

Efficient Mining of Event-Oriented Negative Sequential Rules

¹Yanchang Zhao, ¹Huaifeng Zhang, ¹Longbing Cao, ¹Chengqi Zhang, and ²Hans Bohlscheid

¹Data Sciences & Knowledge Discovery Research Lab
Centre for Quantum Computation and Intelligent Systems
Faculty of Engineering & IT, University of Technology, Sydney, Australia
{yczhao,hfzhang,lbcao,chengqi}@it.uts.edu.au

²Projects Section, Business Integrity Programs Branch,
Centrelink, Australia
hans.bohlscheid@centrelink.gov.au

Abstract

Traditional sequential pattern mining deals with positive sequential patterns only, that is, only frequent sequential patterns with the appearance of items are discovered. However, it is often interesting in many applications to find frequent sequential patterns with the non-occurrence of some items, which are referred to as negative sequential patterns. This paper analyzes three types of negative sequential rules and presents a new technique to find event-oriented negative sequential rules. Its effectiveness and efficiency are shown in our experiments.

Keywords: *Negative sequential patterns, negative sequential rules, sequential patterns, sequence mining*

1 Introduction

Association rule mining [1] and sequential pattern mining [2] were respectively proposed in 1993 and 1995, and have been well studied and developed. Traditional association rules and sequential patterns involve only the co-occurrence of itemsets/events, that is, the positive relationship between itemsets/events. However, it is sometimes interesting to find negative correlations, e.g., two items are seldom bought together in a same basket. Therefore, negative association rules are designed to find such kind of relationships [11, 15, 3, 14]. Similarly, it is also interesting to find negative sequential patterns, e.g., one event seldom occurs after another event. Nevertheless, such kind of negative sequential patterns are seldom studied.

Moreover, in many applications, it is interesting to find which sequential patterns will lead to the occurrence/non-

occurrence of some single events. For example, in a web click-stream scenario, an analysis is to find the relationship between visiting sequences and the click of a link. For a credit card company, it is interesting to discover the positive and negative relationship between transaction sequences and an unrecovered debt. In homeland security, the correlation between a series of activities and a terrorism attack is one of the important aims of analysis.

In this paper, we will study the negative relationship in sequence data, which are defined in two forms: 1) *negative sequential patterns (NSP)* like $a\neg(bc)da\rightarrow e$, where a to e are events/items; and 2) *negative sequential rules (NSR)* like $A \rightarrow \neg B$, $\neg A \rightarrow B$ and $\neg A \rightarrow \neg B$, where A and B are positive sequential patterns composed of items in time order. From NSP, sequential rules like $a\neg(bc)d \rightarrow a\rightarrow e$ can be derived, and we name such kind of rules as *generalized negative sequential rules (GNSR)*. There are some preliminary work on negative sequential pattern mining [5, 8, 9, 13], but they only deal with a small part of the problem and are inefficient for mining such patterns.

To tackle the above problem, we will develop the idea of *negative sequential rules* whose left or/and right sides can be negations of traditional positive sequential patterns, and a new efficient algorithm will be designed for mining *event-oriented negative sequential rules*, where the right side of a rule is a single event or its negation. The contributions of this paper are:

- a definition of negative sequential rules and the support, confidence and lift for negative sequential rules; and
- an efficient algorithm for discovering event-oriented negative sequential rules based on SPAM algorithm [4].

Table 1. Notations

Symbol	Description
A, B	Positive sequential patterns
$A \bowtie B$	The conjunction of A and B , where A is followed by B in a sequence.
AB	Same as $A \bowtie B$. They are used interchangeably in this paper.
$A \& B$	The concurrence of A and B in a sequence. Note that there is no time order between A and B , and they may even be interweaved in a sequence.
C_k	The set of candidate frequent sequential patterns with k items
L_k	The set of frequent sequential patterns with k items
M_k	The set of infrequent sequential patterns with k items

The notations used throughout this paper are given in Table 1.

