International Journal of Database Theory and Application
Vol.9, No.1 (2016), pp.169-184
http://dx.doi.org/10.14257/ijdta.2016.9.1.15

Weighted FP-Tree Mining Algorithms for Conversion Time Data Flow

Xiao-jun Chen[1,2*], Jia Ke[2,3], Qian-qian Zhang[2], Xin-ping Song[3] and Xiao-ming Jiang[2,4]

[1]Affiliated Hospital of Jiangsu University, [2] School of Computer Science and Communication Engineering, JiangSu University [3] School of management, Jiangsu University [4] Automotive Engineering Research Institute, Jiangsu University

[*]cxj@ujs.edu.cn

Abstract

The data distribution in the data streams usually changes dynamically with time. Traditional mining algorithms based on transaction are difficult to establish the correlation between time characteristics and relationship features, thus making the results inaccurate. By analyzing the problems in the processing of time data stream, we put forward the concept of time gap degrees and design a mining algorithms based on weighted FP-Tree. We introduce the concept of FP-Tree node weights to transform the time data dynamically and excavate the data stream association rules. The experiments performed on the actual data set show that the algorithm can improve the recall and precision while consumes comparable computational time.

Keywords*: time data flow, weighted FP-Tree, data mining, association rules*

1. Introduction

The concept of association rules proposed in the literature [1] in 1993 attempted to seek some hidden patterns and interesting knowledge in large amounts of data by mining potential relationship among the set of data items, in order to enhance the performance of decision and analysis. The difficult of association rules mining is to find all frequent item sets [2]. Apriori is a classical algorithm for mining association rules [3]. Through calculating the information related to multiple customers, it finds frequent transaction data sets in supermarket so as to get knowledge sets for analyzing customer buying behavior. With the development of association rules mining, it is widely used in many industries. The literature [4] used a structure called FP-Tree (Frequent Pattern Tree) to save frequent information related to the current mining process in the original database. It first used FP-Growth algorithm to compress database down to a frequent pattern tree (FP-Tree) and then decomposed the compressed database into a set of condition database, each of which is associated with a frequent database. The FP-Growth method can transform the problem of long-term frequent pattern mining into a many problems of short-term frequent pattern mining recursively. By connecting the suffix, this method improves the efficiency of mining greatly [5].

Data stream appears in many fields and these kinds of data are mainly in the form of infinite sequence of data items which are often successive, unbounded, rapid and time-varying. It usually increases with the number constantly. In terms of data stream processing, an algorithm for mining frequent item set must use the limited memory space to save and dispose the data, and has to consider the incremental processing methods for data stream [6]. In order to mine frequent patterns from the time data stream, the literature [7] proposed the "candidate set" pattern and the "test set" pattern. It used the breadth first

ISSN: 2005-4270 IJDTA

search algorithm and a bottom-up sequence to mine. Through improving Apriori algorithm, it designed the U-Apriori algorithm for mining uncertain time data stream. The literature [8] improved classical sliding window method and presented the concept of "tilted window". This method divides the time data segment in different granularities. In such a way, it reduced the space usage in the case of using related time query, and it also designed the FP-stream algorithm to store the support degree of each frequent item set. The literature [9] improved the large sliding window method to deal with a large time data flow. It presented the concept of SWIM by dividing the sliding window into a plurality of segments. Noting that not each segment will have new data, only one piece of data is updated and the efficiency is improved. It also designed a hybrid algorithm of DTV and DFV using the FP-Tree to store frequent pattern data directly. By counting conditions of frequent patterns, it completes the verification and gets more accurate results. Although the above methods can efficiently mine time data flow, all of them are based on the assumption that the transaction and time belong to different dimensions, thus needing multidimensional storage space [10, 11]. Thus, it leads to higher amount of data storage and usually causes the pressure to memory so that a lot of association rules mining work can't continue if it accumulates the affairs calculation.

The FP algorithm is developed to improve the low performance of Apriori algorithm in dealing with long pattern and viscous data set [12]. The FP algorithm can effectively improve the storage density of FP-Tree in the case of dense data set or low supports [13], guaranteeing the mining process have the excellent performance. The main problems of the FP algorithm include: the mining process needs to generate conditional FP-Tree formation recursively thus leading to high time complexity [14]. Because FP-Tree and conditional FP-Tree are required to be bidirectional traversal[15], the tree structure needs to store more information and sometimes it even needs to encounter the insufficient memory problem by using the projection database or recursive projection[16]. In a sparse data set, FP-Tree compression makes frequent time have low information density, which influences the mining performance [17]. The data flow has an effect on the algorithm. The large relational database mining makes it rather difficult to choose the right way to realize the algorithm.

In order to solve the problems of time data stream mining, we improve FP algorithm in this paper by incorporating the characteristics of time sensitive data flow. We put forward the concept of time gap degree and design an efficient and accurate algorithm for mining frequent item sets. This algorithm introduces the weight characteristics into the FP-Growth mining algorithm, making it more suitable for the calculation process of time sensitive data and obtaining a better recall and precision.

2. Related Descriptions

2.1. Time Gap Degree

Data streams have the characteristics of dynamic flow. In other words, data arrival is time-correlated while the data stream has different time intervals. For example, the time interval of each patient's medical hospital has certain classification characteristics. Because the diagnosis and treatment method is different for each patient, the time interval of treatment has its own characteristic. Even for the same diagnosis, it is also different due to the differences in individual patients and doctors habits. In this paper, we suggest the time gap degree to describe the characteristics of transactions, which is the basis of association rules mining algorithm.

