A Generalized Weighted Closed Sequential Pattern Mining Algorithm with Item Interval

Haitao Lu[1,2,a], Shuo Li [3,b]

[1]College of Information Science and Engineering, Yanshan University, Qinhuangdao, China

[2]The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao, China

[3]Library of Yanshan University, Yanshan University, Qinhuangdao, China

[a]donycosmos@163.com, [b]shuoli@ysu.edu.cn

Keywords: weighted sequential patterns, closed sequential patterns, item interval

Abstract. The algorithm of this paper inserts pseudo items which are converted from item interval to obtain equal extended sequence database; it defines item-interval constraints, which are relative to the item weight, to prune the mining patterns. Through doing this, the algorithm avoids mining the patterns which users are not interested in and shortens the running time. It adopts histogram statistic pattern to get the standardization description to item interval of the mining patterns, making the mining sequences include the item interval information which is valuable to user decision.

1. Introduction

In the traditional sequential patterns mining algorithm, the item in the sequence is equal, but in the actual, the importance of item in the sequence is different according to the realistic attributes. WSpan algorithm[1] for the first time put forward the concept of weighted sequential patterns, according to the project properties of the project in the sequence given the corresponding weights.

Now, there are two methods to integrate item interval in sequence pattern mining. One is the mining algorithm based on item interval constraint, the other is extended sequence mining algorithm. Mining algorithm based on item interval constraints is to extract sequential patterns which would satisfy the minimum weighted support and item interval constraint that the user specified[2,3]. Extended sequence mining algorithm in order to extract sequence patterns that contains the same item but different item interval constraint by to perform algorithm just one time, puts forward the algorithm based on extended sequence[4-7], these algorithms by converting item interval into pseudo item to extract contains item interval sequence mode. But these algorithms due to its not any constraints associated with the item interval, it can dig up some meaningless pattern, like a sequence contains very strength item interval constraint.

Because the mining algorithm of now have no different constraints according to the difference of item weights on the interval between sequence items, the eventually mining result does not contain the important information which is item interval. Therefore, this paper put forward a generalized weighted closed sequential pattern mining algorithm with item interval, the algorithm can deal with two kinds of item interval: item spacing and time interval; By inserting pseudo section function based on interval, and get the interval extension sequence of equivalence with the original sequence; Introduction of item interval constraint conditions related to the item's weighting, according to different weights of item set different item interval constraint condition; Take the form of statistical histogram for mining the item interval information in sequential patterns. The generalized weighted closed sequential pattern with item interval is a new algorithm for mining sequential patterns, by introducing an item interval constraint and weight constraints in the process of mining, mining results closer to the user's actual demand.

2. Problem definition

Sequence Database(SDB). $SDB=\{S_1, S_2,..., S_n\}$ is a tuple collection of $<sid,S>$, the sequence $S=<s_1, s_2,..., s_m>$ is an ordered list of item-sets according to the chronological order, *sid* is corresponding sequence identifier. The sequence *S*'s timestamp list $TS(S)=<t_1, t_2,..., t_m>$ is list of corresponding timestamp in sequence of item-sets list.

Weight of Item. Weight is nonnegative real Numbers which used to denote the importance of each item in the sequence. In order to reflect the importance of item in the sequence, we use item's attribute values in sequence database as the item weight.

Item Interval. Item interval refers to the distance between the items. We use the $t_{\alpha,\beta}$ on behalf of item X_α and item X_β. Item interval have two kinds of representation: item spacing and time interval. Item spacing is defined as the number of items between two consecutive items. $t_{\alpha,\beta}$ represents the items number between item X_α, and item X_β. defined as $t_{\alpha,\beta}=\beta-\alpha$. Time interval refers to the time difference between two consecutive times. $t_{\alpha,\beta}$ represents the time interval between item X_α and item X_β, defined as $t_{\alpha,\beta}=X_\alpha.time-X_\beta.time$.

Interval Piecewise Function. In order to make the item interval changes on a smaller scale, we define the interval piecewise function to implement standardize management over item interval.

Interval Extension Sequence. The interval extension sequence refers to a list contains pseudo items, expressed as $IS=<(I(t_{1,1}),X_1),(I(t_{1,2}),X_2),(I(t_{1,3}),X_3),\ldots,(I(t_{1,m}),X_m)>$. Among them, $X_i(1\le i\le m)$ is items, $t_{\alpha,\beta}$ represents the item interval between item X_α, and item X_β. When the data set contains occurs time information of item, like a timestamp, so $t_{\alpha,\beta}$ means time interval, defined as $t_{\alpha,\beta}=X_\alpha.time-X_\beta.time$. On the one hand, when the data set does not have any occurs time information of item, $t_{\alpha,\beta}$ means an item spacing, defined as $t_{\alpha,\beta}=\beta-\alpha$.