The rest of this paper is organized as follows. Section 2 briefly introduces some related work on negative association rules, sequential patterns and negative sequential patterns. The targeted problem is described in Section 3. Section 4 presents a formal definition of negative sequential rules, interestingness measures and an efficient technique for mining event-oriented negative sequential rules. Section 5 shows our experimental results. Conclusions and future work are given in the last section.

2 Background and Related Work

The technique of negative association rules has been well studied [3, 11, 14, 15]. Negative association rules are defined in the form of $A \rightarrow \neg B$, $\neg A \rightarrow B$ and $\neg A \rightarrow \neg B$ [14, 15]. Savasere et al. designed negative association rules as $A \not\rightarrow B$ [11]. Antonie and Zaiiane defined *generalized negative association rule* as a rule containing a negation of an item, such as $A \wedge \neg B \wedge \neg C \wedge D \rightarrow E \wedge \neg F$ [3], but their work is still focused on finding rules like $A \rightarrow \neg B$, $\neg A \rightarrow B$ and $\neg A \rightarrow \neg B$, which are referred to by them as *confined negative association rules*.

The idea of sequential patterns was proposed in 1995 to find frequent sequential patterns in sequence data, such as transactional data [2]. The difference from frequent itemset mining is that time order is taken into account in sequential pattern mining. Well-known algorithms for sequential pattern mining are AprioriALL [2], GSP (Generalized Sequential Patterns) [12], FreeSpan [7], PrefixSpan [10],

SPADE [16] and SPAM (Sequential Pattern Mining) [4].

For sequential patterns, the non-occurrence of an element may also be interesting. For example, in social welfare, the lack of follow-up examination after the address change of a customer may result in overpayment to the customer. Such kind of sequences with the non-occurrence of elements are negative sequential patterns. However, most research on sequential patterns are focused on positives, and negative sequential patterns are underdeveloped. To the best of our knowledge, reported researches on negative sequential patterns are as follows.

Sun et al. proposed negative event-oriented patterns [13]. A negative rule in the form of $\neg P \xrightarrow{T} e$, where e is a target event, P is a negative event-oriented pattern, and the occurrence of P is unexpected rear in T -sized intervals before target events. It is a special case of negative sequential pattern $\neg A \rightarrow B$. In their work, P is supposed to be an "existence pattern" (i.e., a frequent itemset without time order), instead of a sequential pattern, although they claimed that their discussions can be easily extended to sequential patterns.

Bannai et al. proposed a method for finding optimal pairs of string patterns to discriminate between two sets of strings [5]. The pairs are in the forms of $p' \wedge q'$ and $p' \vee q'$, where p' is either p or $\neg p$, q' is either q or $\neg q$, and p and q are two substrings. Their concern is whether p and q appear in a string s . The substring can be taken as a special case of sequential pattern, where the elements in the patterns are continuous.

Ouyang and Huang proposed negative sequences as $(A, \neg B)$, $(\neg A, B)$ and $(\neg A, \neg B)$ [9]. Their idea is generating frequent itemsets first, based on which both frequent and infrequent sequences are found, and then negative sequential patterns are derived from infrequent sequences. A drawback of their algorithm is that both frequent and infrequent sequences have to be found at the first stage, which demands a large amount of space.

Lin et al. designed an algorithm NSPM (Negative Sequential Patterns Mining) for mining negative sequential patterns [8]. In their designed negative patterns, only the last element can be negative, and all other elements are positive.

Chen et al. designed a technique PNSP (Positive and Negative Sequential Patterns Mining) for mining positive and negative sequential patterns in the form of $(abc)\neg(de)(ijk)$ [6]. They proposed some constraints for negative sequential patterns, e.g., a valid negative sequential patterns do not contain contiguous absence of elements, and an itemset must be frequent to make it a valid negative itemset. Their method is broken into three stages. At the first stage, all positive sequential patterns are found. At the second stage, all positive itemsets are found, from which all negative itemsets are derived. At the third stage, both pos-

itive and negative itemsets are joined to generate candidate negative sequential patterns, which are in turn joined iteratively to generate longer negative sequences in an apriori-like way.

