International Journal of Database Theory and Application
Vol.10, No.1 (2017), pp.23-36
http://dx.doi.org/10.14257/ijdta.2017.10.1.03

An Improved Sequential Pattern Algorithm Based on Data Mining

Jin Kao Zhao[1], Runtao Lv[2] and Yu Li[3]

[1, 2]Baotou light industry professional technology Institute College of electronic commerce, Baotou 014030, china
[3]Baotou city bureau of education test center, Baotou 014030, china
[1]1223882175@qq.com, [2] 2538155109@qq.com and [3]2538155109@qq.com

Abstract

This paper mentions several interestingness measures as Lift, Conviction, Piatetsky-Shapiro, Cosine, Jaccard and so on, which have proposed for mining association rules and classification rules but they have not been applied to mine sequential rules in sequence databases except the traditional measures of rule such as the support and confidence. We also propose then an efficient algorithm to generate all relevant sequential rules with the above interestingness measures from the prefix-tree which stored the whole sequential pattern where each child node stores a sequential pattern and its corresponding support value. By traversing the prefix-tree, the algorithm can then easily identify the components of a rule, and can calculate the measured values of the rule. The experimental results show that sequential rule mining with interestingness measures using the proposed algorithm based on the prefix-tree was always much faster than that using the other existing algorithm as modified Full. Especially when mining in large sequence databases with the low minimum support values, the number of sequential patterns generated from sequence databases was large and the proposed algorithm outperformed much because the proposed algorithm only traverse the prefix-tree to immediately determine which sequences are the left- and right-hand sides of a rule as well as their support values to compute the interestingness measure values of the rule from the sequential pattern set. In addition, the experimental results also show that the time for mining sequential rules with the confidence measure was the smallest, because it did not need to revisit the prefix-tree to determine the support of Y (the antecedence of rules), while the other interestingness measures need to revisit the prefix-tree to determine the support values of the consequent of rules or both the antecedence and the consequent.

***Keywords**: Sequential pattern, interestingness measure, sequential rule, prefix-tree*

1. Introduction

Mining sequential rules are an important problem in data mining research. It is commonly used for market decisions, management and behaviour analysis. In traditional association-rule mining, rule interestingness measures such as confidence are used for determining relevant knowledge. They can reduce the size of the search space and select useful or interesting rules from the set of discovering ones. Many studies have examined the interestingness measures for evaluating association rules and classification rules [1-6], but have not been devoted to mine sequential rules in sequence databases except the traditional measures of rule such as the support and confidence [7-11], which was specifically described in Section 2.5. In this chapter, we thus consider and apply several interestingness measures to generate all relevant sequential rules from a sequence database. The prefix-tree structure is also used to compute the interestingness measure values of sequential patterns faster and reduce the execution time for mining sequential rules.

2. Problem Statement

A sequential rule $X \Rightarrow Y$ is defined as a relationship between itemsets $X, Y \in I$ such that

$X \cap Y \in \varnothing$ and X, Y are not empty, described as "if itemset X appears in any sequence of the sequence database then itemset Y is likely to appear in that sequence following X with a given confidence afterward". The overall measured value of the rule is determined when the following measure values including the supports of X, Y and XY are determined. Given the frequent sequential patterns of X and Y, there is a sequential rule $X \Rightarrow Y$, if its confidence satisfies the minimum confidence threshold. The confidence of a sequential rule $X \Rightarrow Y$ is the ratio of the number of sequences that contain both X and Y against the number of those that contain X.

Similar to the association rule mining problem, we also divide the sequential rule mining using interestingness measures from a sequence database into two stages. The first stage is to mine all sequential patterns that satisfy the user-specified minimum support threshold minSup. The next stage is to generate all the sequential rules with their interestingness measures from the above set of sequential patterns. To efficiently mine sequential patterns in the first stage, the PRISM algorithm is adopted, which uses the prime block encoding approach to represent candidate sequences and the join operations over the prime blocks to determine the frequency for each candidate. All the sequential patterns generated by the PRISM algorithm are stored in a prefix-tree structure.

A prefix-tree is used in this work similar to the prefix-tree described. However, in this prefix-tree, the root at level 0 is set to a null sequence $\varnothing$, and each child node stores a sequential pattern and its corresponding support value. Figure1 shows the prefix-tree of sequential patterns generated from the sequence database. Sequences <(A)(B)>and<(A)(C)>are sequence-extended sequences of <(A)>, and <(AB)>is an itemset-extended sequence of <(A)>. Sequence <(A)> is a prefix of all the sequences in T1 and an incomplete prefix of all the sequences in T2. Similarly, a sequence<(B)> has the three sequences-extended sequences<(B)(A)>,<(B)(B)>and<(B)(C)>, and the one itemset-extended sequence<(BC)>. Sequence<(B)>is a prefix of all the sequences in T3 and an incomplete prefix of all the sequences in T4.

