Mining Both Positive and Negative Impact-Oriented Sequential Rules from Transactional Data*

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

[1] Data Sciences and Knowledge Discovery Lab
Faculty of Engineering & IT, University of Technology, Sydney, Australia
{yczhao,hfzhang,lbcao,chengqi}@it.uts.edu.au
[2] Projects Section, Business Integrity Programs Branch, Centrelink, Australia
hans.bohlscheid@centrelink.gov.au

Abstract. Traditional sequential pattern mining deals with positive correlation between sequential patterns only, without considering negative relationship between them. In this paper, we present a notion of *impact-oriented negative sequential rules*, in which the left side is a positive sequential pattern or its negation, and the right side is a predefined outcome or its negation. *Impact-oriented negative sequential rules* are formally defined to show the impact of sequential patterns on the outcome, and an efficient algorithm is designed to discover both positive and negative impact-oriented sequential rules. Experimental results on both synthetic data and real-life data show the efficiency and effectiveness of the proposed technique.

Keywords: negative sequential rules, sequential pattern mining.

1 Introduction

Association rule mining [1] and sequential pattern mining [2] were proposed over a decade ago, and have been well developed and studied by many researchers. Traditional association rules and sequential patterns study only the co-occurrence of itemsets/events, that is, the positive relationship between itemsets/events. However, it is sometimes interesting to find negative correlation, such as two items are seldom bought together in a same basket, or one item is seldom bought after another item. Recently, a couple of techniques have been designed to find *negative association rules* [3,10,12]. However, *negative sequential patterns* are still seldom studied.

Previously we have introduced *event-oriented negative sequential rules* in the form of $P \rightarrow \neg e$, $\neg P \rightarrow e$ or $\neg P \rightarrow \neg e$, where P is a positive sequential pattern

* 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.

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 656–663, 2009.
© Springer-Verlag Berlin Heidelberg 2009

and e denotes a single event [14]. However, in many real-world applications, users are not interested in negative sequential rules associated with all possible events, but only those rules associated with a special target outcome, e.g., fraud or no fraud, debt or no debt, buy or not buy, etc. That is, the target event is the occurrence or non-occurrence of a specific outcome, instead of an arbitrary event. For example, for web click-stream analysis in online retail, an analyst may want to find the relationship between webpage visiting sequences and whether a user buys something. 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 an important target of analysis.

To tackle the above problem, we develop in this paper an idea of *impact-oriented negative sequential rules*, where the left side is a traditional positive sequential pattern or its negation and the right side is a target outcome or its negation. A new efficient algorithm is designed for mining such rules and two novel metrics are defined to measure the impact on outcome.

2 Related Work

The technique of negative association rules has been well studied [3,10,12]. Negative association rules are defined in the form of $A \rightarrow \neg B$, $\neg A \rightarrow B$ and $\neg A \rightarrow \neg B$ [12]. Savasere et al. designed negative association rules as $A \not\rightarrow B$ [10]. Antonie and Zaïane defined *generalized negative association rule* as a rule containing a negation of an item, such as $A \wedge \neg B \wedge \neg C \wedge D \rightarrow E \wedge \neg F$, and defined *confined negative association rules* as $A \rightarrow \neg B$, $\neg A \rightarrow B$ and $\neg A \rightarrow \neg B$ [3].

The idea of sequential patterns was proposed in 1995 to find frequent sequential patterns in sequence data [2]. Some well-known algorithms for sequential pattern mining are AprioriALL [2], FreeSpan [6], PrefixSpan [9], SPADE [13] 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 focus on positive patterns, and negative sequential patterns are underdeveloped. Some reported researches on negative sequential patterns are as follows. Sun et al. proposed negative event-oriented patterns [11] 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 unexpectedly rare in T-sized intervals before target events. 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. Ouyang and Huang proposed negative sequences as $(A, \neg B)$, $(\neg A, B)$ and $(\neg A, \neg B)$ [8]. Lin et al. designed an algorithm NSPM (Negative Sequential Patterns Mining) for mining negative sequential patterns [7].

3 Mining Impact-Oriented Sequential Rules

3.1 Negative Sequential Rules

The negative relationships in transactional data are defined as follows.