The collection of various types of affairs in database is regarded as the item set in the association rule algorithm, which are further defined as $TRASET = \{TRASET_1, TRASET_2, ..., TRASET_m\}$, the time gap degree of transaction is defined as $Span = \{span_1, span_2, ..., span_m\}$, $span_l \in (0,1), l = 1,2,...,m$.

Aiming at the existing problems in data flow processing, we use the time interval description method to improve the traditional FP-Growth algorithm. The time gap is introduced as weight into FP-Growth algorithm, in order to make the association rules generated more accurately reflect the characteristics of time data flow.

2.2. The Relevant Definitions

Definition 1 There is a time flow transaction set including N transactions. The project set $SPANX$ with time gap degrees, $SPANX \subset TRASET$, the support degree of item set $TIMEX$ is defined as support （SPANX） = ($\sum_{t_j \in SPANX} span_j$) *support ($SPANX$). The minimum support threshold is minsup, the minimum confidence threshold is minconf, if support$(SPANX) \geq$ minsup, then $SPANX$ is called frequent item sets of time gap degree. The number of $SPANX$ in transaction which contains X is $SCount(SPANX)$, so $SCount(SPANX) = (minsup * N) / \sum_{t_j} span_j$.

Definition 2 Time flow project set is $SPANX, SPANY$, and $SPANX \cap SPANY = \phi$, if $SPANX \Rightarrow SPANY$ and $confidence(SPANX, SPANY) \geq$ minconf, support$(SPANX \cup SPANY) \geq \leq$ minsup, so $SPANX \Rightarrow SPANY$ is the strong association rule.

Definition 3 Time flow project set is $INTY$, and $INTY$ is q- item sets, $q < k$, so $SPANY$ is the subset of k- item-sets. The time gap degree set of k - set is denoted by $SPAN(SPANY, k)$, so $SPAN(SPANY, K) = \sum_{span_j \in Y} span_j + \sum_{i=1}^{k-q} span_i$ is in the transaction set $SPANDB$, including k- item-sets. The transaction number of $TRASET_k$ including k-item-sets is called time gap support count of $span_k$ in $SPANDB$, which is recorded as $sup\_count(TRASET_k)$. If

$$sup\_count(TRASET_k) \geq \frac{minsup \times N}{\sum_{t_j \in SPANX} TRASET_j},$$

$TRASET_k$ is the time gap frequent k-itemset of transaction sets $SPANDB$.

3. Weighted FP-Tree Algorithm

3.1. The Description of FP-Growth Algorithm

The algorithm is described as the follows:
Input: transactional databases TSET, and minimum support threshold minsup
Output: the complete set of frequent patterns
Method:
(1) Construct FP-Tree:
(a) Scan transaction database TSET. Count the support degree of item set to get the frequent 1- item sets F. Sort the frequent 1- item sets support degree in descending order to get the frequent item table L.
(b) Create the root node of FP-Tree, and use "null" to mark it. For each transaction T in the database TSET, we can perform the following operations: select the frequent items in transactions T and sort it in descending order of L. Set the sorted frequent item table to [p, P]. In the frequent item table, the first element is p, and the table of remaining elements is P. At the last, invoke insert tree ([p, P], T).

The process implementation is described as follows: If the transaction T has children N then make N.item_name=p.item_name, the count for the N will increases l. Otherwise, we create a new node N, and set its count to 1, and link it to its parent T node, and link them to the node having the same item_name by node chain structure. If P is non-empty, recursively invoke the insert_tree (P, N).

(2) Implement FP-Tree mining by invoking FP-Growth (FP-Tree, null). The realization of the process is as follows:

Procedure FP-Growth (Tree, a)

(a) If Tree includes a single path P then

(b) for every combination of nodes of path p (which is called β)

(c) create the pattern $\beta \cup$ a

(d) else for each a_i in the head of the Tree {

(e) produce a pattern $\beta = a_i \cup$ a, whose support degree is support= a_i .support;

(f) construct the conditional pattern base of β , then construct the condition FP-Tree β of β

(g) if Tree $\beta \neq \varnothing$ then

(h) invoke FP-Growth(Tree β , β) ; }

Table 1. Transaction Items TSET

TID	Items
T1	I1, I2, I5
T2	I1, I2, I4
T3	I2, I3
T4	I1, I2, I3, I5
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I2, I4
T9	I1, I2, I3
T10	I1, I2

As shown in Table 1, select 10 transactions from TSET database, set the minimum support count is 2 (that is minsup=2/10=20%). Ti is the transaction number, Ii stands for a specific project.

(1) Generate frequent pattern

(a) The process of using FP-Growth algorithm to mine association rules of the frequent item sets is expressed as follows:

Scan the transaction database TSET. Export the frequent item set (*i.e.*, frequent 1- item sets), and get their support counts. Sort the set of frequent items according to the descending order of the support count, and let the result be denoted as L. L= {I2:8, I1:7, I3:6, I4:2, I5:2}.

(b) Scan the transaction database TSET again, construct FP-Tree.

The mining process of FP-Tree is: from a frequent pattern with length 1 (i.e. the frequent item in header Table of the project), construct its conditional pattern bases. Then construct its conditional FP-Tree, and mine on this tree recursively. It realizes pattern growth by connecting the suffix pattern and the conditional FP-Tree pattern. The mining for FP-Tree is shown in Figure 1 (b). The frequent pattern generated from Figure 1 (b) is shown in Table 2.

