Mining Negative Sequential Patterns

NANCY P. LIN, HUNG-JEN CHEN, WEI-HUA HAO
Department of Computer Science and Information Engineering
Tamkang University,
151 Ying-Chuan Road, Tamsui, Taipei,
TAIWAN

Abstract: - Sequential pattern mining is to discover all frequent sequences from a sequence database and has been an important issue in data mining. A lot of methods have been proposed for mining sequential pattern. However, conventional methods consider only the occurrences of itemsets in a sequence database, and the sequential patterns are referred to as positive sequential patterns. In practice, the absence of a frequent itemset in a sequence may imply significant information. In this paper, we introduce negative sequential pattern concept in which the absence of an itemset in a sequence is also considered. The major difficulties of negative sequential pattern mining are that there may be huge amounts of the candidates of negative sequences and most of them are meaningless. We proposed an algorithm for mining negative sequential patterns (NSPM). Using NSPM, we prune a number of redundant candidates by applying apriori-principle, and extract meaningful negative sequences from a large number of frequent negative sequences using the interestingness measure.

Key-Words: - Data mining, Negative sequential pattern, Large sequence

1 Introduction

Sequential pattern mining is to discover all frequent subsequences from a given sequences database and has become an important data mining problem with many divers applications, such as basket analysis, web access patterns and quality control in manufactory engineering, etc. For example, users' web pages access sequential patterns can be used to improve a company's website structure to provide more convenient access to the most popular links. The sequential pattern can be divided into Sequential Procurement [1], [2] and Cyclic Procurement [3], [4], [5], [6], [7], [8] by the sequence and the section of time.

A number of methods have been proposed to discover sequential patterns. All the conventional methods for sequential pattern mining were developed to discover positive sequential patterns from database up to now. [1], [8], [9], [10], [11], [12]. Such positive sequential patterns consider only the occurrences of itemsets in a sequence. However, in practice, the absence of an itemset in a sequence may imply valuable information. For example, web pages *A*, *B*, *C*, and *D* are accessed frequently by users, but *D* is seldom accessed after A, B and C. The web page access sequence can be denoted as $< A, B, C \neg D >$, and called a negative sequence. This sequence could give us some valuable information. For example, the link between *C* and *D* is need for improving web pages service. However, it is a difficult problem to find such negative patterns because there may be a huge number of candidates generated and most of them are meaningless.

In this paper, we proposed a method for mining negative sequential patterns, which can avoid a number of redundant candidates and extract meaningful frequent negative sequences from a large number of frequent negative sequences.

2 Problem Statement

A sequence is an ordered list of itemsets. A positive sequence is denoted by $< s_1, s_2,..., s_n >$ and a negative sequence is denoted by $< s_1, s_2, ..., \neg s_n >$, where $\neg s_n$ represents the absence of itemset s_n. The length of a sequence is the number of itemsets in the sequence. A sequence with length l is called an l-sequence. We may note that a sequence $< s_1, s_2, ..., s_n >$ (or a negative sequence $< s_1, s_2, ..., \neg s_n >$ can also be written as $< <s_1, s_2, ...s_{n-1}>,< s_n >>$ (or $<< s_1, s_2, ...s_{n-1}.>,< \neg s_n >>$. That is a sequence can be regarded as an $(n-1)$-sequence $<s_1, s_2, ...s_{n-1}>$, denoted by s_{pre} and called a preceding subsequence, followed by a 1-sequence $<s_{n-1}>$(or $<s_{n-1}>$), denoted by s_{tar} and called a target subsequence. A sequence database *D* is a set of tuples (*cid*, *s*) with primary key *cid* that is a customer-id, and *s* that is a customer transaction sequence.

A positive sequence $< a_1, a_2, ..., a_n >$ is contained in a sequence $< s_1, s_2, ..., s_m >$ if there exist integers *1*

$\leqq i_1 < i_2 \ldots < i_n \leqq m$ such that $a_1 \subseteq s_{i_1}$, $a_2 \subseteq s_{i_2}$, …, $a_n \subseteq s_{i_n}$. A negative sequence $b = < b_1, b_2, ..., \neg b_n >$ is contained in a negative sequence $s = < s_1, s_2, ..., \neg s_m >$, if its positive counterpart $< b_1, b_2, ..., b_n >$ is not contained in *s* and the subsequence, $< b_1, b_2, ..., b_{n-1} >$, of *b* is contained in *s*.

