Mining Strong Positive and Negative Sequential Patterns

NANCY P. LIN, HUNG-JEN CHEN, WEI-HUA HAO,
HAO-EN CHUEH, CHUNG-I CHANG
Department of Computer Science and Information Engineering
Tamkang University,
151 Ying-Chuan Road, Tamsui, Taipei,
TAIWAN
nancylin@mail.tku.edu.tw, chenhj@mail.sju.edu.tw, 889190111@s89.tku.edu.tw,
890190134@s90.tku.edu.tw, taftdc@mail.tku.edu.tw

Abstract: - In data mining field, sequential pattern mining can be applied in divers applications such as basket analysis, web access patterns analysis, and quality control in manufactory engineering, etc. Many methods have been proposed for mining sequential patterns. However, conventional methods only consider the occurrences of itemsets in customer sequences. The sequential patterns discovered by these methods are called as positive sequential patterns, i.e., such sequential patterns only represent the occurrences of itemsets. In practice, the absence of a frequent itemset in a sequence may imply significant information. We call a sequential pattern as negative sequential pattern, which also represents the absence of itemsets in a sequence. The two major difficulties in mining sequential patterns, especially negative ones, are that there may be huge number of candidates generated, and most of them are meaningless. In this paper, we proposed a method for mining strong positive and negative sequential patterns, called PNSPM. In our method, the absences of itemsets are also considered. Besides, only sequences with high degree of interestingness will be selected as strong sequential patterns. An example was taken to illustrate the process of PNSPM. The result showed that PNSPM could prune a lot of redundant candidates, and could extract meaningful sequential patterns from a large number of frequent sequences.

Key-Words: - Data mining, Itemset, Frequent sequence, Positive sequential pattern, Negative sequential pattern, Strong sequential pattern

1 Introduction

Sequential pattern mining is to discover all frequent subsequences from a given sequence database, and it can be applied in 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 in order to provide more convenient access to the most popular links. Thus, sequential pattern mining has become an important task in data mining field. Sequential patterns 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. Most of conventional methods for sequential pattern mining were developed to discover positive sequential patterns from database [1, 8, 9, 10, 11, 12]. Positive sequential patterns mining consider only the occurrences of itemsets in sequences. In practice, however, the absences of itemsets in sequences may imply valuable information. For example, web pages *A*, *B*, *C*, and *D* are accessed frequently by users, but *D* is seldom accessed after the sequence *A*, *B* and *C*. The web page access sequence can be denoted as $< A, B, C \neg D >$, and called a negative sequence. Such sequence could give us some valuable information to improve the company's website structure. For example, a new link between *C* and *D* could improve users' convenience to access web page *D* from *C*.

However, it is a very difficult task to find such sequential 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 strong positive and negative sequential patterns PNSPM. In our method, absences of itemsets in sequences are also considered. Besides, only the sequences with high degree of interestingness will be selected as strong sequential patterns.

2 Problem Statement

A sequence is an ordered list of itemsets. A positive sequence is denoted by $< s_1, s_2, \ldots, s_n >$, 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 Mining Strong Sequential Patterns

Two major difficulties in mining sequential pattern, especially negative ones, are that there may be huge number of candidates of sequences generated, and most of these candidates are meaningless. To overcome the first difficulty, in our method, two functions, *p_gen*() and *n_gen*(), are used to generate positive candidates and negative ones, respectively. They are described in subsections 3.1. For dealing with the second difficulty, the measure of interestingness of sequences, *im()*, is proposed. If a sequence whose value of *im*() is greater than or equal to a user-predefined threshold, we regard it as a interesting sequence. The measure, *im()*, is described in subsections 3.2.