Definition 1 (Negative Sequential Rules (NSR)). *A negative sequential rule is in the form of* $A \rightarrow \neg B$, $\neg A \rightarrow B$ *or* $\neg A \rightarrow \neg B$, *where* A *and* B *are positive sequential patterns composed of items in time order.*

Definition 2 (Event-oriented Negative Sequential Rules (ENSR)). *An event-oriented negative sequential rule is a special NSR, where the right side* B *is a single event, that is, the length of* B *is one.*

Definition 3 (Impact-oriented Negative Sequential Rules (INSR)). *An impact-oriented negative sequential rule is a special ENSR, where the right side is a predefined target outcome* T, *such as a specific class or a predetermined event.*

Definition 4 (Negative Sequential Patterns (NSP)). *A negative sequential pattern is a sequence of the occurrence or non-occurrence of items in time order, with at least one negation in it.*

Definition 5 (Generalized Negative Sequential Rules (GNSR)). *A generalized negative sequential rule is in the form of* $A \rightarrow B$, *where one or both of* A *and* B *are negative sequential patterns.*

Based on the above definitions, we can get: $I_{\text{GNSR}} \supset I_{\text{NSR}} \supset I_{\text{ENSR}} \supset I_{\text{INSR}}$, where I_{GNSR}, I_{NSR}, I_{ENSR} and I_{INSR} denotes respectively the sets of the above four kinds of rules. Although an INSR looks similar to an ENSR, the former is specially focused on a specific subset of ENSRs, so it demands more efficient techniques tailored for its special needs.

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 pattern B. We refer to such positive rules as Type I sequential rules. By changing A or/and B to its/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 pattern B;
- Type III: $\neg A \rightarrow B$, which means that if pattern A does not appear, then pattern B will occur; and
- Type IV: $\neg A \rightarrow \neg B$, which means that if pattern A doesn't appear, then pattern B will not occur.

For types III and IV whose left sides are the negations of sequences, the meaning of the rules is: if A doesn't 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

Table 1. Supports, confidences and lifts of four types of sequential rules

Type	Rules	Support	Confidence	Lift
I	$A \rightarrow B$	$P(AB)$	$\frac{P(AB)}{P(A)}$	$\frac{P(AB)}{P(A)P(B)}$
II	$A \rightarrow \neg B$	$P(A)-P(AB)$	$\frac{P(A)-P(AB)}{P(A)}$	$\frac{P(A)-P(AB)}{P(A)(1-P(B))}$
III	$\neg A \rightarrow B$	$P(B)-P(A\&B)$	$\frac{P(B)-P(A\&B)}{1-P(A)}$	$\frac{P(B)-P(A\&B)}{P(B)(1-P(A))}$
IV	$\neg A \rightarrow \neg B$	$1-P(A)-P(B)+P(A\&B)$	$\frac{1-P(A)-P(B)+P(A\&B)}{1-P(A)}$	$\frac{1-P(A)-P(B)+P(A\&B)}{(1-P(A))(1-P(B))}$

patterns, which makes them different from negative association rules. However, if time constraint is considered in sequential rules, the last 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 four types of sequential rules are shown in Table 1. In the table, $P(A\&B)$ denotes the probability of the concurrence of A and B in a sequence, no matter which one occurs first, or whether they are interwoven.

3.2 Algorithm for Mining Impact-Oriented Sequential Rules

To discover impact-oriented negative sequential rules, we use SPAM (Sequential PAttern Mining) [4] as a start point, because it was demonstrated by Ayres et al. to be more efficient than SPADE and Prefixpan [4], another two well-known algorithms for sequential pattern mining. SPAM is very efficient in that it uses bitmap to count the frequency of sequences. 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 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. In this paper, we consider transaction with one item only and an element in the sequence is a single item, instead of an itemset. Therefore, only S-step from SPAM is used in our technique.

Figure 1 gives the pseudocode for finding impact-oriented negative sequential rules, which is based on the function "FindSequentialPatterns" from SPAM [4]. Lines 2-17 show the code for appending the target outcome to a sequential pattern and computing the chi-square and direction for the derived sequential rule. Lines 2-6 use bitmaps to compute the counts, support, confidence and lift for the sequential rule. Lines 7-17 compute the observed frequencies and expected frequencies, and then calculate chi-square and direction. Lines 19-23 generate positive sequential patterns. Lines 25-32 are the S-step of SPAM, which tries to extend the sequential pattern at current node by appending an additional item to it. Lines 34-43 generate three types of negative sequential patterns.