3 Problem Statement

A *negative sequential rule (NSR)* is defined as a rule in the form of $A \rightarrow B$, where either or both sides of the rule is/are the negations of positive sequential patterns. A negative sequential rule with the right side as a single event is referred to as an *event-target negative sequential rule*. The target problem in this paper is as follows: Given a database of sequences and three user-specified parameters, minimum support $minsupp$, minimum confidence $minconf$ and minimum lift $minlift$, find all event-oriented negative sequential rules whose support, confidence and lift are no less than $minsupp$, $minconf$ and $minlift$, respectively.

In this paper, we consider the case where every transaction is composed of one item only. That is, the sequences look like $abcd$, instead of $a\{bc\}ad$, where b and c occur in one transaction in the latter case.

4 Mining Event-Oriented Negative Sequential Rules

In this section, negative sequential rules will be designed and their support, confidence and lift will be derived. Then an efficient algorithm for mining event-oriented negative sequential rules will be presented.

4.1 Negative Sequential Rules

Traditional sequential rules are positive sequential rules, which are in the form of $A \rightarrow B$, where both A and B are positive sequential patterns. It means that pattern A is followed by B . We refer to such positive rules as Type I sequential rules, and their support, confidence and lift are as follows.

$$supp(A \rightarrow B) = P(AB) \quad (1)$$

$$conf(A \rightarrow B) = \frac{P(AB)}{P(A)} \quad (2)$$

$$lift(A \rightarrow B) = \frac{P(AB)}{P(A)P(B)} \quad (3)$$

$$= 1 + \frac{P(AB) - P(A)P(B)}{P(A)P(B)} \quad (4)$$

By changing A or/and B to their negations, we can get the following three types of negative sequential rules:

- Type II: $A \rightarrow \neg B$, which means that pattern A is not followed by B ;
- Type III: $\neg A \rightarrow B$, which means that if A does not appear, then B will occur; and
- Type IV: $\neg A \rightarrow \neg B$, which means that if A does not appear, then B will not occur.

For types III and IV whose left sides are the negations of sequential patterns, the meaning of the rules are: if A does not occur in a sequence, then B will (type III) or will not (type IV) occur in the sequence. That is to say, there is no time order between the left side and the right side. Note that A and B themselves are sequential patterns, which makes them different from negative association rules. However, if time constraint is considered in sequential rules, the two types of rules may have new meanings, which is out of the scope of this paper. The supports, confidences and lifts of the above three types of negative sequential rules are calculated as follows.

$$supp(A \rightarrow \neg B) = P(A\neg B) \quad (5)$$

$$= P(A) - P(AB) \quad (6)$$

$$conf(A \rightarrow \neg B) = \frac{P(A) - P(AB)}{P(A)} \quad (7)$$

$$lift(A \rightarrow \neg B) = \frac{P(A) - P(AB)}{P(A)(1 - P(B))} \quad (8)$$

$$= 1 + \frac{P(A)P(B) - P(AB)}{P(A)(1 - P(B))} \quad (9)$$

$$supp(\neg A \rightarrow B) = P(\neg A \& B) \quad (10)$$

$$= P(B) - P(A \& B) \quad (11)$$

$$conf(\neg A \rightarrow B) = \frac{P(B) - P(A \& B)}{1 - P(A)} \quad (12)$$

$$lift(\neg A \rightarrow B) = \frac{P(B) - P(A \& B)}{P(B)(1 - P(A))} \quad (13)$$

$$= 1 + \frac{P(A)P(B) - P(A \& B)}{P(B)(1 - P(A))} \quad (14)$$

$$supp(\neg A \rightarrow \neg B) = supp(\neg A) - supp(\neg A \rightarrow B) \quad (15)$$