Table 2. Generate Frequent Pattern

Item	conditional pattern base	conditional FP-Tree	frequent pattern
I5	{（I2 I1:1），（I2 I1 I3:1）}	< I2:2, I1:2>	I2I1I5:2,I2I5:2,I1I5:2
I4	{ (I2 I1:1), (I2:1）}	< I2:2>	I2I4:2
I3	{（I2 I1:2），(I2:2），(I1:2）}	< I2:4, I1:2>,< I1:2>	I2I1I3:2,I1I3:2,I2I3:4
I1	{（I2:4）}	< I2:4>	I2I1:4

Through the above analysis, we can get the final frequent item sets generated and the corresponding support degree as: I2I1I5:2, I2I5:2,I1I5:2, I2I4:2, I2I1I3:2,I1I3:2, I2I3:4,I2I1:4.

(2) Discovering the association rules

From the frequent item sets generated by the above process, we can find the corresponding strong association rule, which satisfy the minimum support degree and minimum confidence at the same time.

Confidence ($C \Rightarrow D$) = support($C \cup D$)/support(C)

It can be judged through the following methods:

(a) For any two elements C and D in frequent item sets L, do the judgment:

If ($C \cap D = \varnothing$ and $C \cup D$ is also frequent set (i.e. $C \cup D$ is also in L)

then calculate support($C \cup D$)/support(C)

if (support($C \cup D$)/support(C)≥min_conf)

then generate the strong association rule $C \Rightarrow D$ and save it.

(b) if (all the elements in the L are traversed)

Then output all the strong association rules

else return to 11 and continue

For the above example, according to the association rules generated from the frequent item sets, and for the frequent item sets {I1, I2, I3}, its non-empty subsets include {I1, I3}, {I1, I2}, {I2, I3}, {I1}, {I2}, {I3}

$I1 \wedge I2 \Rightarrow I3$ confidence=2/4=50%

$I1 \wedge I3 \Rightarrow I2$ confidence=2/2=100%

$I2 \wedge I3 \Rightarrow I1$ confidence=2/4=50%

$I1 \Rightarrow I2 \wedge I3$ confidence=2/6=33%

$I3 \Rightarrow I1 \wedge I2$ confidence=2/6=33%

$I2 \Rightarrow I1 \wedge I3$ confidence=2/7=29%

Suppose min_conf (the mimimum confidence) is 90%, then only the rule of $I1 \wedge I3 \Rightarrow I2$ can be selected.

3.2. The Contribution Process of Weighted FP-Tree

The process create (SPANTree) for constructing frequent pattern tree FPSPAN-Tree containing the time gap is described as follows:

Algorithm: create (SPANTree), construct a time gap frequent pattern tree FPSPAN-Tree

Input: the transaction database *SPANDB* (the transaction number in *SPANDB* is N), the item sets, the time gap degree sets *span* corresponding to *TRASET* , and minimum support threshold minsup.

Output: the frequent pattern tree SPANTree related to *SPANDB*

(1) By scanning the transaction database *SPANDB*, generate all the frequent 1- item sets and their support numbers, and insert into the Header table according to its support with descending order.

(2) Scan *SPANDB*, and calculate the support degree for each item, set K as the maximum number of items in each transaction.

(3) Set $k\_min = \left\lceil \frac{minsup \times N}{\sum_{j=1}^{k} span_{r_j}} \right\rceil$;

(4) The project whose support degree is larger than k_min is added to the candidate table List of frequent item, where the items are called frequent candidate.

(5) Create the root node of FPSPAN-Tree which is labeled as NULL. For each item in *SPANDB*, carry out the following operations:

Sort the current transaction according to the order in table List. Let the frequent candidate item sets table be $[p|P]$, where p is the first project, P is the remaining items list after p, and invoke insert_INTTree($[p|P]$,NULL). Function insert_INTTree($[p|P]$, Q_{Tree}) is defined as

If Q_{Tree} is not null then

(1) Take the first item R in Listi

(2) If R is a child node of FPSPAN-Tree, then R.count = R.count+1

else

(2.1) Create a new node R

(2.2) R.count=1

(2.3) Let the parent node pointer pointe to Q_{Tree}, its chain of nodes node-link links it to the node having the same item name;

(3) insert_INTTree($[p|P]$, Q_{Tree})

3.3. Weighted FP-Growth Algorithm

Let $SPAN_X$ be the conditional FPSPAN-Tree of X, then the initial FPSPAN-Tree is $SPAN_\Phi$ (ϕ is the empty set). The project set β is the suffix set of k- item set α. Then FPSPAN-Growth (FPSPAN-Tree, α)

(1) If there is only one path P in FPSPAN –Tree, then

(2) For all nodes in β are added to the path of P, go to (3); otherwise go to (6)

(3) Let the generated $\beta \cup \alpha$ as the weights of frequent candidate set

(4) Let the minimum value of node support degress in β be s, $x = \frac{(\sum_{TRASET_j \in \beta} span_j) * s}{N}$,

(5) If $x \geq$ minsup, $\beta \cup \alpha$ is the frequent item, and finish.

(6) The support degree of the node of $TRASET_j$ in FPSPAN -Tree is denoted as s, and $\beta = TRASET_j \cup \alpha, x = \frac{span_i \times s}{N}$. If $x \geq minsup$, β is the frequent item.