Interval Extension Sequence Database(ISDB). $ISDB = \{IS_1, IS_2, \ldots, IS_n\}$ is tuple collection of $<isid,IS>$, refers to the sequence database composed by the interval extension sequence.

Item Interval Constraint. If there is no item interval constraint, the interval extension sequential patterns of mined may contain many unimportant sequential patterns. To these unimportant sequential patterns are pruned, users should be defined item interval constrain.

We use four types of item interval constraint. $<(t_{1,1},X_1),(t_{1,2},X_2),(t_{1,3},X_3),\ldots,(t_{1,m},X_m)>$ express an interval extension sequence, as shown in the following four types of constraints.

C1: *min_interval* express the minimum item interval between two adjacent items in sequence. $t_{i,i+1}\ge min\_interval$

C2: *max_interval* express the maximum item interval between two adjacent items in sequence. $t_{i,i+1}\le max\_interval$

C3: *min_whole_interval* express the minimum item interval between the head and the tail items in sequence. $t_{1,m}\ge min\_whole\_interval$

C4: *max_whole_interval* express the maximum item interval between the head and the tail items in sequence. $t_{1,m}\le max\_whole\_interval$

C1 and C2 meet the anti-monotonicity (that is, when the sequence A does not satisfy the constraint conditions, any superset of A may not satisfy constraints). In practice, for different weight of the item we have allowed the item interval range between them is different, for example, for small weights of eggs and milk we think at the same time buy two goods within three days, there is relevance between them, but for larger weights of gold we think as long as there in three months and at the same time to buy that. In general, *max_interval* between the small weights items smaller than *max_interval* between the larger weights items. This requests us set different interval constraint conditions for different weights of items, namely introduce item interval constraint conditions related to the item's weights.

The concept of Generalized. The meaning of "generalized" in the generalized weighted closed sequential pattern mining algorithm with item interval includes four points:(1)can deal with two item interval measurement way, spacing and time interval;(2)by inserting pseudo interval item based on the interval piecewise function to get interval extension sequence;(3)use the item interval constraint conditions associated with the item weight, setting different item interval constraint corresponding

different weight of items;(4)for the resulting sequence patterns, get the standardization express of item interval of sequence pattern by adopt histogram statistics forms.

The extension main memory index set with item interval. *Type-1* pattern and *Type-2* pattern. Given a sequence pattern *p* and a weighted frequent items *x*, if the item set *(x)* as a new element into the *p*, the new pattern *p'* is a *Type-1* pattern. If join the *x* into the last element of *p*, the new pattern *p'* is a *Type-2* pattern. The weighted frequent items *x* called *stem* item of sequence *p'* (abbreviated to *stem*), p is called prefix pattern of *p'* (abbreviated to *P-pat*).

Definition 1: main memory index set with item interval *p-iidx*. In the *MDB*, for each data sequence *ds* which contains *p* distribute a ternary group *(ptr_ds, interval, pos)*, in the ternary group, *ptr_ds* is a pointer to the *ds*, *interval* is spacing interval of *p* in *ds*, *pos* refers to the position of *p* in *ds*. The main memory index set with item interval *p-iidx* in pattern *p* is a collection of all ternary groups. For example, <*da*>*-iidx* is main memory index set with item interval of sequence <*da*>.

3. The generalized weighted closed sequential pattern mining algorithm with item interval (GWCSpan)

Thought of Algorithm. firstly, according to the interval piecewise function to add item of pseudo item interval to get equivalent interval extended sequence database, and loaded the interval sequence database into main memory according to the size of main memory, after that, construct the main memory index set for every item which meet the k-minimum weighted support , and then, search item which meet conditions of minimum weighted support and item interval constraint to construct sequence pattern, make closed detection through hash structure, at last, for the resulting sequence patterns, get the standardization express of item interval of sequence pattern by adopt histogram statistics forms.

Algorithm 1: interval extension sequence database construction algorithm (DBChange)

Input: sequence database *SDB*

Output: interval extension sequence database *ISDB*

Begin

(1) For each sequence in the sequence database *SD*

(2) For each item in the sequence

(3) Get time of this item;

(4) timeOfitem←I(time(item));

(5) item←<timeOfitem, item>;

(6) Endfor

(7) Endfor

End

Algorithm 1 output interval extension sequence database, in algorithm, step 2-3 get corresponding timestamp of every item in sequence database, step 4 standardizing the item's change time, step 5 make standardized timestamp and corresponding item as a whole, construct the interval extension sequence database.