The support of a sequence *s*, Supp(*s*), is α%, if α % of customer sequences in D contain s. A positive sequence *a* is called as sequential pattern (or large positive sequence) in *D* if Supp $(a) \geq \lambda_{ps}$, where λ_{ps} is the user-predefined threshold of the support of positive sequences. With the user-predefined threshold of the support of negative sequences, λ_{ns}, a negative sequence $b = < b_1, b_2, ..., \neg b_n >$ is called a negative sequential pattern (or large negative sequence) in *D* if Supp$(b) \geq \lambda_{ns}$ and the counterpart of the last itemset, b_n is a large *1*-sequence. Note that the condition that b_n being a large 1-sequence is a must, which removes the trivial situation where sequences with itemset b_n occur infrequently.

3 Negative Sequential Patterns

Two major difficulties of mining negative sequential pattern problem are: there may be huge amounts of candidates of negative sequence and most of these candidates are meaningless. To overcome the first problem, we design two generation function, *p_gen*() and *n_gen*(), which can generate a few numbers of the candidates of negative sequence. And we adopt the measure of interestingness to solve the second problem. Two generation function, *p_gen*() and *n_gen*(), and the measure of interestingness are introduced as in the following subsections.

3.1 Candidates Generation

The generation function of the candidates of positive sequences, *p_gen*(), includes two phases: the first for generating new candidates and the second for pruning redundant candidates [1]. In the first phase, the candidates of *k*–sequences are generated from the set of large positive (*k-1*)-sequences join with itself. For example, we can generate two candidates $< s_1, s_2,...,s_{n-2},a_{n-1}, b_{n-1} >$ and $< s_1, s_2,...,s_{n-}, b_{n-1}, a_{n-1} >$ from $< s_1, s_2, ..., s_{n-2}, a_{n-1} >$ and $< s_1,s_2 ...,s_{n-2},b_{n-1} >$ using p_gen function. In the second phase, a candidate of positive *k*-sequence will be deleted if any (*k-1*)-subsequence of it is not a large positive sequence. This is because the apriori-principle states the fact that *any super-pattern of an infrequent pattern cannot be frequent*.

The generation function of the candidates of negative sequences, *n_gen*(), includes two phases: the first for generating new candidates and the second for pruning redundant candidates. In the first phase, the candidates of *k*–sequences are generated from the set of large positive (*k-1*)-sequences join with the set of large negative (*k-1*)-sequences. Note that the way to combine two sequences is slightly different from *p_gen*(). For example, we combine $< a_1, s_2, ..., s_{n-1}, >$ and $<, s_1, ..., s_{n-2}, \neg b_{n-1} >$ to generate $< a_1, s_2, ..., s_{n-1}, \neg b_{n-1} >$. In the second phase, a candidate of negative *k*-sequence will be deleted if any (*k-1*)-subsequence of it is not a large negative sequence.

Function: $n\_gen(LP_{k-1}, LN_{k-1})$
Parameters:
LP_{k-1}: Large positive sequences of length $k-1$
LN_{k-1}: Large negative sequences of length $k-1$
Output:
CN_k: // **Negative sequence Candidates**
Method:
(1)// **Generating new candidates**
(2) **for each** sequence $p =< p_1, p_2,..., p_{k-2}, p_{k-1} >$ in LP_{k-1} **do**
(3) **for each** sequence $q =< q_1, q_2,..., q_{k-2}, \neg q_{k-1} >$ in LN_{k-1} **do**
(4) **if** $((p_{j+1} = q_j),\ for\ all\ j = 1...k-2)$ **then**
(5) **begin**
(6) $new =< p_1, p_2,..., p_{k-1}, \neg q_{k-1} >$
(7) $CN_k = CN_k \cup \{new\}$
(8) **end**
(9)// **Pruning redundant candidates**
(10) $CN_k = CN_k -$ $\{ c \mid c \in CN_k$ and any (k-1)-subsequence of $c \notin LN_{k-1} \}$
(11) **return** CN_k;

Fig. 1. Function *n_gen*()

3.2 Measure of Interestingness

There may be a huge number of sequences generated during mining process, and most of them are not interesting. Therefore, defining a function to measure the degree of interestingness of a sequence is needed. Suppose that $s =< s_1 ... s_n >$ (or $< s_1 ... \neg s_n >$), then