(7) Construct the conditional pattern base of β, and order the maximum length of the conditional pattern for k, then FP-Tree of β is denoted by $INTTree_\beta$; invoke FPSPAN_Growth($SPANTree_\beta, \beta$);

In FPSPAN-Growth, because the time gap degrees of each frequent item are different, the time gap support degrees are also different. The time gap support counting in the non-

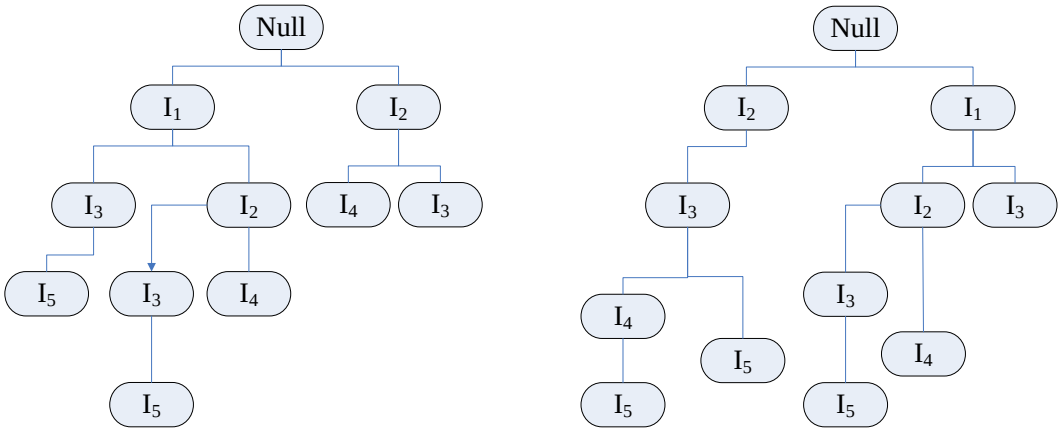
empty proper subset of generated frequent item sets also different. We take the mean of each time gap support degree in a non-empty subset as the time gap support of this subset, which is a basis to generate strong association rules. In what follows, we analyze the transaction set in Table 4.1 by using the improved algorithm of FPSPAN-Growth with time discontinuous.

3.4. The Prototype Case of FPSPAN-Tree

The 10 transaction degree sets with time gap degree are shown in Table 3.

Table 3. Transaction Items of TSET with Weight

TSET	The project list of affairs（What in parentheses is the time gap values）
T1	I1（4）, I2（2）, I5（4）
T2	I1（5）, I2（2）, I4（4）
T3	I2（1）, I3（2）
T4	I1（3）, I2（2）, I3（1）, I5（4）
T5	I1（5）, I3（2）
T6	I2（2）, I3（4）
T7	I1（1）, I3（4）
T8	I2（2）, I4（3）
T9	I1（3）, I2（2）, I3（2）
T10	I1（3）, I2（1）



(a) FPSPAN-Tree (b) FP-Tree

Figure 1. Compare with FPSPAN-Tree and FP-Tree

Due to the introduction of time gap degree weights, the resulting set of the frequent items by FPSPAN-Growth algorithm is L1={ I1:24,I3:15,I2:14,I5:8,I4:7 }.We can see that L1 is different from L={ I2:8,I1:7,I3:6,I4:3,I5:2 } which is generated by unimproved FP-Growth algorithm. Specifically, the location of I1,I2 and I4,I5 are changed. This is because although the number of nodes of I2 is more than that of I1, the time gap degrees of I1 is relatively high, so I1 ranks in the front.

The comparison of FPSPAN-Tree and FP-Tree is shown in Figure 1. We can see the relationship between the position of the data item tree leaf nodes have obvious changes. In particular, the position of the four leaf nodes of I1, I2 and I4, I5 are exchanged.

The inference rules generated are as follows:

$I1 \Rightarrow I3 \wedge I2$ confidence=2/2=100%

$I3 \Rightarrow I2 \wedge I5 \wedge I4$ confidence=2/2=100%

$I3 \wedge I2 \Rightarrow I1$ confidence=2/4=50%

We can see from this example that after introducing the time gap degree, FPSPAN-Growth modifies the derivation rules by means of weights.

4. Experiments

We implement FPSPAN-Growth on Visual C # 2010, and the environments are: 4GB memory, Petium2.31GHz CPU and Windows XP operating system.

4.1. The Space Complexity Analysis of the FPSPAN-Growth Algorithm

The FPSPAN-Growth algorithm needs to store the weight value in the space complexity. The space complexity of FP-Growth on FP-Tree is O(N), and the space complexity of FPSPAN-Growth on FPSPAN-Tree is O(N), both of which have the same complexity. In contrast, the other multidimensional association rules mining algorithms all need the matrix to store the multidimensional relation vector. As a result, the space complexity of multidimensional association rules mining algorithms is $O(N^2)$. That is to say, the FPSPAN-Growth algorithm developed in this paper can use less store space, and at the same time achieve the similar effect of multidimensional association rules mining algorithms.

