. (2) Strong association rules are generated from frequent itemsets. Traditional mining considers this step to be a simple process of finding probability. We believe that the association between data flow items in reality is very complicated, so we use a variety of parameter data fusion methods to analyze strong association rules.

3. The Frequent Tree Interpolation Method (FTIM) Algorithm

3.1 Frequent Itemset Mining

Take the transaction database of AllElectronics in Table 1 as an example to introduce the traditional frequent pattern tree algorithm (FP-TREE), and then we propose the frequent tree interpolation method (FTIM) based on the analysis of the insufficiency of the PF-TREE algorithm.Suppose there are 9 transactions in the database, namely |D|=9, and the items in the transaction are stored in lexicographic order.

Table 1. Transaction data of a branch of AllElectronics (data items in descending order)

TID	List of Item ID
T100	I2,I1,I5
T200	I2,I4
T300	I2,I3
T400	I2,I1,I4
T500	I3,I1
T600	I2,I3
T700	I3,I1
T800	I2,I1,I3,I5
T900	I2,I1,I3

The first scan in the database derives a collection of frequent items (1-items set), and obtains their support counts (frequencies). Suppose the minimum support is 2; the set of frequent items is sorted in the descending order of the support count, and the result set is denoted as L. In this way, we have L={I2: 7, I1: 6, I3: 6, I4: 2, I5: 2}.

The structure of FP-TREE is as follows: First, create the root node of the tree and mark it with "null". Scan database D for the second time. The items in each transaction are processed in descending order in L (that is, sorted by decreasing support degree count) and a branch is created for each transaction. For example, the first transaction {T100: I1, I2, I5} contains three items {I1, I2, I5} in the order of L, resulting in the first branch of the construction tree <(I2:1), (I1: 1), (I5:1)>. The branch has three nodes, among which I2 is the child link of the root node, I1 is linked to I2, and I5 is linked to I1. The second transaction T200 contains items I2 and I4 in the order of L, which leads to a branch, where I2

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

is linked to the root node and I4 is linked to I2. However, this branch should share the prefix <I2> with the existing path of T100. This increases the count of node I2 by 1, creating nodes and links for items following the prefix. To traverse the tree exhaustively, create an item list so that each item points to its appearance in the tree through a node chain. The tree obtained after scanning all transactions is shown in Figure 1. In this way, the problem of mining frequent patterns in the database is transformed into a problem of mining FP-TREE.\

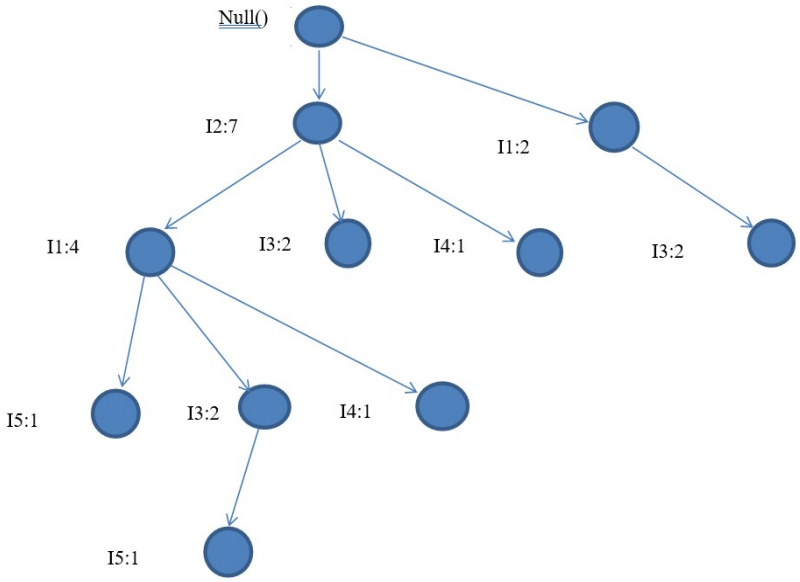

Figure 1. FP-TREE frequent pattern tree construction diagram

FP-TREE mining proceeds as follows: Start with a frequent pattern of length 1 (initial suffix pattern), construct its conditional pattern (a "sub-database" consisting of a set of prefix paths that appear together with the suffix pattern in FP-TREE); then, construct its (conditional) FP-TREE, and recursively mine on the tree; pattern increment is realized by connecting the suffix pattern with the frequent pattern generated by the conditional FP-TREE.