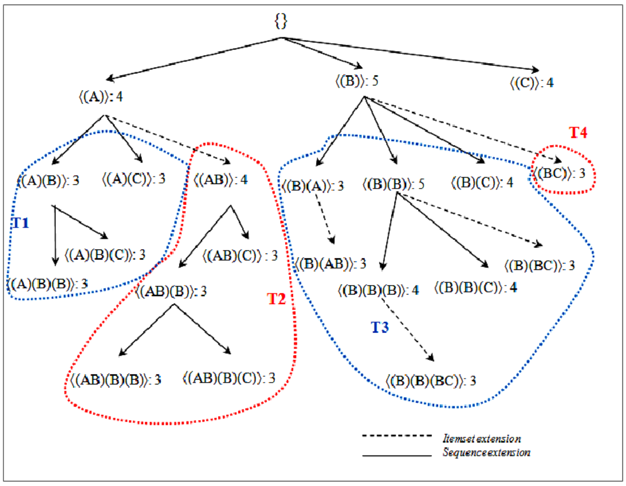


Figure 1. A Prefix-Tree Structure Storing Sequential Patterns

3. Mining Sequential Rules with Interestingness Measures

A sequential rule has the form $X \Rightarrow Y(q, imv)$, where X and Y are sequential patterns, $X \cap Y = \varnothing$, q is the support of the rule Problem statement, and imv is an interestingness measure value of the rule. In the traditional sequential rules, imv is the confidence of a rule, and $imv = Sup(X, Y) / Sup(X)$.

A sequential rule can be created by splitting a sequential pattern into two parts: the prefix (pre) and the postfix (post). If the pre is concatenated with post, denoted pre++post, then the result is the original sequential pattern. A sequential rule r can thus be formed as pre post (Sup, imv). The support Sup(r) of r is thus Sup(pre++post). The interestingness measure value of r is imv, and the traditional measure value of r is the confidence measure Conf(r) of r. That is, Conf(r) = Sup(pre++post)/Sup(pre). A sequence of size k has (k-1) prefixes, and can thus have (k-1) sequential rules. For example, if there is a sequential pattern <(A)(BC)(D)>□ whose size is 3, then 2 rules will be generated:<(A) $\Rightarrow$ (BC)(D)>, and <(A)(BC)$\Rightarrow$< (D)>.□.

2.1. Interestingness Measures

Interestingness measures are important metrics for rule mining in the data mining research. They can be used to reduce the search space size and thus improve mining efficiency, or to rank patterns according to the arrangement of their interestingness values. Besides, they play an important role in selecting useful or interesting rules from a set of discovering rules. For example, we can use the support threshold to remove patterns with low support and the confidence threshold to select all rules that have significant associations during the mining process and thus improve efficiency. The interestingness measures can be classified into two categories: subjective and objective. Subjective measures explicitly depend on the user's goals and his/her knowledge or beliefs; they are combined with specific supervised algorithms in order to compare the extracted rules with the user's expectations [12-13]. Consequently, subjective measures allow the capture of rule novelty and unexpectedness in relation to the user's knowledge or beliefs. While objective measures are numerical indexes that only rely on the data distribution. Interestingness refers to the degree to which a discovered pattern is of interest to the user and is driven by factors such as novelty, utility, relevance and statistical significance [14-15]. So, in this thesis, we only focus on several interestingness measures of the objective measures. Many studies have examined interestingness measures to mine rules, including support, confidence, cosine, lift, x^2, gini-index, Laplace, and phi-coefficient [16-19] and so on. Table1 shows some interestingness measures. However, to the best of our knowledge, these measures have been used for mining association rules in transaction databases [20-22] but have not been used to mine sequential rules in sequence databases except the traditional measures of support and confidence. Thus, the first aim in our works is to apply these different metrics to the sequential rule mining problem.

Table 1. Some Interestingness Measures for a Rule $X \Rightarrow Y$

Interestingness	measure	Equation	Value
Confidence	$\frac{n_{XY}}{n_X}$	$\frac{3}{5}$	[33.37,50]
Support	$\frac{n_{XY}}{n}$	$\frac{3}{5}$	[33.37,50]
Conviction	$\frac{n_X n_{\bar{Y}}}{n n_{X\bar{Y}}}$	$\frac{5*2}{5*2}=1$	[33.37,50]
Lift	$\frac{n n_{XY}}{n_X n_Y}$	$\frac{5*3}{5*3}=1$	[33.37,50]
Piatetsky-Shapiro	$n_{XY}-\frac{n_X n_Y}{n}$	$3-\frac{5*3}{5}$	[33.37,50]
Cosine	$\frac{n_{XY}}{\sqrt{n_X n_Y}}$	$\frac{3}{\sqrt{5*3}}$	[33.37,50]
Jaccard	$\frac{n_{XY}}{n_X+n_Y-n_{XY}}$	$\frac{3}{5+3-3}=\frac{3}{5}$	[33.37,50]

From the equations in Table1, it can be easily observed that the terms often used to calculate a measured value of the rule $X \Rightarrow Y$ are the total number of sequences in a sequence database (n), the number of sequences that contain $X(n_x)$, the number of sequences that contain $Y(n_Y)$, the number of sequences that contain both X and $Y(n_{XY})$, the number of sequences that contain X but not $Y(n_{X\bar{Y}})$, and the number of sequences that contain Y but not $X(n_{\bar{X}Y})$. If we know n, n_x, n_Y, and n_{XY}, other terms for calculating the measured value in these equations can be easily determined like $n_{X\bar{Y}} = n_X - n_{XY}$, $n_{\bar{X}Y} = n_Y - n_{XY}$. Consider the sequence database in Table 1. If X =<(B)>□and Y = <(B)(BC)>, then N=5, $n_X = 5, n_Y = 3$ and $n_{XY} = 3$ .Table1 also presents the interestingness measure values for the rule $X \Rightarrow Y$.

2.2 Algorithm

The previous algorithm PRISM [23] is first applied to generate sequential patterns stored in the prefix-tree structure. An algorithm based on the characteristics of the prefix-tree is then proposed to generate sequential rules with interestingness measures. By traversing the prefix-tree, the algorithm can then easily identify the components of a rule, such as the pre and the post parts, and can calculate the measured values of the rule. Figure2 presents the proposed algorithm to mine sequential rules with interestingness measures.