4.2. The Time Complexity Analysis of the FPSPAN-Growth Algorithm

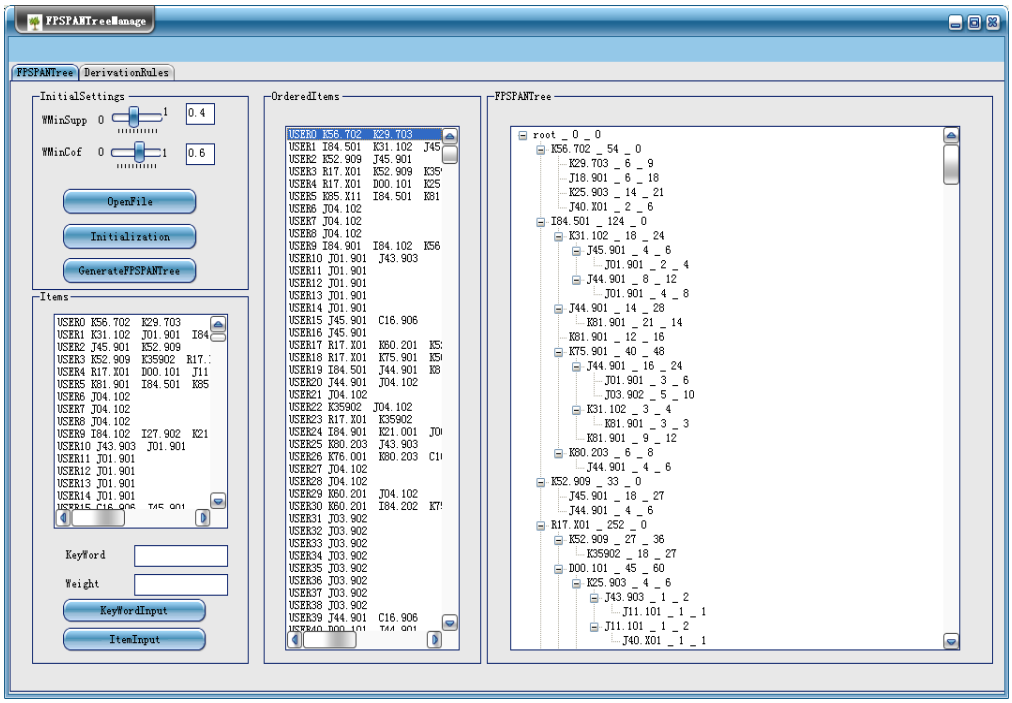
In time complexity, the FPSPAN-Growth algorithm only modifies the counting rules of the original algorithm. The FP-Growth algorithm needs to perform accumulative count on the intensive transaction in constructing FP-Tree, while the FPSPAN -Growth algorithm is to perform accumulative count on the transaction weights. In the algorithmic language, the FP-Growth algorithm and the FPSPAN -Growth algorithm all perform add operation. As a result, the FPSPAN -Growth algorithm does not increase the computation burden in time complexity. FP-Growth is a very fast derivation rule algorithm which owns higher efficiency in comparison with other algorithms based on the Apriori algorithm, neural network and genetic algorithm. So FPSPAN-Growth algorithm is also an efficient algorithm.

4.3. Simulation Experiments

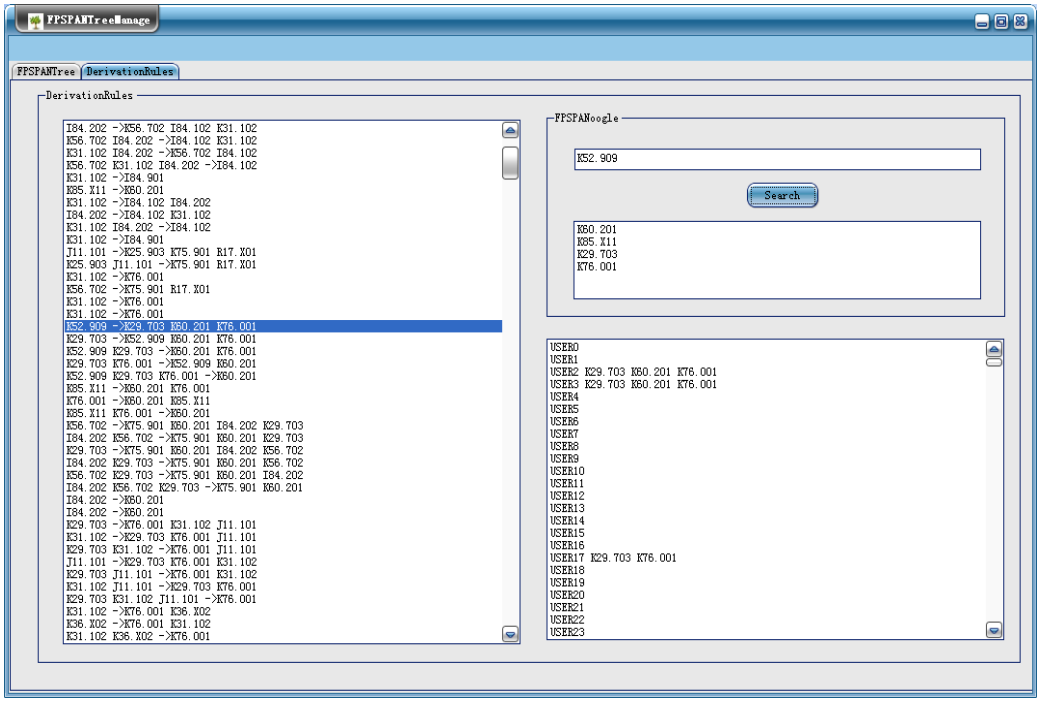
We perform experiments on actual datasets (international classification of diseases 10 from Hospital Information System) to demonstrate the computation efficiency and computation results of our method. Using the operating time of the system and the derivation rule evaluation, we compare FP-Growth algorithm and FPSPAN− Growth algorithm. The actual test user interface is shown in Figure 2. Figure 2 (a) is a window for the realization of FPSPAN-Tree. The user can set the weighted minimum support degree minsup and minimum confidence threshold minconf in the left side of the interface.

The lower left is the concept sets converted from the users searching key. The middle part of interface shows item sets generated after the first scanning the database while the right side of the interface shows the generated FPSPAN-Tree.

Figure 2 (b) illustrates the results of the frequent association rules by mining the potential relationship between events of FPSPAN-Tree. The button in the top right of the window is the research button, and the search can recommend other items related to users.