Algorithm: NSPM
Input:
TD : Transaction database
λ_{ps} : Threshold of positive sequences
λ_{ns} : Threshold of negative sequences
λ_{ni} : Threshold of interestingness of negative sequences
Output:
N : Negative sequential patterns
Method:
(1) $LP_1 = \{<i>) \mid i \in I,\ Supp(i) \geq \lambda_{ps}\}$
(2) $LN_1 = \{<\neg i>) \mid i \in LP_1\}$
(3) $N = \phi$
(4) **for** ($k = 2$; $P_{k-1} \neq \phi$; $k++$) **do**
(5) **begin**
(6) **// Mining positive sequential patterns**
(7) $CP_k = p\_gen(LP_{k-1})$
(8) $LP_k = \{<i>) \mid i \in I,\ Supp(i) \geq \lambda_{ps}\}$
(9) **// Mining negative sequential patterns**
(10) $CN_k = n\_gen(LP_{k-1}, LN_{k-1})$
(11) $LN_k = \{<c>) \mid c \in CN_k,\ Supp(c) \geq \lambda_{ns}\}$
(12) $IN_k = \{<l>) \mid l \in LN_k, Im(l) \geq \lambda_{ni}\}$
(13) $N = N \cup IN_k$
(14) **end**
(15) **return** N ;

we have the preceding subsequence $< s_1 ... s_{n-1} >$, s_{pre} , and the target subsequence $< s_{n-1} >$ (or $< \neg s_{n-1} >$), s_{tar} .

We define a measure of interestingness as in the following equation:

$Im(s) = Supp(s) / Supp(s_{pre}) - Supp(s_{tar})$ (1)

If the value of $im(s)$ is greater than or equal to a user-predefine threshold, we can predict that s_{tar} follows s_{pre} with a relatively high probability. In our method, we use $Im()$ to measure the degree of interestingness of a sequence and extract meaningful sequences.

3.3 Algorithm NSPM

In this algorithm, each iteration k consists of two phases: the positive sequential patterns mining phase and the negative sequential patterns mining phase. In the positive sequential patterns mining phase (line 6 - 7), the positive candidates of length k , CP_k , are generated from LP_{k-1} join with LP_{k-1} by p_gen function described in 3.1.Then, support of these candidates is counted by scanning the database D to select large k -sequences, LP_k . In the negative sequential patterns mining phase (line 10 - 13), the negative candidate sequences of length k , CN_k , are generated from LP_{k-1} join with LN_{k-1} by n_gen function described in 3.1, Next, support of these candidates is counted to determine large k -sequences LN_k . Then, the value of the interestingness measure function im of these large sequences is computed for finding negative sequential patterns IN_k that we are interested in. Finally, IN_k is added into N which contains all negative patterns that have already been mined so far.

3.4 Example

Suppose we are given a customer sequence database shown in as Table 1. The threshold of the support of a positive sequence, λ_{ps} , the threshold of the support of a negative sequence, λ_{ns} and the threshold of interestingness of a negative, λ_{ni} are set to 0.4, 0.6 and 0.8, respectively. The processes of the algorithm are shown as in table 2 to table 7. The discovered negative sequential patterns are shown as in table. 8.

CID	Sequence
C01	<(1),(2,3,6),(4)>
C02	<(2,3,6)>
C03	<(1),(3,4,7)>
C04	<(2)>
C05	<(1),(2,3,6),(4,5,8)>

Table 1. Sequence database

In table 2, all candidates of positive 1-sequences (CP_1), their support (*Supp*), large positive *1*-sequences (LP_1) obtained from CP_1. , and large negative *1*-sequences (LN_1) obtained from LP_1 are listed.

CP_1	*Supp*	LP_1	LN_1
<1>	0.6	<1>	<¬1>
<2>	0.8	<2>	<¬2>
<3>	0.8	<3>	<¬3>
<4>	0.6	<4>	<¬4>
<5>	0.2	-	-
<6>	0.6	<6>	<¬6>
<7>	0.2	-	-
<8>	0.2	-	-

Table 2. Positive and negative 1-sequences

In table 3, all candidates of positive 2-sequences (CP_2) and large positive 2-sequences (LP_2) obtained from CP_2. are listed.