Our research on the performance of FP-TREE mining frequent sets shows that the advantage is that it is effective and scalable for mining long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm. The disadvantage is that for streaming data: 1. Mining frequent sets can only use conditional search; using unconditional search will generate a large number of candidate frequent sets, which increases the computational overhead. 2. For an item that already exists in the previous transaction, if the item appears in the first item in the transaction, FP-TREE must create a new branch. With the continuous influx of data flow, this will cause the FP-TREE branch to continue Expansion will continue to squeeze memory space, greatly increasing memory space overhead. In order to reduce the time cost of calculation and the space cost of memory, we propose the ascending Frequent Tree Interpolation Method (FTIM) to mine frequent sets.

3.2 Ascending Frequent Tree Interpolation Method (FTIM) Analysis

There are two main differences between the ascending FTIM and the FP-TREE method: 1. The frequent tree is constructed in ascending order instead of descending order, with the smallest item as the first node, and the number of suffix nodes will increase sequentially. In this way, if the count of a node meets the minimum support number, the itemsets of the suffix path all meet the minimum support number, and the frequent itemsets are mined; therefore, the frequent itemsets search method is mined from the FP-TREE conditional search method to FTIM The unconditional search method reduces the time cost of calculation. 2. Use the interpolation method when constructing frequent trees. For an item that already exists in the previous transaction, if the item appears in the first item in the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

transaction, FP-TREE must create a new branch. And FTIM inserts the transaction into the frequent tree of existing items instead of creating a new branch, which can reduce the number of branches of the frequent tree, and therefore can reduce the memory space of the frequent tree of data flow based on memory calculation.

Here, let us take the transaction database of AllElectronics as an example (Table 1) to introduce the FTIM mining algorithm. Table 2 here is based on the original Table 1 data in ascending order.

The FTIM mining algorithm is constructed as follows: First, create the root node of the tree and mark it with "null". The database D is scanned for the second time, and the items in each transaction are processed in ascending order in the table (that is, sorted by increasing support count) and a branch is created for each transaction. First consider the transaction that contains the smallest item I5. For example, the eighth transaction {T800: I5, I3, I1, I2} contains four items {I5, I3, I1, I2} in the order of the data in the table, resulting in the construction of the first tree branch <(I5:1), (I3:1), (I1:1), (I2, 1)>. The first transaction {T100: I5, I1, I2} shares the first node I5 to create a new branch. The first item I4 of the fourth transaction {T400: I4, I1, I2} is a new item, and a new first item node I4 is created under the "null" root node to construct a new branch. The main innovation of FTIM lies in transactions {T200, I4, I2}, {T300, I3, I2}, {T500, I3, I1}, {T600, I3, I2}, {T700, I3, I1}, {T900, I3 , I1, I2} The first item already exists in the frequent tree, so insert the corresponding transaction into the tree, and the number of corresponding nodes can be added to 1, as shown in Figure 2.

Table 2. Transaction data of a branch of AllElectronics (data items in ascending order)

TID	List of Item ID
T800	I5,I3,I1,I2
T100	I5,I1,I2
T400	I4,I1,I2
T200	I4,I2
T900	I3,I1,I2
T500	I3,I1
T700	I3,I1
T300	I3,I2
T600	I3,I2