(a) The Construction of System Interface of FPSPAN-Tree



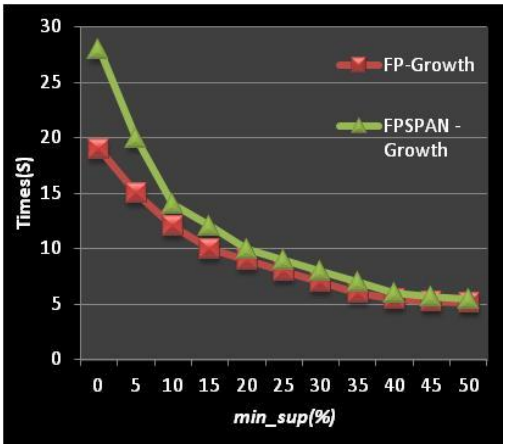
(b) Association Rules of Mining

Figure 2. Example Data Set Data and a Sample Data

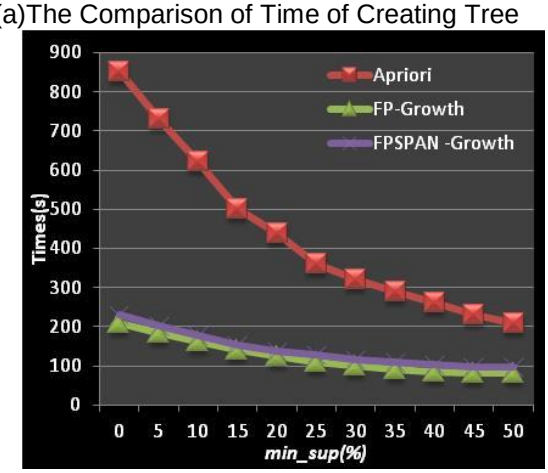
4.4. Executive Time Experiment

To test the performance of the algorithm, we compare the executive time of the FP-Growth, FPSPAN-Growth and multidimensional derivation rule algorithm based on the Apriori. In experiments, the number of transaction items is 50, the number of the transaction sets are 5000 records,

Minsup= 0,5%,10%,15%,20%,25%,30%,35%,40%,45%,50% . The initial experimental result for constructing tree is shown in Figure 3 (a). The comparison of total computation time for each method is shown in Figure 3 (b).



(a)The Comparison of Time of Creating Tree



(b) The Comparison of Total Time

Figure 3. The Comparison of Time

It is not difficult to see that, the executive time of FP-Growth is less than the FPSPAN − Growth, this is because the FPSPAN− Growth needs calculation on real numbers, which consumes a little more time.

Through comparing the overall actual computation time, we can see that, the executive time of FP-Growth algorithm is less than that of FPSPAN-Growth algorithm, while the time of the FPSPAN-Growth algorithm is less than the multidimensional derivation rule algorithm based on the Apriori. The FPSPAN -Growth algorithm is slower than FP-Growth because in the calculation of cumulative time gap weight process, the FPSPAN-Growth uses slightly more time. In contrast, the multidimensional derivation rule algorithm based on the Apriori needs more time to calculate.

In order to test the algorithm further, the minsup in FPSPAN-Growth is set to 40% (because when minsup is 40%, the variation of executive time of the system began to flatten). We compare FP-growth to FPSPAN-Growth, and input several different query conditions to perform experiments. The result is shown in Table 4.

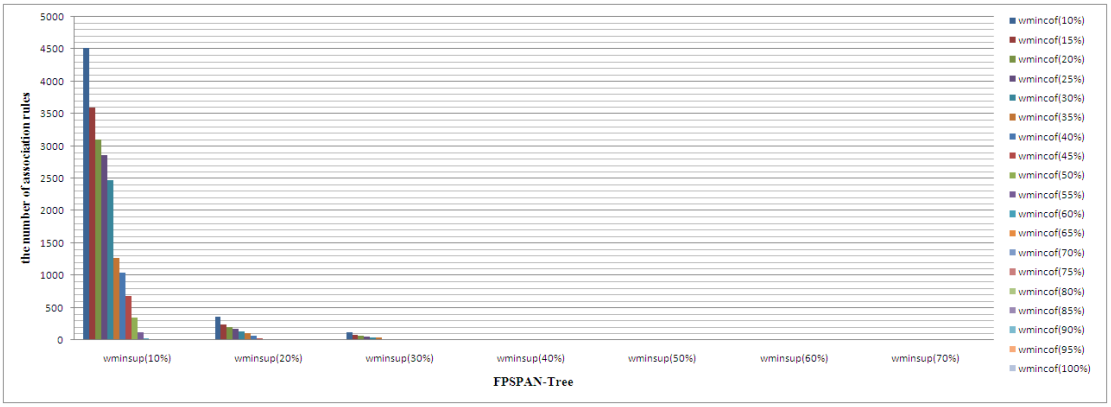
Table 4. The Comparison of Executive Time between FP-Growth and FPSPAN-Growth

query categories	FP-Growth (s)	FPSPAN-Growth(s)
Respiratory system	2.57	2.59
Digestive system	3.33	3.41
Anorectal	4.45	4.52
Orthopedics	3.55	3.69
Cardiovascular system	2.64	2.71
Cardiovascular system	4.56	4.68
Gynecological system	3.54	3.74

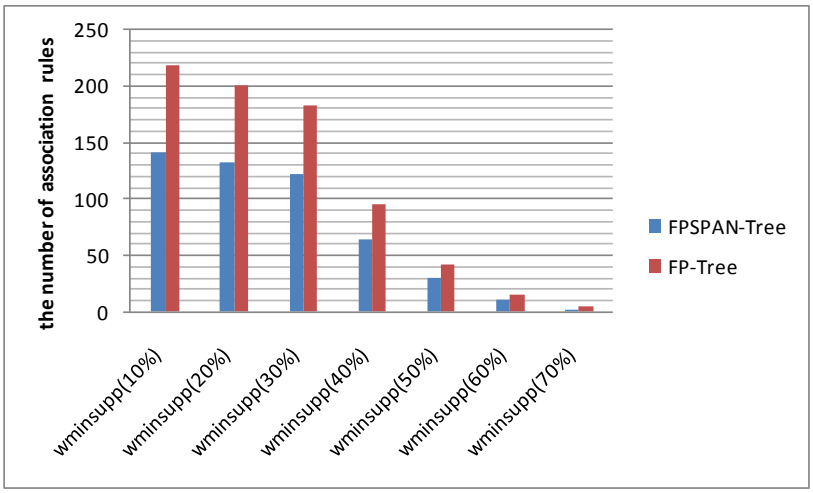
From Table 4, we can know that, the executive time of FPSPAN-Growth is a little lager than that of FP-Growth. This is because filtering FPSPAN-Growth uses slightly more time in calculating cumulative user interest right value.