CP_2	*Supp*	LP_2	CP_2	*Supp*	LP_2
<1,2>	0.4	<1,2>	<3,4>	0.4	<3,4>
<1,3>	0.6	<1,3>	<3,6>	0	-
<1,4>	0.6	<1,4>	<4,1>	0	-
<1,6>	0.4	<1,6>	<4,2>	0	-
<2,1>	0	-	<4,3>	0	-
<2,3>	0	-	<4,6>	0	-
<2,4>	0.4	<2,4>	<6,1>	0	-
<2,6>	0	-	<6,2>	0	-
<3,1>	0	-	<6,3>	0	-
<3,2>	0	-	<6,4>	0.4	<6,4>

Table 3. Positive 2-sequences

Now, we consider negative sequences, in table 4, all candidates of negative 2-sequences (CN_2) are generated from the joint of LP_1 and LN_1. After the comparisons of support (*Supp*) and measure of interestingness (*Im*) with λ_{ns} and λ_{ni} , large negative 2-sequences (LN_2) obtained from CN_2., and interested negative 2-sequences (IN_2) are obtained and listed.

CN_2	*Supp*	*Im*	LN_2	IN_2
<1,¬2>	0.2	0.13	-	-
<1,¬3>	0	-0.2	-	-
<1,¬4>	0	-0.4	-	-
<1,¬6>	0.2	-0.07	-	-
<2,¬1>	0.8	0.6	<2,¬1>	-
<2,¬3>	0.8	0.8	<2,¬3>	<2,¬3>
<2,¬4>	0.4	0.1	-	-
<2,¬6>	0.8	0.6	<2,¬6>	-
<3,¬1>	0.8	0.6	<3,¬1>	-
<3,¬2>	0.8	0.8	<3,¬2>	<3,¬2>
<3,¬4>	0.4	0.1	-	-
<3,¬6>	0.8	0.6	<3,¬6>	-
<4,¬1>	0.6	0.6	<4,¬1>	-
<4,¬2>	0.6	0.8	<4,¬2>	<4,¬2>
<4,¬3>	0.6	0.8	<4,¬3>	<4,¬3>
<4,¬6>	0.6	0.6	<4,¬6>	-
<6,¬1>	0.6	0.6	<6,¬1>	-
<6,¬2>	0.6	0.8	<6,¬2>	<6,¬2>
<6,¬3>	0.6	0.8	<6,¬3>	<6,¬3>
<6,¬4>	0.2	-0.07	-	-

Table 4. Negative 2-sequences

In table 5, all candidates of positive 3-sequences (CP_3) and large positive 3-sequences (LP_3) obtained from CP_3. are listed.

CP_3	*Supp*	LP_3
<1,2,4>	0.4	<1,2,4>
<1,3,4>	0.4	<1,3,4>
<1,6,4>	0.4	<1,6,4>

Table 5. Positive 3-sequences

In table 6, all candidates of negative 3-sequences (CN_3) generated from the joint of LP_2 and LN_2, support (*Supp*), measure of interestingness (*Im*), large negative 3-sequences (LN_3) obtained from CN_3. , and interested negative 3-sequences (IN_3) are listed.

CN_3	*Supp*	*Im*	LN_3	IN_3
<1,2,¬3>	0.4	0.8	-	-
<1,2,¬4>	0	-0.4	-	-
<1,2,¬6>	0.4	0.6	-	-
<1,3,¬2>	0.6	0.8	<1,3,¬2>	<1,3,¬2>
<1,3,¬4>	0.2	-0.07	-	-
<1,3,¬6>	0.6	0.6	<1,3,¬6>	-
<1,4,¬2>	0.6	0.8	<1,4,¬2>	<1,4,¬2>
<1,4,¬3>	0.6	0.8	<1,4,¬3>	<1,4,¬3>
<1,4,¬6>	0.6	0.6	<1,4,¬6>	-
<1,6,¬2>	0.4	0.8	-	-
<1,6,¬3>	0.4	0.8	-	-
<1,6,¬4>	0	-0.4	-	-
<2,4,¬1>	0.4	0.6	-	-
<2,4,¬3>	0.4	0.8	-	-
<2,4,¬6>	0.4	0.6	-	-
<3,4,¬1>	0.4	0.6	-	-
<3,4,¬2>	0.4	0.8	-	-
<3,4,¬6>	0.4	0.6	-	-
<6,4,¬1>	0.4	0.6	-	-
<6,4,¬2>	0.4	0.8	-	-
<6,4,¬3>	0.4	0.8	-	-

Table 6. Negative 3-sequences

In table 7, all candidates of negative 4-sequences (CN_4) generated from the joint of LP_3 and LN_3, After the comparisons of support (*Supp*) and measure of interestingness (*Im*) with λ_{ns} and λ_{ni} , no more sequences are satisfied, therefore we stop here.