Algorithm 2: The generalized weighted closed sequential pattern mining algorithm with item interval (GWCSpan)

Input: sequence database *SDB*, minimum weighted support *minwsup*

Output: collection of generalized weighted closed sequential patterns *GWCSP*

Input: sequence database *SDB*, minimum weighted support *minwsup*

Output: collection of generalized weighted closed sequential patterns *GWCSP*

Begin

(1) Initialize collection of weighted closed sequential pattern *GWCSP*←{}, colletion of item which maybe construct weighted closed sequential pattern *tempS*←{};

(2) Call DBChange (*SDB*) to get the *ISDB* which corresponding *SDB* and read it into main memory;

(3) To find out each item in the database and its corresponding support count, and store Items and their support in the Items;

(4) Calculate the minimum weighed support count minSC.

(5) For each item in Items

(6) If (count<minSC) prune this item;

(7) Else if (WSup(item)>=*minwsup*) GWSCP←item; *tempS*←item;

(8) Else *tempS*←item;

(9) End if

(10) End if

(11) Endfor

(12) For each α in *tempS*

(13) Build index set of main memory with item interval recursively;

(14) In index set of main memory recursively mining weighted closed sequential patterns which meet interval constraint condition and minimum weighted support;

(15) Endfor

(16) For each s in the *WCSP*

(17) Use the statistic way Histogram to get the item interval of s.

(18) Endfor

(19) Return *GWCSP*

End

Algorithm 2 output the generalized weighted sequence pattern with item interval, in the algorithm, step 1 is initialized firstly, step 2 call subroutine DBChange to get the corresponding interval extension sequence ISDB and read it to the main memory, and step 3 calculate and store all support of items, step 4 calculate minimum weighted support of interval extension sequence database, step 5-6 directly delete item does not meet the minimum weighted support, step 7 to save items which weighted support is not less than the weighted support threshold into MIWCSP and tempS, step 8 to save items which support is greater than k-minimum weighted support and weighted support is not greater than the minimum weighted support into tempS, step 12-13 build index set of main memory with item interval recursively, step 14 recursively mining weighted closed sequential patterns which meet interval constraint condition and minimum weighted support in index set of main memory, step 16-18 use the histogram statistics form to obtain the item interval information of weighted sequential patterns, step 19 return the generalized weighted closed sequential patterns set eventually found.

4. The algorithm implementation and performance analysis

For process performance verification of algorithms in this paper, public authority data set must be used for experiment and comparative analysis. At website IBM Almaden (http://www.almaden.ibm.com/cs/quest/syndata.html) provides a standard data sets synthesizer, the data set use it to generate is widely used in many of the existing in the study of sequential pattern mining algorithm. Through setting the value of the parameters of synthesizer can generate the

corresponding synthetic data set. This paper's experiment data sets are to use the IBM data sets synthesizer to generate and add the item's weight and item interval.

Running time analysis of the algorithm. In this section, we through the experiment on the data set of D20I4D100K to analyze running time of the algorithm GWCSpan. Support set from 0.2% to 1%, the increase of 0.2%. Figure 1 and figure 2 shows the number of patterns and running time generated by algorithm under the condition of the different support.

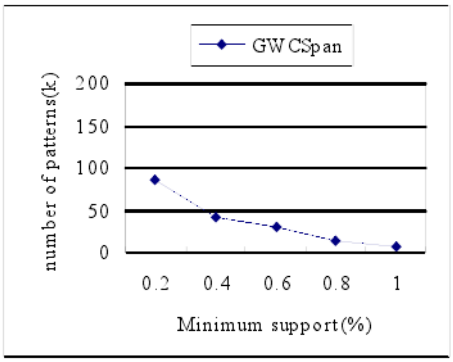

Figure 1. Number of patterns

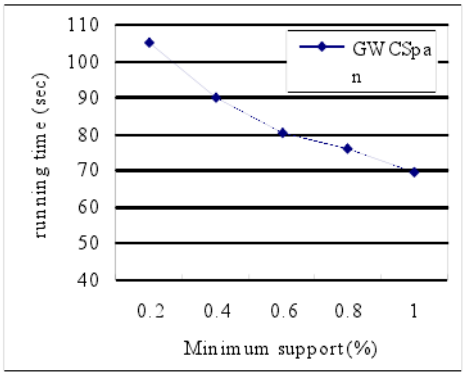

Figure 2. Running time

GWCSpan algorithm introducing weights constraint, at the same time, GWCSpan algorithm introducing the item interval constraint conditions related to the weights of item, in the mining process, algorithm to filter out the weighted sequential patterns which do not conform to the conditions by further pruning. It is makes the results of mining are more interesting for user and saves running time. Through the closed detection again in the mining process to filter out pattern not interested for user, to further reduce the running time of the algorithm.