In Algorithm1, the algorithm first calls the PRISM(SD, minSup) procedure to generate all sequential patterns and store these patterns in the prefix-tree structure. For each node SP at level 1 of the prefix-tree, it calls the GENERATE_SR_FROM_TREE_ROOT(SP_Root) procedure to generate sequential rules from each sub-tree with SP as its root node. When the procedure GENERATE_SR_FROM_TREE ROOT(SP_Root) is processed, there are two types of nodes: sequence-extended and itemset-extended nodes. Based on the definition of itemset extension then the size of the itemset-extended nodes set does not change w.r.t the size of the root node and the root node becomes an incomplete prefix of the all its itemset-extended nodes. To pruning the search space, this procedure do not generate sequential rules from the root node SP_Root to its itemset-extended nodes set, and only sequential rules from sequences on the subtrees whose nodes are sequence-extended nodes of the root are generated from the called procedure GENERATE_SR_FROM_SUBTREE(Pre, Subtree), because the sequence at the root denoted as pre will form the prefix of all extended sequences from the sequence-extended nodes of the root. Hence, for each sub–tree, sequential rules from the sequences on the subtree following the prefix pre are generated. All the extended nodes of the current root then become prefixes of the subtrees at the next level, and this procedure is recursively called for every extended node of the root. This recursive process is repeated until the last level of the prefix-tree is reached. Besides, in the procedure GENERATE_SR_FROM_SUBTREE (Pre, Subtree), the input is sequences pre and Subtree so that pre is a common prefix of all the sequences on the subtree. For each sequence SP in the subtree, the rule “pre $\Rightarrow$ post” is generated such that the post is a postfix of SP with respect to the prefix pre.

Most of the interestingness measures (imv) for a rule depend on the support (n_{Post}) of the Post. To obtain the support of the Post, the procedure FIND_SUP_POST (RNode, Post) is called, where RNode is a not-empty and the first root node of the Post on the prefix-tree. The procedure FIND_SUP_POST(RNode, Post) produces the support of the Post by traversing the branch of the prefix-tree based on the root node RNode, which is the prefix of the Post.

Algorithm1 The proposed algorithm for generating sequential rules based on a prefix-tree

```
Input: A sequence database SD, minimum support minSup, and minimum interestingness measure
minThreshold.
Output: A set of sequential rules SRs and their measure values.
Method:
        Call the procedure PRISM(SD, minSup) in [23] to generate sequential patterns stored in a prefix tree.
        SRs = $\varnothing$; //for storing the set of sequential rules
        L1 = All nodes at level 1 of the prefix tree;
        For each node SP in L1
        Call the procedure GENERATE_SR_FROM_TREE_ROOT(SP) to generate sequential
        rules from the root of a subtree with the root node of the subtree SP;
        Return SRs;
//Generating sequential rules from a root node on the tree.
 GENERATE_SR_FROM_TREE_ROOT(SP_Root)
            Let Sequence_ext_pattern = Sequence extensions of SP_Root;
            Let Itemset_ext_pattern = Itemset extensions of SP_Root;
            For each node PSeq in SP_Root.Sequence_ext_pattern do
           Let Subtree = the subtree with its root node at PSeq;
              GENERATE_SR_FROM_SUBTREE(SP_Root, Subtree);
           For each node PItems in SP_Root.Itemset_ext_pattern do
              GENERATE_SR_FROM_TREE_ROOT(PItems);
          For each node PSeq in SP_Root.Sequence_ext_pattern do
           GENERATE_SR_FROM_TREE_ROOT(PSeq);
        // Generating all rules for the sequences on the subtree with a given prefix
         GENERATE_SR_FROM_SUBTREE(Pre, Subtree)
            Let n be the total number of sequences in the sequence database;
            Let $n_{\mathrm{Pre}}$ be the support of Pre;
            For each node n in Subtree
              Let SP be the sequence kept at node n;
              Set Post = SP – Pre // re presenting a postfix of SP w.r.t the prefix Pre;
            Generate a rule R = “Pre $\Rightarrow$ Post”;
              Let $n_R$ = the support of SP;
              Let RNode be the first root node of Post;
              Set $n_{Post}$ = FIND_SUP_POST(RNode, Post); //getting the support of Post
        Calculate the interestingness measure value $imv_R$ of the rule R from n, $n_{\Pr e}$, $n_{Post}$ and
        $n_R$; //depending on the formula used
            If ($imv_R$ >= minthreshold)
        Add rule R($n_R$, $imv_R$) to SRs;
// find the support of Post in the rule “Pre □ Post”
FIND_SUP_POST(RNode, Post)
          If sequence Post == the sequence at RNode then
          return the support of RNode;
            LetSequence_ext_pattern be the sequence extensions of RNode;
            Let Itemset_ext_pattern be the itemset extensions of RNode;
            For each node PSeq in RNode.Sequence_ext_pattern
           If PSeq is a prefix of Post then
                FIND_SUP_POST(PSeq, Post);
            For each node PItems in RNode.Itemset_ext_pattern
        If PItems is a prefix of Post then
                FIND_SUP_POST(PItems, Post);
```

2.3. Illustration

An example is given here to illustrate the above algorithm. Consider the sequence database presented in Table 2, with minSup=50%. Table 2 shows the results of the sequential rules generated from the prefix-tree with the different interestingness measures.