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

Fig. 1. Function *n_gen*()

3.1 Candidates Generation

The function, *p_gen*(), for generating candidates of positive sequences 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, two candidates, $< s_1, s_2, ..., s_{n-2}, a_{n-1}, b_{n-1} >$ and $< s_1, s_2, ..., s_{n-}, b_{n-1}, a_{n-1} >$, are generated by combining two positive sequence, $< s_1, s_2, ..., s_{n-2}, a_{n-1} >$ and $< s_1, s_2 ..., s_{n-2}, b_{n-1} >$. In the second phase, the candidates of positive k-sequences that contain any infrequent (k-*1*)-subsequence will be deleted. This is because the apriori-principle states the fact that *any super-pattern of an infrequent pattern cannot be frequent.*

The function, *n_gen*(), for generating candidates of negative sequences is shown in fig. 1. It 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, in *n_gen*(), the way to combine two sequences to generate a candidate of negative sequence is slightly different from *p_gen*(). For example, the candidate of negative sequence, $< a_1, s_2, ..., s_{n-1}, \neg b_{n-1} >$, is generated by combining the positive sequence $< a_1, s_2, ..., s_{n-1}, >$ and the negative sequence $< , s_1, ..., s_{n-2}, \neg b_{n-1} >$. In the second phase, candidates of negative *k*-sequences that contain any infrequent (*k-1*)-subsequence will be deleted.

3.2 Measure of Interestingness

There may be a huge number of sequences generated during sequential pattern mining, and most of them are uninteresting. Therefore, defining a function to measure the degree of interestingness of a sequence is needed.

Suppose that $s =< s_1...s_n >$ (or $< s_1...\neg s_n >$), the preceding subsequence, s_{pre}, is $< s_1...s_{n-1} >$, and the target subsequence, s_{tar}, is $< s_n >$ (or $< \neg s_n >$). We define the measure of interestingness as following equation:

$$im(s) = supp(s) / supp(s_{pre}) - supp(s_{tar}) \quad (1)$$

If the value of $im(s)$ is large enough, we could say that the probability of occurrence of s_{tar} after s_{pre} is higher than the probability of s_{tar} occurrence in average case. And this sequence, *s*, is worth to pay attention to, i.e., it is an interesting sequence. We call *s* as a strong positive sequential pattern (or a strong negative sequential pattern) if s is a positive sequential pattern (or a negative sequential pattern), and the value of $im(s)$ is greater than or equal to a user-predefined threshold.

3.3 Algorithm PNSPM

The algorithm PNSPM is an iterative procedure as shown in fig. 2. In the algorithm, the iteration contains two phases: the phase of positive sequential pattern mining (line 6-9), and the phase of negative sequential pattern mining (line 10-13).

In the positive sequential pattern mining phase, the candidates of positive sequences with length k, CP_k, are generated from LP_{k-1} join with LP_{k-1} by p_gen function (line 6). Next, large k -sequences, LP_k, are selected if their supports are greater than or equal to a user-predefined threshold (line 7). Then, the strong positive sequential patterns IP_k are selected if their values of *im* are greater than or equal to a user-predefined threshold (line 8). Finally, IP_k are added into P, which contains all strong positive patterns have already been mined so far (line 9).

In the negative sequential pattern mining phase, the candidates of negative sequences with length k, CN_k, are generated from LP_{k-1} join with LN_{k-1} by n_gen function (line 10), Next, large sequences LN_k are selected if their supports are greater than or equal to a user-predefined threshold (line 11). Then, strong negative sequential patterns IN_k, are selected if their values of *im* are greater than or equal to a user-predefined threshold (line 12). Finally, IN_k are added into N, which contains all strong negative patterns have already been mined so far.

Algorithm: PNSPM

Input:

TD : Transaction database

λ_{ps} : Threshold of support of positive sequences

λ_{ns} : Threshold of support of negative sequences

λ_{pi} : Threshold of interestingness of positive sequences

λ_{ni} : Threshold of interestingness of negative sequences

Output:

P : Strong positive sequential patterns

N : Strong negative sequential patterns

Method:

(1) $P = IP_1 = LP_1 = \{< i >) \mid i \in I, supp(i) \geq \lambda_{ps}\}$

(2) $LN_1 = \{< \neg i >) \mid i \in LP_1\}$

(3) $N = \phi$

(4) **for** $(k = 2; LP_{k-1} \neq \phi; k++)$ **do**

(5) **begin**

// **Mining Positive sequential patterns**

(6) $CP_k = p\_gen(LP_{k-1})$

(7) $LP_k = \{< c >) \mid c \in CP_k, supp(c) \geq \lambda_{ps}\}$

(8) $IP_k = \{< l >) \mid l \in LP_k, im(l) \geq \lambda_{pi}\}$

(9) $P = P \cup IP_k$

// **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** P, N;