3.3 New Metrics for Impact-Oriented Sequential Rules

Two new metrics, *contribution* and *impact*, are designed as follows to select interesting impact-oriented sequential rules.

```
ALGORITHM: FindINSR - a recursive call that goes down the lattice to find INSR
INPUT: curNode: information about the current node
OUTPUT: impact-oriented negative sequential rules
 1: /* Assume that n is the number of customers. */
 2: cntA = curNode → count; cntT=targetEventCount;
 3: bitmapAT =SequentialAnd(bitmapA, bitmapT); cntAT = bitmapAT → Count();
 4: bitmapAorT =Or(bitmapA, bitmapT); cntAorT = bitmapAorT → Count();
 5: cntAandT = cntA + cntT − cntAorT;
 6: supp = cntAT/n; conf = cntAT/cntA; lift = cntAT∗n/(cntA∗cntT);
 7: /* observed frequencies of AT, A¬T, ¬AT and ¬A¬T */
 8: f1 = cntAT; f2 = cntA − cntAT;
 9: f3 = cntT − cntAT; f4 = n − cntA − cntT + cntAT;
10: /* expected frequencies of AT, A¬T, ¬AT and ¬A¬T*/
11: ef1 = cntA ∗ cntT/n; ef2 = cntA ∗ (1 − cntT/n);
12: ef3 = (1 − cntA/n) ∗ cntT; ef4 = (1 − cntA/n) ∗ (n − cntT);
13: chiSquare = Σ (fi−efi)^2/efi;
14: IF chiSquare < 3.84 /* 95% confidence to reject the independence assumption */
15:     direction = 0;
16: ELSE IF lift > 1 THEN direction = +1, ELSE direction = −1;
17: END IF
18: IF cntAT ≥ minsupp ∗ n
19:     /* generating positive sequential patterns */
20:     IF direction = +1
21:         compute supp, conf and lift for Type I rule based on Table 1;
22:         output "A → T" when supp ≥ minsupp, conf ≥ minconf and lift ≥ minlift;
23:     END IF
24:     /* s-step*/
25:     FOR each possible s-extension i from this level
26:         tempAndBitmap=Bit-Wise-And(bitmap of curNode, bitmap of i);
27:         cntAB=tempAndBitmap.count; /* corresponding to P(AB) */
28:         IF cntAB ≥ minsupp ∗ n
29:             add i to nextNode's s-extension list;
30:             FindINSR(nextNode); /* checking the node at next level */
31:         END IF
32:     END FOR
33: ELSE
34:     /* generating negative sequential patterns */
35:     IF direction = −1
36:         compute supp, conf and lift for Type II rule based on Table 1;
37:         output "A → ¬T" if supp ≥ minsupp, conf ≥ minconf and lift ≥ minlift;
38:         compute supp, conf and lift for Type III rule based on Table 1;
39:         output "¬A → T" if supp ≥ minsupp, conf ≥ minconf and lift ≥ minlift;
40:     ELSE IF direction = +1
41:         compute supp, conf and lift for Type IV rule based on Table 1;
42:         output "¬A → ¬T" if supp ≥ minsupp, conf ≥ minconf and lift ≥ minlift;
43:     END IF
44: END IF
```

Fig. 1. Pseudocode for discovering impact-oriented negative sequential rules

Definition 6 (Contribution). *For a sequential rule $P \rightarrow T$, where P is a sequential pattern, assume i to be the last item in P. The contribution of i to the occurrence of outcome T in rule $P \rightarrow T$ is*

$$contribution(i, P) = \frac{lift(P \rightarrow T)}{lift(P \setminus i \rightarrow T)} \tag{1}$$

where $P \setminus i$ denotes the sequential pattern derived by removing i from P.

Definition 7 (Impact). *For the above rule and i, the impact of i on the outcome in the rule is*

$$impact(i, P) = \begin{cases} contribution(i, P) - 1 : if\ contribution \geq 1, \\ \frac{1}{contribution(i,P)} - 1 \quad : otherwise. \end{cases} \tag{2}$$

Contribution shows how much the last item i in the rule contributes to the occurrence of the outcome T, and impact measures how much it can change the outcome. Both of them fall in $[0, +\infty)$.

4 Experimental Results

4.1 Performance and Scalability

Our designed algorithm (referred to as INSR) was implemented with C++ based on SPAM [4], and its performance and scalability was tested on synthetic datasets generated with IBM data generator [2]. All the tests were conducted on a PC with Intel Core 2 CPU of 1.86GHz, 2GB memory and Windows XP Pro. SP2. The number of items per transaction was set to one when generating data.