Note that when the minimum interestingness measure threshold minThreshold is 0, for all of the different interestingness measures are equal (totally 23 sequential rules) as shown in Table3. However, when the minimum interestingness measure threshold minThreshold is greater than 0, the numbers of sequential rules generated are different which depend on the generality, confident, reliability of the rule and the correlation between antecedence and consequent of the rule for each measure. For example, if the minimum interestingness measure for minConfidence, minLift, and minCosine are set at 0.8, then 10 sequential rules satisfy minConfidence, 17 sequential rules satisfy minLift and only 6 sequential rules satisfy minCosine generated as shown in Table 4. To quickly get the support (n_{Post}) of the right-hand side of the rule, the algorithm only needs to traverse the branch of the prefix-tree based on the root nodes that are the prefixes <(A)> has one itemset-extended sequence <(AB)>and two sequence-extended sequences<(A)(B)>and <(A)(C)>. Because <(A)>is an incomplete prefix of <(AB)>and all sub-nodes of <(AB)> which extended from <(AB)>, the algorithm does not need to generate rules from the nodes with prefix <(A)>. On the contrary, since <(A)> is a prefix of the two sequence-extended sequences <(A)(B)> and <(A)(C)>, the following rules can be generated: <(A)$\Rightarrow$(B)> and <(A)$\Rightarrow$(C)>. For the sequential rule <(A)$\Rightarrow$(B)>, since the support value of the sequential pattern B is 5 by traversing the prefix-tree and the calculated Lift measure value of the rule in Table1 is less than minLift, the rule <(A)$\Rightarrow$(B)>is not generated. Similarly for the rule<(A)$\Rightarrow$(C)>, since the support value of the sequential pattern C is 4 and the calculated Lift measure value in Table 1 is 0.9375, which is greater than minLift, the sequential rule <(A)⇒(C)> (5, 4, 4, 3) is generated. Moreover, <(A)>is a prefix of all the sub-nodes of<(A)(B)> and <(A)(C)>, such that the algorithm can generate rules as well from the subnodes in a similar process, the subnodes include sequences<(A)(B)(B)>and <(A)(B)(C)>. The above generating sequential rules process is applied for two these subnodes and only <(A)⇒(B)(C)>(5, 4, 4, 3) sequential rule is generated. The above process can then be repeated for all the sub-nodes of <(A)> to generate sequential rules. The results are shown in Table4.

Table 2. An Example Sequence Database (SD)

SID	Sequence
1	□<(AB)(B)(B)(AB)(B)(AC)>□
2	□<(AB)(BC)(BC)>□
3	□<(B)(AB)>□
4	□<(B)(B)(BC)>□
5	□<(AB)(AB)(AB)(A)(BC)>□

Table 3. The Sequential Rules Generated for Any Interestingness Measures in Table 3.1 With

Prefix	Rules
<(A)>□	(A) ⇒B)□; (A) ⇒ (C)□; □(A) ⇒(B)(B)□; □(A) ⇒ (B)(C)□
<(A)(B)> □□	□ □(A)(B) ⇒ (B)□; □(A)(B) ⇒ (C)
□ <(AB)>□ □	(AB) ⇒ (B); (AB) ⇒ (C); (AB)⇒(B)(B)□; □(AB) ⇒ (B)(C)□;□
□ <(AB)(B)> □	□□(AB)(B) ⇒ (B)□; □(AB)(B) ⇒ (C)□;
□ <(B)> □ □	(B) ⇒ (A)□; (B) ⇒ (B); (B) ⇒ (C); (B) ⇒ (AB)□; □(B) ⇒ (BC)□; (B) ⇒ (B)(B); (B) ⇒ (B)(C); (B)⇒(B)(BC)□;
<(B)(B)> □	□(B)(B) ⇒ (B)□; □(B)(B) ⇒ (C)□; □(B)(B) ⇒ (BC)□;

Table 4. The Sequential Rules with Minthreshold = 0.8

Prefix	Rules with the confidence measure $(\sup, imv)$	Rules with the lift measure $(\sup, imv)$	Rules with the cosine measure $(\sup, imv)$
□<(A)>□	□	<(A)> $\overset{(3.093)}{\Rightarrow}$ <(C)>(5,4,4,3) <(A)> $\overset{(3.093)}{\Rightarrow}$ <(B)(C)>(5,4,4,3)	
<(A)(B)>□	<(A)(B)> $\overset{(3.1)}{\Rightarrow}$ <(B)>(5,3,5,3) <(A)(B)> $\overset{(3.1)}{\Rightarrow}$ <(C)>(5,3,4,3)	<(A)(B)> $\overset{(3.1)}{\Rightarrow}$ <(B)>(5,3,5,3) <(A)(B)> $\overset{(3.1)}{\Rightarrow}$ <(C)>(5,3,4,3)	<(A)(B)> $\overset{(3.088)}{\Rightarrow}$ <(C)>(5,3,4,3)
<(AB)>□		<(AB)> $\overset{(3.093)}{\Rightarrow}$ <(C)>(5,4,4,3) <(AB)> $\overset{(3.093)}{\Rightarrow}$ <(B)(C)>(5,4,4,3)	
<(AB)(	<(AB)(B)> $\overset{(3.1)}{\Rightarrow}$ <(B)>(	<(AB)(B)> $\overset{(3.1)}{\Rightarrow}$ <(B)>(5,3,5,3)	<(AB)(B)> $\overset{(3.088)}{\Rightarrow}$ <(C