CN_4	*Supp*	*Im*	LN_4	IN_4
<1,2,4,¬3>	0.4	0.8	-	-
<1,2,4,¬6>	0.4	0.6	-	-
<1,3,4,¬2>	0.4	0.8	-	-
<1,3,4,¬6>	0.4	0.6	-	-
<1,6,4,¬2>	0.4	0.8	-	-
<1,6,4,¬3>	0.4	0.8	-	-

Table 7. Negative 4-sequences

Finally, in table 8, all negative sequential patterns discovered are listed.

2-sequences	3-sequences
<2,¬3>	<1,3,¬2>
<3,¬2>	<1,4,¬2>
<4,¬2>	<1,4,¬3>
<4,¬3>	
<6,¬2>	
<6,¬3>	

Table 8. The discovered negative sequential patterns

4 Conclusion

We introduced negative sequential pattern mining concept in which the absence of itemsets in a sequence are also considered. The major difficulties of negative sequential pattern mining are that there may be huge amounts of negative sequence candidates and most of them are meaningless. In the proposed algorithm NSPM, we reduce a number of redundant candidates by applying the apriori-principle and therefore the computational time is reduced. Additionally, we extract meaningful sequential patterns that we are interested in by using the interestingness measure.

References:

[1] R. Agrawal and R. Srikant, Mining Sequential Patterns, *Proceedings of the Elventh International Conference on Data Engineering*, Taipei, Taiwan, March, 1995, pp. 3-14.

[2] R. Srikant and R. Agrawal, Mining Sequential Patterns: Generalizations and Performance Improvements, *Proceedings of the Fifth International conference, Extending Database Technology (EDBT'96)*, 1996, pp. 3-17.

[3] J. Han, G. Dong, Y. Yin, Efficient Mining of Partial Periodic Patterns in Time Series Database, *Proceedings of Fifth International Conference on Data Engineering*, Sydney, Australia, IEEE Computer Society, 1999, pp.106-115.

[4] F. Masseglia, F. Cathala, P. Ponelet, The PSP Approach for Mining Sequential Patterns, *Proceeding of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, Vol. 1510, 1998, pp. 176-184.

[5] J. S. Park, M. S. Chen, P. S. Yu, An Effective Hash Based Algorithm for Mining association rule, *Proceeding of the ACM SIGMOD Conference on management of data*, 1995, pp. 175-186.

[6] J. Pei, B. Motazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M-C. Hsu, Prefixsapn Mining Sequential Patterns Efficiently by Prefix Prejected Pattern Growth, *Proceeding of the International Conference of Data Engineering*, 2001, pp. 215-224.

[7] R. Srikant, R. Agrwal, Mining Assoiation Rules with Item Constraints, *Proceedings of the Third International Conference on Knowledge Discovery in Database and Data Mining*, 1997.

[8] M. J. Zaki, Efficient Enumeration of Frequent Sequences, *Proceedings of the Seventh CIKM,* 1998.

[9] J. Ayres, J. E. Gehrke, T. Yiu, and J. Flannick, Sequential Pattern Mining Using Bitmaps, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada, July 2002.

[10] X. Yan, J. Han, and R. Afshar, CloSpan: Mining Closed Sequential Patterns in Large Datasets, *Proceedings of 2003 SIAM International Conference Data Mining (SDM'03),* 2003, pp. 166-177.

[11] M. Zaki, SPADE: An Efficient Algorithm for Mining Frequent sequences, *Machine Learning,* vol. 40, 2001, pp. 31-60.

[12] M. Zaki, Efficient Enumeration of Frequent Sequences, *Proceedings of the Seventh International Conference Information and Knowledge Management (CIKM'98),* 1998, pp. 68-75.