Fig. 2. Algorithm PNSPM

3.4 Example

Suppose a customer sequence database is given as

shown in Table 1. The threshold of the supports of positive sequences, λ_{ps} , the threshold of the interestingness of positive sequences, λ_{pi} , the threshold of the supports of negative sequences, λ_{ns} , and the threshold of interestingness of negative sequences, λ_{ni} are set to 0.4, 0.2, 0.6, and 0.8, respectively. The process of the algorithm is shown in table 2 to table 7. The discovered strong positive and negative sequential patterns are shown in table 8.

Table 1. Sequence database

CID	Sequence
1	<(a),(c,d,g)>
2	<(b)>
3	<(a),(b,c,f),(d)>
4	<(a),(b,c,f),(d,e,h)>
5	<(b,c,f)>

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.

Table 2. Positive and negative 1-sequences

CP_1	*supp*	LP_1	LN_1
<a>	0.6	<a>	<¬a>
<b>	0.8	<b>	<¬b>
<c>	0.8	<c>	<¬c>
<d>	0.6	<d>	<¬d>
<e>	0.2	-	-
<f>	0.6	<f>	<¬f>
<g>	0.2	-	-
<h>	0.2	-	-

In table 3, all candidates of positive 2-sequences (CP_2) are generated from the joint of LP_1 and LP_1. After the comparisons of support (*supp*) and measure of interestingness (*im*) with λ_{ps} and λ_{pi} , large positive 2-sequences (LP_2) are obtained from CP_2, and strong positive 2-sequences (IP_2) are obtained from LP_2 , respectively.

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) are obtained from CN_2, and strong negative 2-sequences (IN_2) are obtained from LN_2 , respectively.

Table 3. Positive 2-sequences

CP_2	*supp*	LP_2	*im*	IP_2
<a,b>	0.4	<a,b>	-0.13	-
<a,c>	0.6	<a,c>	0.2	<a,c>
<a,d>	0.6	<a,d>	0.4	<a,d>
<a,f>	0.4	<a,f>	0.07	-
<b,a>	0	-	-	-
<b,c>	0	-	-	-
<b,d>	0.4	<b,d>	-0.1	-
<b,f>	0	-	-	-
<c,a>	0	-	-	-
<c,b>	0	-	-	-
<c,d>	0.4	<c,d>	-0.1	-
<c,f>	0	-	-	-
<d,a>	0	-	-	-
<d,b>	0	-	-	-
<d,c>	0	-	-	-
<d,f>	0	-	-	-
<f,a>	0	-	-	-
<f,b>	0	-	-	-
<f,c>	0	-	-	-
<f,d>	0.4	<f,d>	0.07	-

Table 4. Negative 2-sequences

CN_2	*supp*	LN_2	*im*	IN_2
<a,¬b>	0.2	-	-	-
<a,¬c>	0	-	-	-
<a,¬d>	0	-	-	-
<a,¬f>	0.2	-	-	-
<b,¬a>	0.8	<b,¬a>	0.6	-
<b,¬c>	0.8	<b,¬c>	0.8	<b,¬c>
<b,¬d>	0.4	-	-	-
<b,¬f>	0.8	<b,¬f>	0.6	-
<c,¬a>	0.8	<c,¬a>	0.6	-
<c,¬b>	0.8	<c,¬b>	0.8	<c,¬b>
<c,¬d>	0.4	-	-	-
<c,¬f>	0.8	<c,¬f>	0.6	-
<d,¬a>	0.6	<d,¬a>	0.6	-
<d,¬b>	0.6	<d,¬b>	0.8	<d,¬b>
<d,¬c>	0.6	<d,¬c>	0.8	<d,¬c>
<d,¬f>	0.6	<d,¬f>	0.6	-
<f,¬a>	0.6	<f,¬a>	0.6	-
<f,¬b>	0.6	<f,¬b>	0.8	<f,¬b>
<f,¬c>	0.6	<f,¬c>	0.8	<f,¬c>
<f,¬d>	0.2	-	-	-