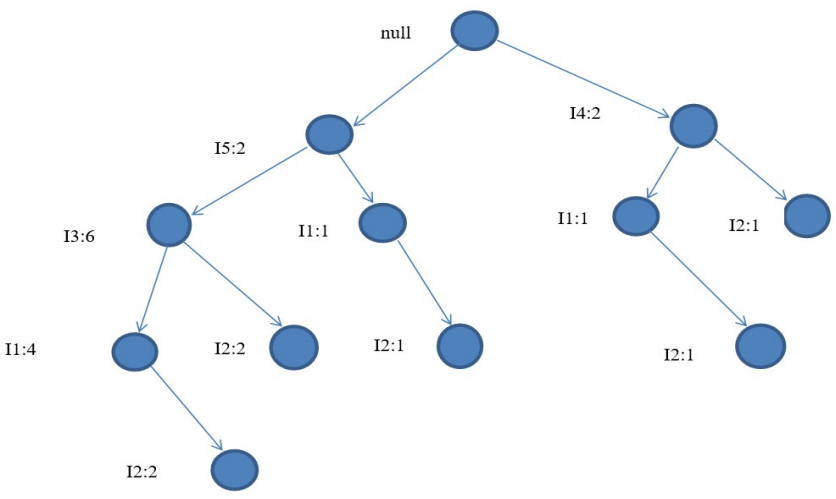

Figure 2. FTIM structure tree

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

The process of FTIM mining frequent sets is as follows. First consider I2, which is the last node of the pattern tree, and the mining idea is to search from bottom to top.I2 appears in the five branches in Figure 2 (the appearance of I2 is found by the chain of nodes along it). In branch (I2: 2, I1: 4, I3: 6, I5: 2) all node counts are not less than the minimum support number 2, so the longest frequent set {I2, I1, I3, I5}; the longest frequent set is {I2, I1, I3, I5}. Each subset is also a frequent set.In the same branch (I2: 2, I3: 6, I5: 2), all node counts are not less than the minimum support number 2, so (I2, I3, I5) are also frequent sets. In branch (I2:1, I1:1, I5:2), although there are nodes (I2, I1), their counts are less than the minimum support number 2, it can be combined with other branches that contain its path. If the combined branch is not less than the minimumsupport number is 2, it is a frequent set, such as (I2, I1, I5). Otherwise, it is an infrequent set, such as branch (I2:1, I1:1, I4:2).

3.3 Mining Association Rules from Frequent Itemsets

In the past, it was straightforward to find frequent itemsets from transactions in database D, and to generate strong association rules from them. For confidence, the following formula can be used, where the conditional probability is expressed by the itemset support count:

$$\text{confidence}(A \rightarrow B)=P(B/A)=\text{support\_count}(A \cup B)/\text{support\_count}(A).$$

Association rules can be generated as follows:

(1) For each frequent item set l, all non-empty subsets of l are generated;

(2) For each non-empty subset s of l, if support_count(l)/support_count(s) ≥ min_conf, output the rule "s→(l-s)", where min_conf is the minimum confidence threshold.

However, in reality, the relationship between data flow projects is very complex, not a simple confidence (A→B) relationship. This is also the main reason for the low industrial application rate of data mining technology. That is to say, the association rules studied at this stage have not really analyzed the association relationship of complex affairs. To this end, we use a multi-parameter data fusion method to analyze the association rules.

Assuming that transactions A and B are strongly related, how to analyze who is the promoter of the relationship and who is the puller of the relationship. For this reason, we use the two parameters of confidence (A→B) and confidence (A→B) to analyze. Taking the original origin data of the association rules as an example, we all know that the diapers (set as A) and beer (set as B) in Wal-Mart chain supermarkets are strongly related, which is defined by confidence (A→B) = 30%~40 % Probability; however, we can further analyze that confidence(B→A)= 3%~5%, that is, confidence(A→B) ›› confidence(B→A), which infers that A is the promoter, B is the puller, which means that the relationship between diapers and beer is that diapers drive beer sales, not the relationship that beer promotes diapers. We analyze the three possibilities of the two parameters of confidence(A→B) and confidence(B→A): (1) The probability of the two parameters is high at the same time, then A→B, B→A is a strong correlation with each other, A , B two business items promote each other. For example, according to the data analysis of the research, Hunan people like chilly (set to A) fried with meat (set to B). The two parameters of confidence (A→B) and confidence (B→A) have high probabilities, and it can be determined that the sales of chilly and meat promote each other. (2) The probability of the two parameters is large and the other is small, then the data item with high probability is the promoter, and the data item with low probability is the puller, such as the association between diapers and beer. (3) The probabilities of the two parameters are both small, indicating that there is a weak correlation between A and B.