$$= 1 - P(A) - P(B) + P(A \& B) \quad (16)$$

$$conf(\neg A \rightarrow \neg B) = \frac{1 - P(A) - P(B) + P(A \& B)}{1 - P(A)} \quad (17)$$

$$lift(\neg A \rightarrow \neg B) = \frac{1 - P(A) - P(B) + P(A \& B)}{(1 - P(A))(1 - P(B))} \quad (18)$$

$$= 1 + \frac{P(A \& B) - P(A)P(B)}{(1 - P(A))(1 - P(B))} \quad (19)$$

In the above equations, $P(A \& B)$ denotes the probability of the concurrence of A and B in a sequence, no matter which one occurs first, or whether they are interweaved. So we can get $P(A \& B) \geq P(AB)$.

Based on the above equations for lifts, we get the following: 1) If $P(AB) > P(A)P(B)$, then $P(A\&B) > P(A)P(B)$, so $lift(A \rightarrow B) > 1$ (see Eqn. 4) and $lift(\neg A \rightarrow \neg B) > 1$ (see Eqn. 19), which suggests that $A \rightarrow B$ and $\neg A \rightarrow \neg B$ may be interesting. Otherwise, $A \rightarrow \neg B$ may be interesting (see Eqn. 9). 2) If $P(A\&B) > P(A)P(B)$, then $\neg A \rightarrow \neg B$ may be interesting (see Eqn. 19). Otherwise, we have $P(AB) \leq P(A\&B) \leq P(A)P(B)$, so $A \rightarrow \neg B$ (see Eqn. 9) and $\neg A \rightarrow B$ (see Eqn. 14) may be interesting. The above observation coincides with Antonie and Zaïane’s idea to find positive and negative association rules [3]. They proposed to check the correlation between itemsets, and then generate negative association rules like Types II & III when the correlation is negative and generate association rules like Types I & IV when the correlation is positive. However, our observation works for sequential patterns, while theirs works for association rules.

Note that the above four types of sequential rules and the above equations for their supports, confidences and lifts are not limited to event-oriented sequential rules, that is, they can also be applied to generalized sequential rules where the right sides are sequences with two or more items.

4.2 Algorithm for Mining Event-Oriented Negative Sequential Rules

To discover event-oriented negative sequential rules, we use SPAM (Sequential PAttern Mining) [4] as a start point, because it is more efficient than SPADE and Prefixpan [4], two well-known algorithms for sequential pattern mining. SPAM uses a vertical bitmap representation of the database and a depth-first search strategy for efficient mining of sequential patterns. It searches the sequence lattice in a depth-first way, and candidates of longer sequences S_g are generated by append frequent items $\{i\}$ to existing frequent sequences S_a . The algorithm is very efficient in that it uses bitmap to count the frequency of sequences and only the nodes in the path to the current node are kept. The candidate generation of SPAM is composed of two steps: S-step and I-step. The S-step appends i to S_a , which builds a longer sequence $S_g = S_a \bowtie i$. The I-step adds i to the last itemset of S_a , which builds a new sequence of the same length as S_a . Since we consider transaction with one item only, an element in a sequence is a single item, instead of an itemset. Therefore, only S-step from SPAM is used in our technique.

The frequency of items (i.e., L_1 sequence) are counted at first and kept for the computation of sequential rules. Since the algorithm works with a depth-first search strategy, so the support of S_a is available in the parent node. The support of item i is available at the very beginning. Because the bitmaps of S_a and i are available at each node for counting the support of $S_a \bowtie i$ (with a bit-wise AND-operation), the

support of $S_a \& i$ can be easily derived. Therefore, all information is available for computing the support, confidence and lift of the sequential rules.