In table 5, all candidates of positive 3-sequences (CP_3), and large positive 3-sequences (LP_3) obtained from CP_3 are listed. Note that no strong positive 3-sequence (IP_3) are obtained from LP_3 since there is no large positive 3-sequences (LP_3), whose *im* is greater than or equal to λ_{pi}. Moreover, no candidates of positive 4-sequences (CP_4) can be generated from LP_3, therefore, the phase of mining positive sequential patterns is stopped here.

Table 5. Positive 3-sequences

CP_3	*supp*	LP_3	*im*	IP_3
<a,b,d>	0.4	<a,b,d>	0.07	-
<a,c,d>	0.4	<a,c,d>	0.07	-
<a,f,d>	0.4	<a,f,d>	0.07	-

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 strong negative 3-sequences (IN_3) obtained from LN_3 are listed.

Table 6. Negative 3-sequences

CN_3	*supp*	LN_3	*im*	IN_3
<a,b,¬c>	0.4	-	-	-
<a,b,¬d>	0	-	-	-
<a,b,¬f>	0.4	-	-	-
<a,c,¬b>	0.6	<a,c,¬b>	0.8	<a,c,¬b>
<a,c,¬d>	0.2	-	-	-
<a,c,¬f>	0.6	<a,c,¬f>	0.6	-
<a,d,¬b>	0.6	<a,d,¬b>	0.8	<a,d,¬b>
<a,d,¬c>	0.6	<a,d,¬c>	0.8	<a,d,¬c>
<a,d,¬f>	0.6	<a,d,¬f>	0.6	-
<a,f,¬b>	0.4	-	-	-
<a,f,¬c>	0.4	-	-	-
<a,f,¬d>	0	-	-	-
<b,d,¬a>	0.4	-	-	-
<b,d,¬c>	0.4	-	-	-
<b,d,¬f>	0.4	-	-	-
<c,d,¬a>	0.4	-	-	-
<c,d,¬b>	0.4	-	-	-
<c,d,¬f>	0.4	-	-	-
<f,d,¬a>	0.4	-	-	-
<f,d,¬b>	0.4	-	-	-
<f,d,¬c>	0.4	-	-	-

In table 7, all candidates of negative 4-sequences (CN_4) are generated from the joint of LP_3 and LN_3. After the comparison of support (*supp*) with λ_{ns}, no large negative 4-sequences (LN_4) are obtained from CN_4, since no more candidates are satisfied. Therefore, the algorithm is stopped here.

Table 7. Negative 4-sequences

CN_4	*supp*	LN_4	*im*	IN_4
<a,b,d,¬c>	0.4	-	-	-
<a,b,d,¬f>	0.4	-	-	-
<a,c,d,¬b>	0.4	-	-	-
<a,c,d,¬f>	0.4	-	-	-
<a,f,d,¬b>	0.4	-	-	-
<a,f,d,¬c>	0.4	-	-	-

Finally, in table 8, all strong positive and negative sequential patterns discovered from customer sequence database, are listed.

Table 8. The discovered strong positive and negative sequential patterns

2-sequences	3-sequences
positive	
<a,c> <a,d>	
negative	
<b,¬c> <c,¬b> <d,¬b> <d,¬c> <f,¬b> <f,¬c>	<a,c,¬b> <a,d,¬b> <a,d,¬c>

4 Conclusion

The two major difficulties in mining sequential patterns, especially negative ones, are that there may be huge number of the candidates generated, and most of them are meaningless. In this paper, we proposed a method, PNSPM, for mining strong positive and negative sequential patterns. In our method, the absences of itemsets in sequences are also considered. Besides, only the sequences with high degree of interestingness will be selected as strong sequential patterns. The result showed that PNSPM could prune a lot of redundant candidates by applying apriori-principle, and could extract meaningful sequential patterns from a large number of frequent sequences.

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.