Analyze the positive or negative correlation between A and B. The relationship between some transactions is a mutually reinforcing relationship, such as the relationship between diapers and beer, we call it a positive relationship. The relationship between some transactions is a weakening

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

relationship, and transaction items with the same effect are often weakened by each other, such as rice and bread, which we call negative correlation. For this reason, the incremental ratio is adopted for analysis. Suppose ΔA and ΔB are the increments of the two transaction items A and B in a certain period of time. Analysis and calculation of ΔA/ΔB are positive and it is a positive correlation, ΔA/ΔB negative means negative association.

Analyze the correlation between the weight of transaction data of A and B. In the traditional confidence (A→B) correlation analysis, only the ratio of the number of occurrences of A including the number of occurrences of B is considered, and the relationship between the purchase amount of A and B is not analyzed. Here we set the purchase amount as weight, let A1, A2, ... An, B1, B2 ,... Bn represent the purchase amount of each item of A and B, respectively, and use the total weight ratio and the average weight ratio to analyze the correlation of weight. Total weight ratio = (B1+B2+…Bn)/(A1+A2+…An), average weight ratio = (B1/A1+B2/A2+…Bn/An)/n. The two parameters of total weight ratio and average weight ratio are used to analyze the relationship between the two transaction items A and B.

The association between multiple data items. The relationship between transactions is often multi-dimensional, so confidence (A∪B→C), confidence (A∪C→B), confidence (C∪B→A) are used to analyze the complex cross-correlation among multiple data items.

4. Experimental Evaluation

We performed experiments in the following settings. Hardware environment: CPU: IntelE5-2620, memory: 32G DDR4, hard disk: 500GSATA6.0-Gb/s; software environment: Windows 10/Windows 7; visual studio 2017.The Apache Drill tool software was used to collect the big data of Spark Wallet and the transaction data stream data of Safeway, a well-known retail chain in the United States, and realized the comparative analysis of the FTIM algorithm and the traditional FP-TREE algorithm. Figure 3 shows the performance changes of the two algorithms under different minimum support degrees (divided into 5 levels: 1%, 2%, 3%, 4%, 5%). The Y axis is the time spent in running the algorithm. Experimental results show that the smaller the minimum support degree, the more frequent sets and the more CPU computing time. Figure 4 examines the scalability of the two algorithms. The fixed minimum support is 3%. Different transaction numbers (15k, 20k, 25k, 30k, 35k) are randomly selected from the data stream for testing. The experiment also shows that the FTIMalgorithm's existing frequent set utilization is very targeted, so the efficiency is relatively high.

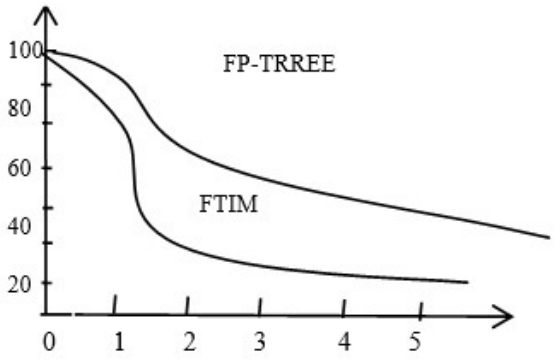

Figure 3. Minimum support

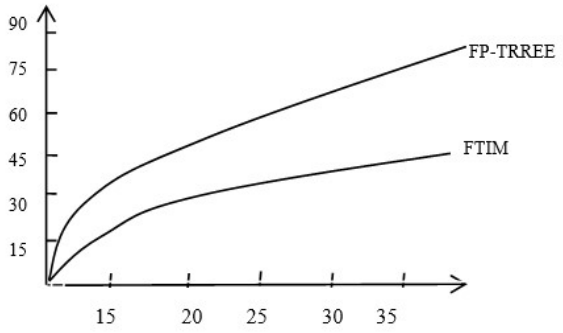

Figure 4. Number of transactions (K)