Our algorithm was first tested on a dataset with 50,000 customers, 40 items per sequence and the length of maximal patterns as 13. The minimum supports range from 0.2 to 0.7, and the results are shown in Figure 2a. From the figure, both INSR and Spam [4] run faster with larger minimum support, because the search space becomes smaller. Moreover, INSR runs faster than Spam, and the reason is that, when a pattern A is frequent and $A \bowtie T$ is infrequent, INSR doesn't search A's children nodes, but Spam continues checking all its descendants until it becomes infrequent.

The scalability with the number of sequences was tested on datasets with average sequence length as 30, length of maximal patterns as 11. The number of customers ranges from 10,000 to 100,000, and the support threshold is set to 0.3. Figure 2b shows the result of the above test. It's clear from the figure that INSR is linear with the number of sequences.

The running time with varying sequence lengths is shown Figure 2c, where the datasets used have 50,000 customers, with length of maximal patterns as 10, and the average sequence length ranging from 10 to 45. The support threshold is set to 0.3. The figure shows that the running time becomes longer with the increase of the average number of items per sequence and that INSR is almost linear with the length of sequences.

4.2 Selected Results in a Case Study

The proposed technique was applied to the real data from Centrelink, Australia. Centrelink is a Commonwealth Government agency distributing social welfare payments to entitled customers. For various reasons, customers on benefit payments or allowances sometimes get overpaid and these overpayments lead to debts owed to Centrelink. We used impact-oriented negative sequential rules to find the relationship between transactional activity sequences and debt occurrences, and also find the impact of additional activities on debt occurrence.

A sample of historical transactional data from July 2007 to February 2008 were used for the analysis. After data preprocessing, 15,931 sequences were constructed. Minimum support was set to 0.05, that is, 797 out of 15,931 sequences. There are 2,173,691 patterns generated and the longest pattern has 16 activities. Some selected sequential rules are given in Table 2, where "DEB" stands for debt and the other codes are activities. "Direction" shows whether the pattern is positively (+1) or negatively (-1) associated with debt occurrence.

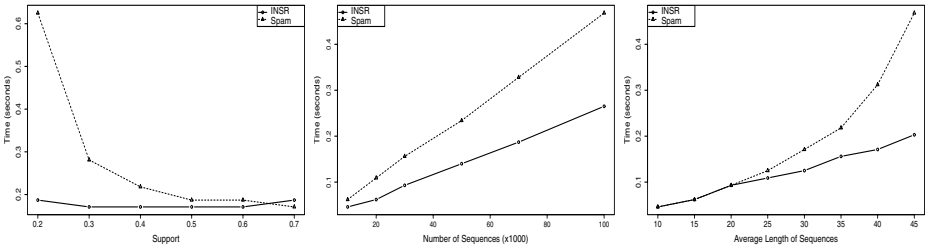

Fig. 2. Scalability with (a) support; (b) the number of sequences; and (c) the length of sequences (from left to right)

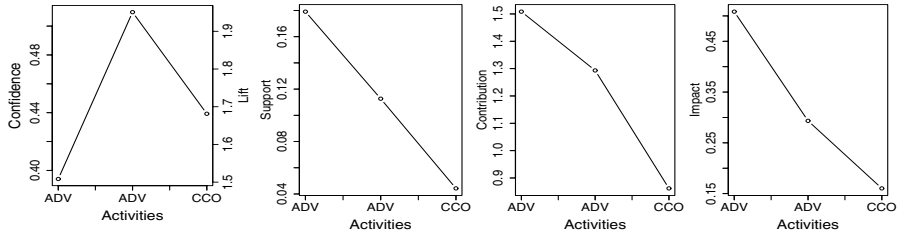

Fig. 3. A growing sequential pattern "ADV ADV CCO"