B)> □	5,3,5,3) <(AB)(B)> $\overset{(3.1)}{\Rightarrow}$ <(C)>(5,3,4,3)	<(AB)(B)> $\overset{(3.1)}{\Rightarrow}$ <(C)>(5,3,4,3)	)>(5,3,4,3)
□ <(B)> □	<(B)> $\overset{(5.1)}{\Rightarrow}$ <(B)>(5,5,5,5) <(B)> $\overset{(4.08)}{\Rightarrow}$ <(C)>(5,5,4,4)	<(B)> $\overset{(5.1)}{\Rightarrow}$ <(B)>(5,5,5,5) <(B)> $\overset{(4.1)}{\Rightarrow}$ <(C)>(5,5,4,4)	<(B)> $\overset{(5.1)}{\Rightarrow}$ <(B)>(5,5,5,5) <(B)> $\overset{(4.088)}{\Rightarrow}$ <(C)>(5,5,4,4)
<(B)> □	<(B)(B)> $\overset{(4.08)}{\Rightarrow}$ <(B)>(5,5,5,4) <(B)(B)> $\overset{(4.08)}{\Rightarrow}$ <(C)>(5,5,4,4)	<(B)(B)> $\overset{(4.08)}{\Rightarrow}$ <(B)>(5,5,5,4) <(B)(B)> $\overset{(4.1)}{\Rightarrow}$ <(C)>(5,5,4,4)	<(B)(B)> $\overset{(4.088)}{\Rightarrow}$ <(C)>(5,5,4,4)

3. Experiment Design and Discussion

Experiments were then made to evaluate the performance of the proposed algorithm for sequential rule mining using different interestingness measures. An algorithm modified from the Full algorithm [7], called modified Full, for generating only traditional sequential rules by using the confidence measure was also run for comparison. All the experiments were performed on a PC machine with dual-core 2.81 GHz, 2 GBs RAM, running Windows XP professional, and implemented by C#. The synthetic databases were generated by the IBM synthetic data generator to mimic transactions in a retail environment. The synthetic data generation program used the following parameters: C was the average number of itemsets per sequence, T was the average number of items per itemset, S was the average number of itemsets in maximal sequences, I was the average number of items in maximal sequences, N was the number of distinct items, and D was the number of sequences.

Two synthetic databases, C6T5S4I4N1kD1k and C6T5S4I4N1kD10k, were used in the experiments. In the databases, the number of items was set to 1,000 (denoted as N1k). There were 1,000 sequences in the C6T5S4I4N1kD1k database (denoted as D1k) and 10,000 sequences in the C6T5S4I4N1kD10k database (denoted as D10k). The average number of items within itemsets was set to 5 (denoted as T5), the average number of itemsets in maximal sequences was set to 4 (denoted as S4), the average number of items in maximal sequences was set to 4 (denoted as I4), and the average number of itemsets in sequences was set to 6 (denoted as C6). The results are shown in Table5.

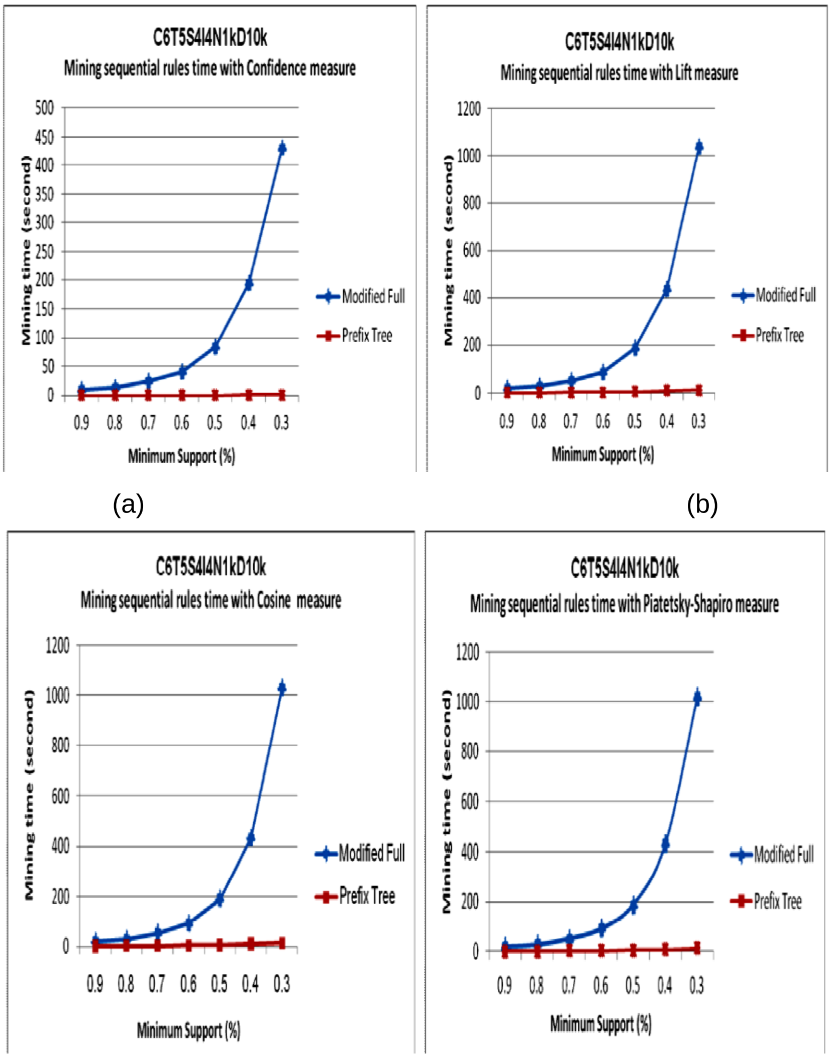
Table 5. The Time Ratios for Different Interestingness Measures