Scalability testing. In this section, we use experiments on five data sets to verify scalability of GWCSpan algorithm. Data set size were 20k, 40k, 60k, 80k, 100k. Figure 3 shows the result of the experiment, of which the specified Minimum support is 0.3%. From figure can be found GWCSpan algorithm has a good scalability.

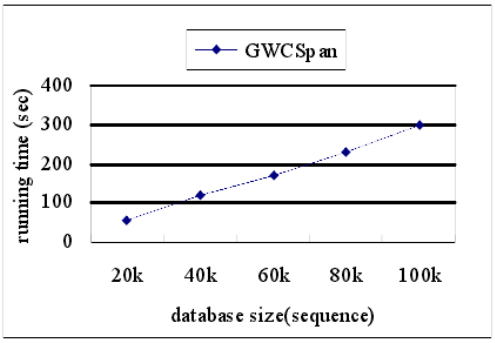

Figure 3. Result of scalability testing

GWCSpan algorithm used extension main memory index set structure with item interval in the process of mining, only scans the database one time, loads sequence database into main memory, do not need to generate a large number of candidate sets or build temporary database, reduce the waste of main memory space. With the size increase of the data sets, the execution time of the algorithm is a little change, so GWCSpan algorithm has a good scalability.

5. Summary

For overall consideration of item's weight and item interval in sequence, and make full use of the advantages of existing mining algorithm based on item interval constraint and mining algorithm based on extension sequence, we propose a generalized weighted sequential pattern mining algorithm with item interval(GWCSpan). Algorithm defines an item interval extension function that associated with the item weight, through the item interval piecewise function converts the item interval into

pseudo item, and added pseudo item into the original sequence database to make the equivalent extension sequence database. We introduce item interval constraints and weight constraints during the mining process for extension sequence database, and then obtained weighted sequential patterns which users really interested. Algorithm use a new structure of main memory index set, do not need to generate candidate set or build projection database to get implementation of weighted sequential pattern mining. Algorithm firstly uses item interval piecewise function converts item interval into the pseudo item to get equivalent sequence database with the original sequence database, according to the main memory size to load the extension sequence database into main memory, according to the item interval constraint conditions, direct delete the item which support less than the minimum weighted support. Then for each item that support is greater than k-minimum weighted support to build main memory index set with time interval, and then find item in index set that meet minimum weighted support to extend it. In the process of mining, for each weighted sequence pattern which weighted support is greater than minwsup, to obtained extended weighted sequential patterns by statistical histogram, and make closed detection by use hash structure. Experiments show that GWCSpan algorithm can get the extension sequential patterns with item interval which user is more interested, and has better scalability.

References

[1] Yun U, Leggett J J. WSpan:Weighted Sequential Pattern Mining in Large Sequence Databases. Proceedings of the 3rd International IEEE Conference On Intelligent Systems, (2006),pp.512-517; London, United kingdom.

[2] Chang J H, Park N H. Comparative Analysis of Sequence Weighting Approaches for Mining Time-interval Weighted Sequential Patterns. Journal of Expert Systems with Applications .39 (2012), pp.3867-3873.

[3] Pei J, Han J, and Wang W. Mining Sequential Pattern with Constraints in Large Database. Proceeding of CIKM'02,(2002),pp.18-25

[4] Kitakami H, Kanbara T, Mori Y. Modified Prefix Span Method for Motif Discovery in Sequence Databases. Proceeding of PRICAI2002,(2002),pp.482–491.

[5] Chen Y L, Chiang M C, Ko M T. Discovering Time-interval Sequential Patterns in Sequence Databases. Expert Syst. Appl., 3,25(2003),pp.343–354.

[6] Chen Y L, Huang T C. Discovering Fuzzy Time-interval Sequential Patterns in Sequence Databases. In IEEE Trans. on Systems, Man, and Cybernetics, 5,35(2005),pp.959–972.

[7] Hirate Y, Yamana H. Sequential Pattern Mining with Time Intervals. Proceeding of 10thPacific-Asia Conference on Knowledge Discovery and Data Mining,(2006),pp.775–779.