Table 2. Selected positive and negative sequential rules

Type	Rule	Supp	Conf	Lift	Direction
I	REA ADV ADV→DEB	0.103	0.53	2.02	+1
I	RPR ANO→DEB	0.111	0.33	1.25	+1
I	STM PYI→DEB	0.106	0.30	1.16	+1
II	MND→ ¬DEB	0.116	0.85	1.15	-1
II	REA PYR RPR RPT→ ¬DEB	0.176	0.84	1.14	-1
II	REA CRT DLY→ ¬DEB	0.091	0.83	1.12	-1
III	¬{PYR RPR REA STM}→DEB	0.169	0.33	1.26	-1
III	¬{PYR CCO}→DEB	0.165	0.32	1.24	-1
III	¬{PLN RPT}→DEB	0.212	0.28	1.08	-1
IV	¬{REA EAN}→ ¬DEB	0.650	0.79	1.07	+1
IV	¬{DOC FRV}→ ¬DEB	0.677	0.78	1.06	+1

Figure 3 shows an example of discovered growing sequential pattern,

$$\begin{cases} ADV \rightarrow DEB \\ ADV, ADV \rightarrow DEB \\ ADV, ADV, CCO \rightarrow DEB \end{cases}. \tag{3}$$

Each point in every chart gives the value for the sequential pattern from the first activity to the corresponding activity. All four charts in Figure 3 show the growth from "ADV" to "ADV ADV" and "ADV ADV CCO". ADV increases the probability of debt occurrence, because its confidence in debt occurrence is 0.395, 1.5 times the likelihood of debt occurrence in the whole population (see the first chart). There are 18% of all sequences supporting that ADV is followed by debt (see the second chart). As shown in the third chart, the two ADVs contributes to debt occurrence, but CCO contributes negatively, as its contribution is less than one. The impacts of two ADVs on outcome are different, with the first one having larger impact (see the fourth chart).

5 Conclusions

We have defined impact-oriented negative sequential rules and have designed an efficient algorithm for mining such sequential rules. We have also designed two metrics, contribution and impact, to measure the effect of an item on the outcome, which help to select interesting growing sequential patterns. A case study has been presented to show the effectiveness of the proposed technique.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data, Washington D.C., USA, May 1993, pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Yu, P.S., Chen, A.S.P. (eds.) Proc. of the 11th Int. Conf. on Data Engineering, Taipei, Taiwan, pp. 3–14 (1995)
3. Antonie, M.-L., Zaïane, O.R.: Mining positive and negative association rules: an approach for confined rules. In: Proc. of the 8th Eur. Conf. on Principles and Practice of Knowledge Discovery in Databases, New York, USA, pp. 27–38 (2004)
4. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In: KDD 2002: Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 429–435 (2002)
5. Bannai, H., Hyyro, H., Shinohara, A., Takeda, M., Nakai, K., Miyano, S.: Finding optimal pairs of patterns. In: Jonassen, I., Kim, J. (eds.) WABI 2004. LNCS (LNBI), vol. 3240, pp. 450–462. Springer, Heidelberg (2004)
6. Han, J., Pei, J., et al.: Freespan: frequent pattern-projected sequential pattern mining. In: KDD 2000: Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 355–359 (2000)
7. Lin, N.P., Chen, H.-J., Hao, W.-H.: Mining negative sequential patterns. In: Proc. of the 6th WSEAS Int. Conf. on Applied Computer Science, Hangzhou, China, pp. 654–658 (2007)
8. Ouyang, W.-M., Huang, Q.-H.: Mining negative sequential patterns in transaction databases. In: Proc. of 2007 Int. Conf. on Machine Learning and Cybernetics, Hong Kong, China, pp. 830–834 (2007)
9. Pei, J., Han, J., et al.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: ICDE 2001: Proc. of the 17th Int. Conf. on Data Engineering, Washington, DC, USA, p. 215 (2001)
10. Savasere, A., Omiecinski, E., Navathe, S.B.: Mining for strong negative associations in a large database of customer transactions. In: ICDE 1998: Proc. of the 14th Int. Conf. on Data Engineering, Washington, DC, USA, pp. 494–502 (1998)
11. Sun, X., Orlowska, M.E., Li, X.: Finding negative event-oriented patterns in long temporal sequences. In: Proc. of the 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Sydney, Australia, pp. 212–221 (May 2004)
12. Wu, X., Zhang, C., Zhang, S.: Efficient mining of both positive and negative association rules. ACM Transactions on Information Systems 22(3), 381–405 (2004)
13. Zaki, M.J.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning 42(1-2), 31–60 (2001)
14. Zhao, Y., Zhang, H., Cao, L., Zhang, C., Bohlscheid, H.: Efficient mining of event-oriented negative sequential rules. In: Proc. of the 2008 IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI 2008), Sydney, Australia, pp. 336–342 (December 2008)