Database	minSup (%)	Number of Sequential Patterns	Number of Rules	Scale of Prefix Tree w.r.t modified Full in Confidence measure (%)	Scale of Prefix Tree w.r.t modified Full in Lift measure (%	Scale of Prefix Tree w.r.t modified Full in Cosine measure (%	Scale of Prefix Tree w.r.t modified Full in Piatetsky - Shapiro measure (%	Scale of Prefix Tree w.r.t modified Full in Conviction measure (%	Scale of Prefix Tree w.r.t modified Full in Jaccard measure (%
C6T5S4I4N1kD10k	0.9	6980	8345	0.47	7.05	5.92	5.93	5.96	5.94
	0.8	8340	10576	0.39	5.38	5.32	5.41	5.34	5.31
	0.7	10480	13582	0.25	4.23	4.20	4.31	4.15	4.11
	0.6	13628	18313	0.17	3.54	3.39	3.39	3.41	3.41
	0.5	18461	25848	0.13	2.62	2.57	2.65	2.57	2.47
	0.4	27168	39661	0.09	2.01	2.01	2.01	2.02	2.04
	0.3	44584	67808	0.09	1.19	1.19	1.22	1.21	1.21
C6T5S4I4N1kD1k	0.9	8795	11214	0.29	5.00	4.88	4.76	4.84	4.85
	0.8	11211	14815	0.23	4.20	4.22	4.12	4.16	4.11
	0.7	14802	20224	0.17	2.96	2.87	2.87	2.90	2.90
	0.6	20644	29364	0.11	2.11	2.13	2.15	2.13	2.12
	0.5	31311	46577	0.08	1.61	1.57	1.59	1.57	1.58
	0.4	54566	85846	0.04	1.10	1.11	1.12	1.13	1.13
	0.3	124537	214445	0.02	0.75	0.67	0.67	0.65	0.66

Table 5 shows numbers of sequential patterns, numbers of sequential rules with interestingness measures, and the execution time ratio in the two synthetic databases, C6T5S4I4N1kD1k and C6T5S4I4N1kD10k, corresponding to their minimum supports and different rule measures between the proposed algorithm and the modified Full algorithm.

Figure2 and Figure3 compare the sequential rule mining times with interestingness measures between the modified Full algorithm and the proposed algorithm, according to the prefix-tree structure in the two synthetic databases. The results in Figures1 and Figures1 (a) compare the sequential rule mining times using the Confidence measure. Figure2 and Figure3, (c), (d), (e) and (f) are for the Lift, Cosine, Piatetsky-Sharipo, Conviction and Jaccard measures, respectively. The experimental results from Figure2 and Figure3 show that sequential rule mining with interestingness measures using the proposed algorithm based on the prefix-tree was always much faster than that using the modified Full algorithm. The former only consumed a small amount of time when compared with the latter. The time ratio was calculated as follows: (mining time on the prefix-tree / mining time on the modified Full) *100%. For the C6T5S4I4N1kD1k dataset with minSup = 0.5% and the confidence measure, the mining time based on the Prefix-tree was 0.22, and based on the Full algorithm was 265.77, such that the time ratio was (0.22/165.77)*100%, which was 0.08%. If the Lift measure was used, the time ratio was (9.47/588.92)*100%, which was 1.61%. Similarly, the time ration for the cosine measure was (9.35/595.27)*100%, which was 1.57%, for the Piatetsky-Sharipo measure was

(9.39/592.15)*100%, which was 1.59%, for the conviction measure was (9.38/596.41)*100%, which was 1.57%, and for the Jaccard measure was (9.4/594.09)*100%, which was 1.58%. Among all the above time ratios, the one for the confidence measure was the smallest, because it did not need to revisit the prefix-tree to determine the support of Y (the antecedence of rules), while the other interestingness measures need to revisit the prefix-tree to determine the support values of the consequent of rules or both the antecedence and the consequent. Table5 shows the time ratio of these measures as well as the number of sequential rules generated with different minimum supports. According to the results in Figure2, Figure3 and Table5, it could be easily seen that for low minimum support values, the number of sequential rules generated from sequence databases was large and the proposed algorithm outperformed the modified Full algorithm much. Though the modified Full algorithm had to scan a set of sequential patterns to determine the support of the right-hand side of each rule, the proposed algorithm only traversed the branch of the prefix-tree based on the root nodes that were the prefixes of the sequence on the right-hand side of each rule.



(a) (b)

(c) (d)