Figure 1 gives the pseudocode for finding negative sequential rules, which is based on the function FindSequentialPatterns() from SPAM. A recursive call goes down the search lattice to find sequential rules. For each possible extension i at a level, the support is computed with a bit-wise AND of post-processed bitmap with the candidate frequent-1 itemset (see lines 2-6). If the result shows that the support is no less than a given support threshold, then i is added to the extension list of the node at next level and the confidence and lift for positive sequential patterns (Type I) are calculated (see lines 8-14). Otherwise, the support, confidence and lift for negative sequential patterns of types II, III and IV are computed (see lines 16-33). If the support, confidence and lift are all above predefined thresholds, the corresponding rules are outputted. Then the node at next level is checked recursively to find all sequential rules (see line 35). At each level of the sequence lattice, only one node is kept in memory, which makes it very space efficient.

5 Experimental Evaluation

Our designed algorithm (referred to as SpamNeg) was implemented with C++ based on the source code of the SPAM algorithm from <http://himalaya-tools.sourceforge.net/Spam/> and tested with synthetic dataset generated with IBM AssocGen transactional data generator [2]. All the tests were conducted on a PC with Intel Core 2 CPU of 1.86GHz, 2GB memory and Windows XP Professional SP2. The number of items per transaction was set to one when generating datasets.

Our algorithm was first tested on a dataset with 10,000 customers, 50 items per sequence and the length of maximal patterns as 20. By setting the minimum supports ranging from 0.2 to 0.7, the results are shown in Figure 2. From the figure, both SpamNeg and Spam [4] run faster with larger minimum support, because the search space becomes smaller. Moreover, SpamNeg needs longer running time than Spam, and the extra time is caused by generating negative sequential rules when the support of a candidate is less than the support threshold.

The scalability of our algorithm was tested on datasets with average sequence length as 40, length of maximal patterns as 10. The number of customers ranged from 10,000 to 60,000, and the support threshold was set to 0.3. Figure 3 shows the result of the above test. It’s clear from the figure that SpamNeg is linear with the number of sequences.

The running time with varying sequence lengths is shown Figure 4, where the datasets used have 100,000 customers, with length of maximal patterns as 20, and the average sequence length ranging from 10 to 45. The support

ALGORITHM: FindNegativeSequentialRules, a recursive call that goes down the search lattice to find negative sequential rules
INPUT: *curNode*: information about the current node
OUTPUT: event-oriented negative sequential rules

```

1: FOR each possible s-extension i from this level
2:   /* AND the post-processed s-step bitmap with the candidate frequent-1 itemset; */
3:   cntA = curNode.parent.count;
4:   cntB = i.count;
5:   tempAndBitmap=Bit-Wise-And(post-processed bitmap of curNode, bitmap of i);
6:   cntAB=tempAndBitmap.count; /* corresponding to  $P(AB)$  */
7:   IF  $cntAB \geq minsupp * totalCust$  /* totalCust is the number of customers/sequences. */
8:     add i to nextNode's s-extension list;
9:     /* generate positive sequential patterns */
10:    // Type I:  $A \rightarrow B$ 
11:    supp =  $cntAB / totalCust$ ;
12:    conf =  $cntAB / cntA$ ;
13:    lift =  $cntAB * totalCust / (cntA * cntB)$ ;
14:    output " $A \rightarrow B$ " if  $supp \geq minsupp$ ,  $conf \geq minconf$  and  $lift \geq minlift$ ;
15:  ELSE
16:    /* generate negative sequential patterns */
17:    tempOrBitmap=Bit-Wise-Or(bitmap of curNode, bitmap of i);
18:    cntAorB = the count of "1" in tempOrBitmap;
19:    cntAandB =  $cntA + cntB - cntAorB$ ; /* corresponding to  $P(A \& B)$  */
20:    IF  $P(A \& B) < P(A)P(B)$  /*  $P(AB) \leq P(A \& B) < P(A)P(B)$  */
21:      compute support, confidence and lift according to Equations 5, 7 and 8.
22:      output " $A \rightarrow \neg B$ " if  $supp \geq minsupp$ ,  $conf \geq minconf$  and  $lift \geq minlift$ ;
23:      compute support, confidence and lift according to Equations 10, 12 and 13.
24:      output " $\neg A \rightarrow B$ " if  $supp \geq minsupp$ ,  $conf \geq minconf$  and  $lift \geq minlift$ ;
25:    ELSE IF  $P(AB) < P(A)P(B)$  /*  $P(AB) < P(A)P(B) \leq P(A \& B)$  */
26:      compute support, confidence and lift according to Equations 5, 7 and 8.
27:      output " $A \rightarrow \neg B$ " if  $supp \geq minsupp$ ,  $conf \geq minconf$  and  $lift \geq minlift$ ;
28:      compute support, confidence and lift according to Equations 15, 17 and 18.
29:      output " $\neg A \rightarrow \neg B$ " if  $supp \geq minsupp$ ,  $conf \geq minconf$  and  $lift \geq minlift$ ;
30:    ELSE /*  $P(A)P(B) \leq P(AB) \leq P(A \& B)$  */
31:      compute support, confidence and lift according to Equations 15, 17 and 18.
32:      output " $\neg A \rightarrow \neg B$ " if  $supp \geq minsupp$ ,  $conf \geq minconf$  and  $lift \geq minlift$ ;
33:    END IF
34:  END IF
35:  FindNegativeSequentialRules(nextNode); /* A recursive call to check the node at the next level */
36: END FOR