4.5. Comparison of Generation Rules

We perform experiments to compare the number of rules generated by the multidimensional derivation rule algorithms of FP-Growth and FPSPAN-Growth. In experiments, the amount of transaction projects is set to 50, and 5000 records of the transaction set are selected. We set minsup = 10%, 20%, 30%, 40%, 50%, 60%, 70% and minconf = 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, 55%, 60%, 65%, 70%, 75%, 80%,85%,90%,95%,100%. The comparison of the number of association rules with different values for minsup and minconf is shown in Figure 4. (a). The comparison of the number of association rules produced by FP-Tree and FPSPAN-Tree is shown in Figure 4. (b).



(a) The Comparison of the Number of Association Rules between Minsup and Minconf



(b) The Comparison of the Number of Association Rules between FP-Tree and FPSPAN-Tree

Figure 4. The Comparison of the Number of Association Rules

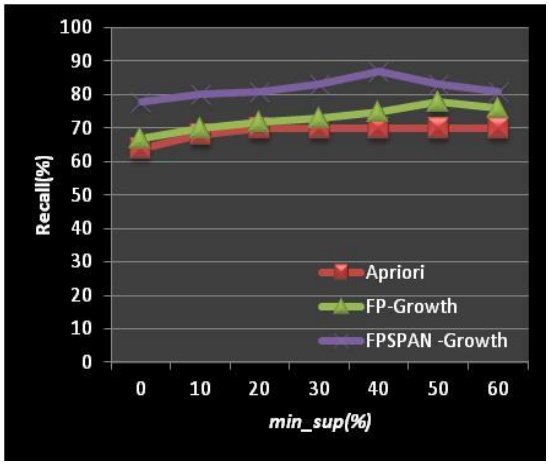
Figure 4 (a) shows the diminishing trend corresponding to the number of the association rule generated from the system, the reason of which is that the larger value of minsup, the more frequent item sets filtered. Moreover, the number of rules gotten from the system in FPSPAN-Tree is less than FP-Tree. The reason is that more non-frequent items are filtered in FPSPAN-Tree method with timing parameters, so the FPSPAN-Tree is more accurate than FP-Tree.

4.6. The Evaluation of the Derivation Rules

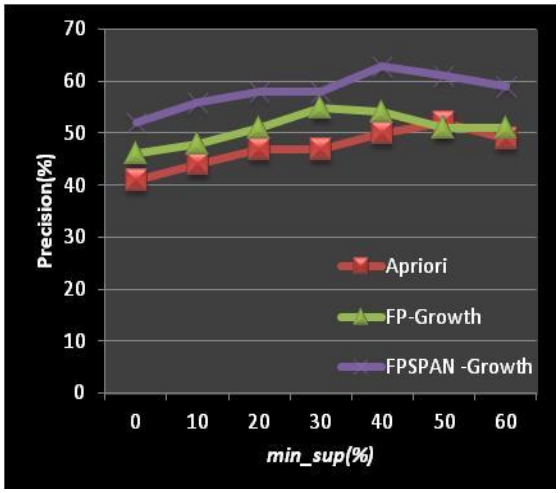
The experimental evaluation indices used in this part are Recall and Precision. Recall reflects the completeness of the association rule mining while Precision reflects the accuracy of the association rule mining. The formula is shown as follows where S denotes the correct set of mining rules, and the R denotes the calculated mining rules.

$$Recall = \frac{|S \cap R|}{|R|} \times 100\% \quad (1)$$

$$Precision = \frac{|S \cap R|}{|S|} \times 100\% \quad (2)$$



(a) The Comparison of the Recall



(b) The Comparison of the Precision

Figure 5. The Comparison of the Recall and the Precision

(1) Recall test

From Figure 5 (a), we can know that, when minsup= 40% , the Recall of FPSPAN– Growth is the highest. This is because when minsup< 40% , the item sets are too large and can't reflect the rules, while when minsup> 40% , the item sets are too small, and many frequent item sets are filtered and the rules are reduced. All of these factors lead to the low Recall. But comparing with the FP-Growth, no matter the value of minsup, the Recall of FPSPAN–Growth is much higher than the FP-Growth.

(2) Precision test

When minsup= 40% , we compare the Precision of FPSPAN-Growth and FP-Growth and show the result in Figure 5. (b). From this result, we can know that when minsup= 40% , the Precision of FPSPAN–Growth is much higher than that of FP-Growth.

5. Conclusion

Data stream is temporal correlated. Aiming at the problem of the conversion of the time data flow, this paper proposes the concept of the time gap degree, designs an algorithm based on FPSPAN-Tree and FPSPAN-Growth. It can merge the time data flow in the calculation of the single-dimensional FP-Growth algorithm in order to get a more reasonable method for mining frequent item sets in the time data flow. After the comprehensive analysis and comparison of the experimental results in terms of the running time of the system and the derivation rules, we draw a conclusion that FPSPAN-Growth leads to better Recall and Precision than other related methods while sharing similar time complexity.