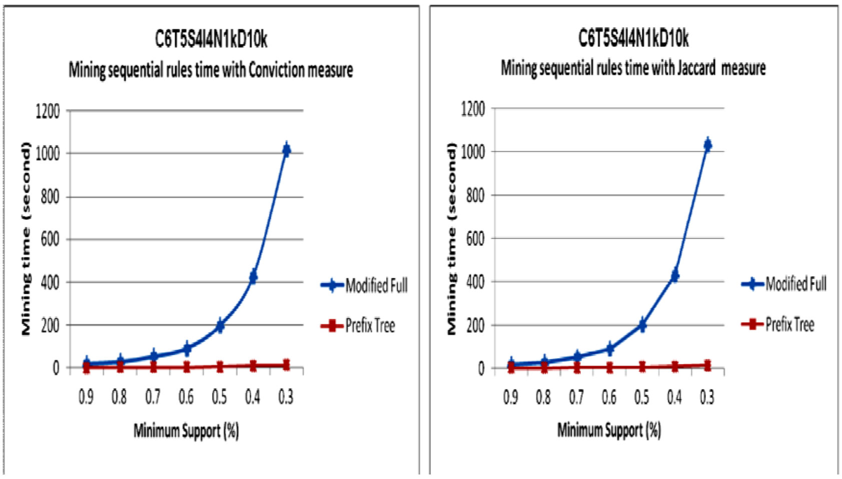
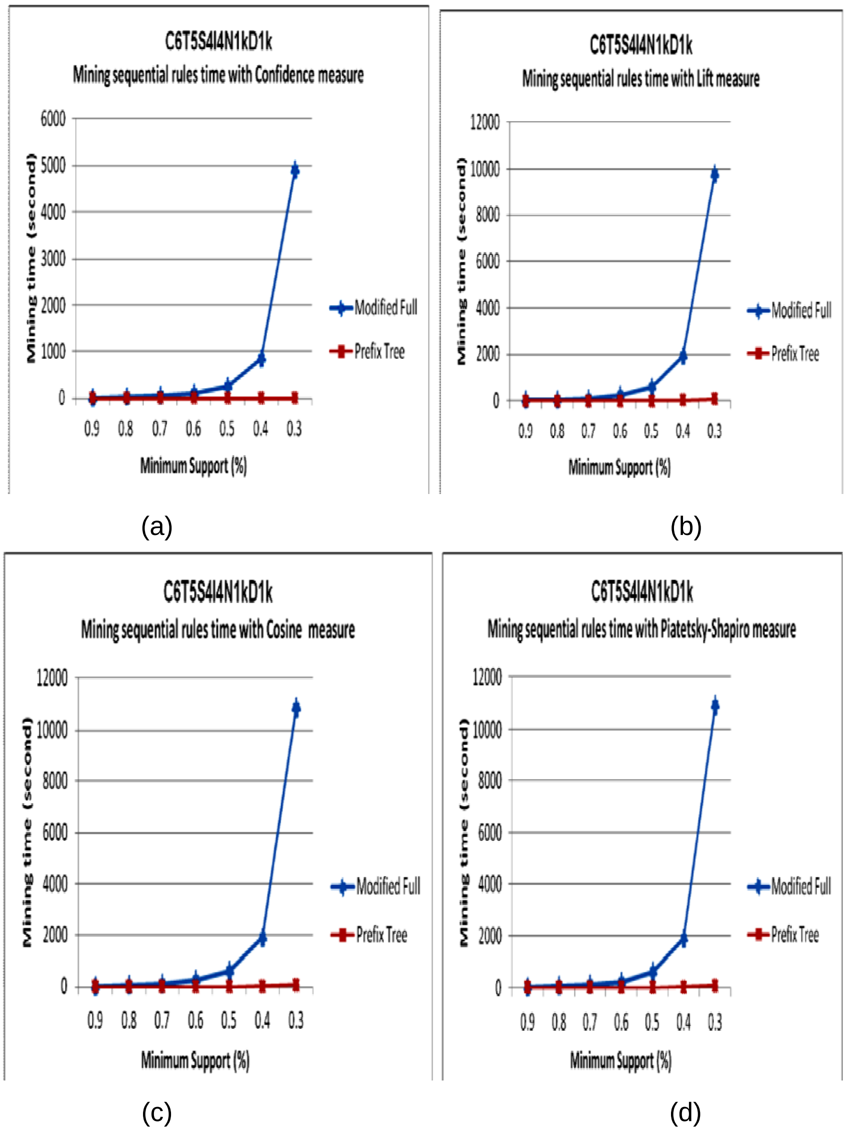


Figure 2. The Mining Times of the Two Algorithms for Different Interestingness Measures in C6t5s4i4n1kd10k

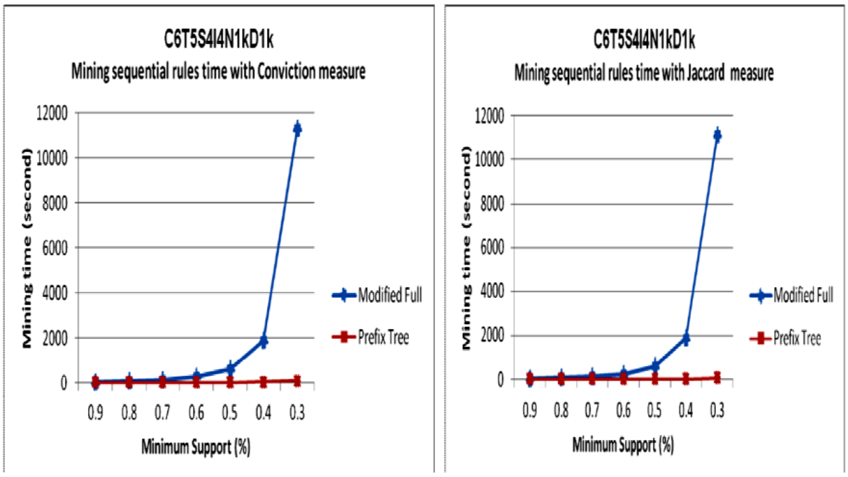


(a)

(b)

(c)

(d)



(e) (f)

Figure 3. The Mining Times of the Two Algorithms for Different Interestingness Measures in C6t5s4i4n1kd1k

4. Conclusion

In this paper, we have considered and applied several interestingness measures to mine sequential rules from a set of sequential patterns in sequence databases. In large sequence databases, the determination of measured values becomes difficult, and the time required to compute measure values and generate rules is long. The prefix-tree structure is also used to compute the values fast and to reduce the time for mining sequential rules. By traversing the prefix-tree, the proposed approach can immediately determine which sequences are the left- and right-hand sides of a rule as well as their support values to compute the interestingness measure values of the rule from the sequential pattern set. The experimental results show that the performance of the proposed algorithm for mining sequential rules with different interestingness measures on the prefix-tree structure is much better than that of the modified Full algorithm.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under grant No. 61163025 and the natural science foundation of the Inner Mongolia autonomous region No. 2010BS0904.