```

Figure 1. Algorithm for Discovering Event-Oriented Negative Sequential Rules

threshold was set to 0.2. The figure shows that the running time becomes longer with the increase of the average number of items per sequence.

The running time of our algorithm was also tested on datasets with average length of maximal patterns ranging from 5 to 15. Each dataset has 100,000 customers, with average sequence length as 30. The support threshold was also set to 0.2. The experimental results in Figure 5 show that the running time decreases slightly with the increase of the average length of maximal patterns.

6 Conclusions

We have formally defined negative sequential rules and have derived the supports, confidences and lifts for negative sequential rules. An efficient algorithm for mining event-oriented negative sequential rules is designed based on the SPAM algorithm. Our algorithm has been tested on numerous synthetic datasets generated with IBM data generator, which shows its efficiency and scalability.

The proposed algorithm can only find negative sequential rules with a single event on the right side. Sometimes it may be interesting to find more generalized negative sequential rules, which will be included in our future work. Moreover, negative sequential rules with time constraints will also be a part of our future research.

Acknowledgments

This work was supported by the Australian Research Council (ARC) Linkage Project LP0775041 and Discovery Projects DP0667060 & DP0773412, and by the Early Career Researcher Grant from University of Technology, Sydney, Australia.

References

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington D.C. USA, May 1993.
- [2] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen, editors, *Proc. of the 11th International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.
- [3] M.-L. Antonie and O. R. Zaïane. Mining positive and negative association rules: an approach for confined