Acknowledgements

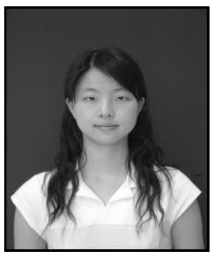
This research has partially been supported by the project funded of the Department of Transportation Informatization under Grant No. 2013-364-836-900, National Natural Science Foundation of China under Grant No. 71573107, 41374129, 41474095, 60673190 and 61203244, College Natural Science Research of Jiangsu Province under Grant No. 14KJB520008, Senior Technical Personnel of Scientific Research Fund of Jiangsu University under Grant No. 13JDG126, Research Innovation Program for College Graduates of Jiangsu Province under Grant No. KYLX15_1078.

References

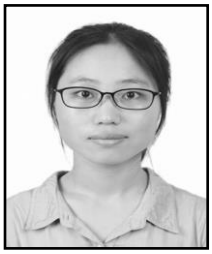
[1] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", Proc.ACM SIGMOD, vol. 5, **(1993)**, pp. 207-216.
[2] V. S. Tseng, B.-E. Shie, C.-W. Wu and P. S. Yu, "Efficient Algorithms for Mining High Utility Item sets from Transactional Databases", IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 25, no. 8, **(2013)** August, pp.1772-1786.
[3] R. Agrawal and R. Stikant, "Fast Algorithms for Mining Association Rules", Proceeding of the 20th International Conference on Very Large Database. Santiago, Chile, **(1994)**, pp. 487-499.
[4] J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation", Proc. of the ACM SIGMOD International Conference on Management of Data (SIGMOD'00). New York: ACM Press, **(2000)**, pp. 1-12.
[5] W. Song, Y. Liu and J. Li, "Mining high utility item sets by dynamically pruning the tree structure", Applied Intelligence, vol. 40, no. 1, **(2014)**, pp. 29-43.
[6] Z. Ao-Ying, J. Che-Qing and W. Guo-Ren, "A survey on the Management of Uncertain Data", Chinese journal of computers, vol. 32, no. 1, **(2009)**, pp. 1-16.
[7] C. K. Chui, B. Kao and E. Hung, "Mining frequent item sets from uncertain data. Lecture notes on computer science", advances in knowledge discovery and data mining, 5012/2008, **(2008)**, pp. 47-58.
[8] C. Giannella, J. Han, J. Pei, X. Yan and P. S. Yu, "Mining frequent patterns in data streams at multiple time granularities", Kargupta H, Joshi A, Sivakumar K, Yesha Y eds. Next Generation Data Mining AAAI/MIT, **(2003)**, pp. 191-210.
[9] B. Mozafari, H. Thakkar and C. Zaniolo, "Verifying and mining frequent patterns from large windows over data streams", Proceedings of the International Conference on Data Engineering Cancun, Mexico, **(2008)**, pp. 179-188.
[10] C. K. S. Leung and B. Hao, "Mining of frequent item sets from streams of uncertain data", In proceedings of the IEEE international conference on data engineering, **(2009)**, pp. 1663~1670.
[11] C. C. Aggarwal, Y. Li and J. Wang, "Frequent pattern mining with uncertain data", In proceedings of the International conference on knowledge discovery and data mining, **(2009)**, pp. 29~37.
[12] J. Han and J. Pei, "Pattern growth methods for sequential pattern mining: principles and extensions", ACM SIGKDD, San Francisco, California, USA, **(2001)**.
[13] Z. Z. Meng, "Frequent patterns mining algorithm and privacy protection based on IFP tree", Taiyuan: Taiyuan University of science & Tethnology, **(2012)**.
[14] W. Hua, M. Han and Y. Gong, "Baseball scene classification using multimedia features", In: Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 1, **(2002)**, pp. 821–824.
[15] M. Fan and C. Li, "Mining frequent patterns in a FP-Tree without conditional FP-Tree generation", Journal of Computer Research and Development, vol. 40, no. 8, **(2003)**, pp. 1216-1222.
[16] Q. X. Ma, G. S. Li and M. Sun, "Current status and typical application on frequent pattern mining", Computer Engineering and Applications, vol. 47, no. 15, **(2011)**, pp. 138-144.
[17] C. F. Ahmed, S. K. Tanbeer and B.-S. Jeong, "An efficient method for incremental mining of share-frequent patterns", Web Conference (APWEB), **(2010)**, pp. 147-153.

Authors

Xiao-jun Chen, He is a doctoral candidate (studying) of Jiangsu University, Zhenjiang in Jiangsu province, China. He received the BS degree in software design and MS degree in computer application both from Jiangsu University in 2004 and 2007.He has working in Affiliated Hospital of Jiangsu University since 2009. His main research interests include pattern recognition and network.



Jia Ke, She is Lecturer School of Business Administration of Jiangsu University since 2004. Meanwhile, she received her Doctor degree in School of computer science from Jiangsu University, Zhenjiang, China in 2013.She received her Ms degree in School of computer science from Jiangsu university in 2007. Her main research interests include pattern recognition and Data mining.



Qian-qian Zhang, She is a master student of Jiangsu University, Zhenjiang in Jiangsu province, China, and her major is computer technology. She received the BS degree of software engineering from Jiangsu University in 2014. Her main research is digital image processing and digital watermark technology.

Xin-ping Song, She is Professor at School of Business Administration of Jiangsu University since 2015. Meanwhile, she received her Doctor degree in Management science and Engineering from Donghua University, China in 2008. Her main research interests include business intelligence and information management, competitive intelligence and knowledge management, research and application of network marketing.

Xiao-ming Jiang, He is Associate Professor at School of computer science and communication engineering of Jiangsu University since 2003. Meanwhile, he received his Ms degree in School of computer science from Jiangsu university in 2008. His main research interests include optical communication technology, mobile communication technology and Internet of things technology.