References

[1] L. Geng and H. J. Hamilton, “Interestingness Measures for Data Mining: A Survey”, ACM Computing Surveys, vol. 38, no. 3, **(2006)**.

[2] H. X. Huynh, “Interestingness Measures for Association Rules in A KDD Process: Post-processing of Rules with Arqat Tool”, Ph D Thesis, Universit E De Nantes, **(2006)**.

[3] B. Vo and B. Le, “Interestingness measures for association rules Combination between lattice and hash tables”, Expert Systems with Applications, vol. 38, no. 9, pp. 11630-11640.

[4] I. N. M. Shaharanee, F. Hadzic and T. S. Dillon, “Interestingness measures for association rules based on statistical validity”, Knowledge-Based Systems, vol. 24, no. 3, **(2011)**, pp. 386-392.

[5] H. X. Huynh, F. Guillet, J. Blanchard, P. Kuntz, R. Gras and H. Briand, “A graph based clustering approach to evaluate interestingness measures a tool and a comparative study”, Quality measures in data mining. Springer-Verlag, vol. 43, **(2007)**, pp. 25-50.

[6] P. N. Tan, V. Kumar and J. Srivastava, “Selecting the right interestingness measure for association patterns”, Proceeding of the ACM SIGKDD international conference on knowledge discovery in databases (KDD’02), **(2002)**, pp. 32-41.

[7] D. Lo, S. C. Khoo and L. Wong, “Non-redundant sequential rules-theory and algorithm”, Information Systems, vol. 34, no. 4/5, **(2009)**, pp. 438-453.

[8] H. Zang, Y. Xu and Y. Li, "Non-Redundant Sequential Association Rule Mining and Application in Recommender Systems", Proceeding of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, DC, USA, vol. 3, **(2010)**, pp. 292-295.

[9] M. Spiliopoulou, "Managing interesting rules in sequence mining", Proceeding of European Conference on Principles of Data Mining and Knowledge Discovery, **(1999)**, pp. 554-560.

[10] T. T. Van, B. Van and B. Le, "Mining sequential rules based on prefix-tree", Studies in Computational Intelligence (Springer), vol. 351, **(2011)**, pp. 147-156.

[11] A. Silberschatz and A. Tuzhilin, "What makes patterns interesting in knowledge discovery systems", IEEE Transactions on Knowledge and Data Engineering, vol. 5, no. 6, **(1996)**, pp. 970-974.

[12] B. Liu, W. Hsu, L. F. Mun and H. Y. Lee, "Finding interesting patterns using user expectations", IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 6, **(1999)**, pp. 817-832.

[13] B. Padmanabhan, and A. Tuzhilin, "A belief-driven method for discovering un-expected patterns. KDD'98", Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, **(2012)**, pp. 94-100.

[14] W. J. Frawley, G. P. Shapiro and C. J. Matheus, "Knowledge discovery in databases an overview", Knowledge Discovery in Databases, **(1991)**, pp. 1-27.

[15] G. P. Shapiro and C. J. Matheus, "The interestingness of deviations", AAAI'94, Knowledge Discovery in Databases Workshop, **(1994)**, pp. 25-36.

[16] R. Agrawal, T. Imielinski and A. N. Swami, "Mining association rules between sets of items in large databases", Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data, **(1993)**, pp. 207-216.

[17] P. Lenca, P. Mayer, B. Valliant and S. Lallich, "On selecting interestingness measures for association rules User oriented description and multiple criteria decision aid", European Journal of Operational Research, vol. 1842, **(2008)**, pp. 610-626.

[18] J. Aze′ and Y. Kodratoff, "A study of the effect of noisy data in rule extraction systems", Proceeding of the Sixteenth European Meeting on Cybernetics and Systems Research, **(2002)**, pp. 781-786.

[19] R. Hilderman and H. Hamilton, "Knowledge discovery and measures of interest", Kluwer Academic Publishers, **(2001)**.

[20] R. A. Huebner, "Diversity-based interestingness measures for association rule mining", In Proceedings of ASBBS, Las Vegas, vol. 16, no. 1, **(2009)**.

[21] P. Shapiro, "Discovery, analysis, and presentation of strong rules", Knowledge Discovery in Databases, **(1991)**, pp. 229-248.

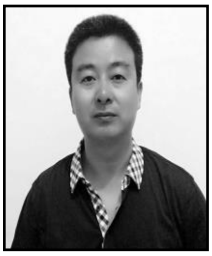
[22] P. N. Tan, V. Kuma and J. Srivastava, "Selecting the right objective measure for association analysis", Information Systems, vol. 29, no. 4, **(2004)**, pp. 293-313.

[23] K. Gouda, M. Hassaan and M. J. Zaki, "PRISM: A primal-encoding approach for frequent sequence mining", Journal of Computer and System Sciences, vol. 76, no. 1, **(2010)**, pp. 88-102.

Authors

Jin Kao Zhao, He received his B.S degree from Inner Mongolia normal university computer science education. He is an associate professor in baotou light industry professional technology institute college of electronic commerce. His research interests include computer network.

Runtao Lv, She received her B.S degree from Inner Mongolia normal university computer science education. She is an associate professor in baotou light industry professional technology institute college of electronic commerce. Her research interests include computer network or database.



Yu li, He received his B.S degree from Inner Mongolia normal university computer science education. He is a lecturer in Baotou city bureau of education test center. His research interests include Computer application.