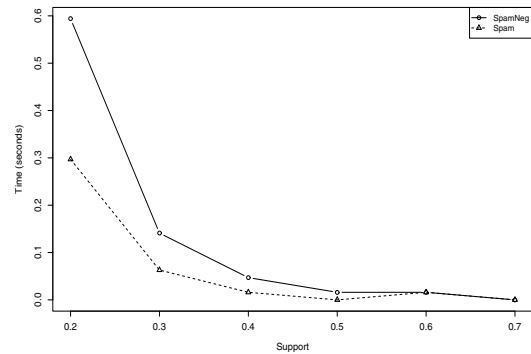


Figure 2. Scalability with Minimum Supports

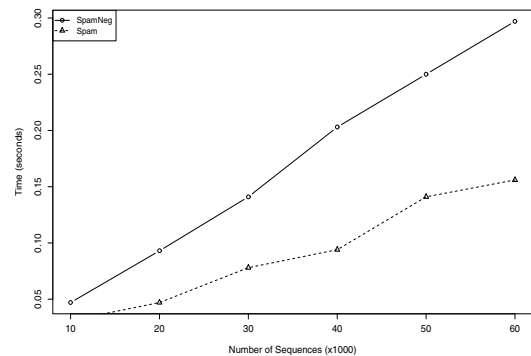


Figure 3. Scalability with the Number of Sequences

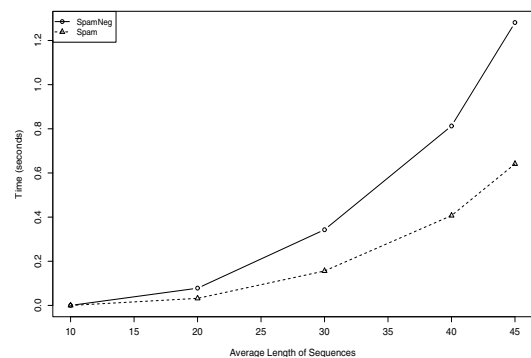


Figure 4. Scalability with the Number of Items per Sequence

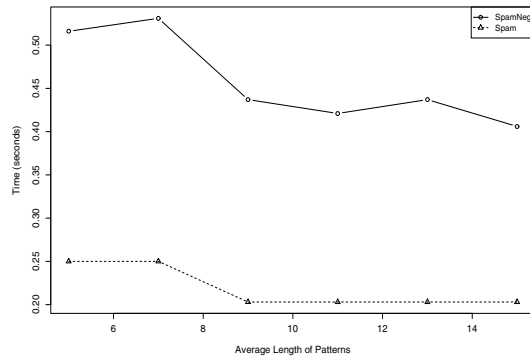


Figure 5. Scalability with the Average Length of Patterns

rules. In *PKDD '04: Proc. of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 27–38, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

- [4] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *KDD'02: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 429–435, New York, NY, USA, 2002. ACM.
- [5] H. Bannai, H. Hyyro, A. Shinohara, M. Takeda, K. Nakai, and S. Miyano. Finding optimal pairs of patterns. In *Proc. of the 4th Workshop on Algorithms in Bioinformatics (WABI'04)*, 2004.
- [6] C.-L. Chen and M.-Y. Lin. Positive and negative sequential pattern mining for practical applications. url: <http://selab.iecs.fcu.edu.tw/course/96/dp2/dpclass/07.ppt>, May 2008.
- [7] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *KDD '00: Proc. of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 355–359, New York, NY, USA, 2000. ACM.
- [8] N. P. Lin, H.-J. Chen, and W.-H. Hao. Mining negative sequential patterns. In *Proc. of the 6th WSEAS International Conference on Applied Computer Science*, pages 654–658, Hangzhou, China, 2007.
- [9] W.-M. Ouyang and Q.-H. Huang. Mining negative sequential patterns in transaction databases. In *Proc. of 2007 International Conference on Machine Learning and Cybernetics*, pages 830–834, Hong Kong, China, 2007.
- [10] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICDE '01: Proc. of the 17th International Conference on Data Engineering*, page 215, Washington, DC, USA, 2001. IEEE Computer Society.
- [11] A. Savasere, E. Omiecinski, and S. B. Navathe. Mining for strong negative associations in a large database of customer transactions. In *ICDE '98: Proc. of the 14th International Conference on Data Engineering*, pages 494–502, Washington, DC, USA, 1998. IEEE Computer Society.
- [12] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT '96: Proc. of the 5th International Conference on Extending Database Technology*, pages 3–17, London, UK, 1996. Springer-Verlag.
- [13] X. Sun, M. E. Orłowska, and X. Li. Finding negative event-oriented patterns in long temporal sequences. In H. Dai, R. Srikant, and C. Zhang, editors, *PAKDD*, volume 3056 of *Lecture Notes in Computer Science*, pages 212–221. Springer, 2004.
- [14] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, Vol. 22(No. 3):381–405, 2004.
- [15] X. Yuan, B. P. Buckles, Z. Yuan, and J. Zhang. Mining negative association rules. In *ISCC '02: Proc. of the 17th International Symposium on Computers and Communications (ISCC'02)*, page 623, Washington, DC, USA, 2002. IEEE Computer Society.
- [16] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.