5. Conclusion

Our conclusions include two main points. First, mining all frequent itemsets. The research adopts the interpolation frequent tree method (FTIM), which can save more memory space than the traditional frequent pattern tree (FP-TREE) algorithm, and it can also reduce the time for mining frequent itemsets. Second, association rules are generated from frequent itemsets. Traditional mining considers

ISSN: 2414-1895 DOI: 10.6919/ICJE.202202_8(2).0027

this step to be a simple process of finding probability. We believe that the association between items of data flow in reality is very complicated. Therefore, multi-parameter data fusion technology is used to analyze strong association rules: 1. Using confidence(A→B) and confidence (A→B) are two parameters to analyze who is the promoter of the relationship and who is the puller of the relationship between A and B.2. Analyze the positive or negative correlation between A and B using incremental ratios.3. Analyze the correlation between the weight of transaction data of A and B.4. Mining the cross-correlation among multiple data items. Experiments show that the method used in this article can save calculation time and memory space more comprehensively to reveal the true relationship of data items.

Acknowledgments

The research is sponsored by: The natural science foundation of Hunan Province: “Research on Data Flow Association Rules Based on Data Fusion Technology” (code:2019JJ40152), Key scientific research project fund of Hunan province Department of Education: “Research on Data Flow Association Rules Based on Data Fusion Technology”(code: 18A307).

References

[1] Agrawal R,Imielinski T,Swami A.Mining association rules between sets of items in large databases.In:Proceedings of ACMSIGMOD International Conference on Management of Date, Washington DC, 1993.207~216.

[2] PAN M,HUANG X J,HE T T ,et al. A Simple Kemel Co-occu-rrence-Based Enhancement for Pseudo-Relevance Feedback. Journal of the Association for Information Science and Technology, 2020, 71(3):264-281.

[3] BOUZIRI A, LATIRI C,GAUSSIER E. LTR-Expand:Query Expansion Model Based on Leaming to Rank Association Rules. Journal of Inteligent Information Systems,2020,55:261-286.

[4] Han J,Kamber M.Fan Ming,Meng Xiao-Feng et al.Translated.Data Mining:Concepts and Techniques. Beijing: China Mechine Press,2001(in Chinese).

[5] Agrawal R,Shafer J C.Parallel mining of association rules:Design,implementation,and experience.IBM Research Report RJ10004, 1996.

[6] Savasere A,Omiecinski E,Navathe S.An efficient algorithm for mining association rules.In:Proceedings of the 21th International Conference on VLDB,Zurich,Switzerland,2005.432~444.

[7] Han J,Jian P et al.Mining frequent patterns without candidategeneration.In:Proceedings of ACM SIGMOD International Conference on Management of Data,Dallas,TX,2000.1~12.

[8] Cheung D W,Lee S D,Kao B.A general incremental techniquefor maintaining discovered association rules. In: Proceedings ofdatabases systems for advanced applications,Melbourne,Australia,2007.185~194.

[9] Feng Yu-Cai,Feng Jian-Lin.Incremental updating algorithms formining association rules.Journal of Software, 2008,9(4):301~306(in Chinese).

[10] Jian P.Mining access patterns efficiently from web logs.In:Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining(PAKDD'00),Kyoto,Japan,2000.396~407.

[11] Agrawal R,Srikant R.Mining sequential pattern.In:Proceedings of the 1lth International Conference on Data Engineering,Taipei,2005.3~14.

[12] Tan J.Weighted association rules mining algorithm research[J].Applied Mechanics and Materials, 2013, 241-244:1598-1601.

[13] Datar M.Algorithms for data stream system[D].Stanford: Stanford University,2016.

[14] Brin S,Motwani R,Silverstein C. Beyond market baskets:generalizing association rules to correlation[J]. ACM SIGMOD RECORD,2016,26(2):265-276.

[15] Han J.Cheng H. Frequent pattern mining: Current status and future directions[J].Data Mining and Knowledge Discovery,2017 14(1):55-86.

[16] OMIECINSKI E. Alternative interesting measures for mining associations in databases[J].IEEE Trans on Knowledge and Data Engineering.2015,15(1):57-69.