Journal of Intelligent & Fuzzy Systems 29 (2015) 2759–2767
DOI:10.3233/IFS-151980
IOS Press

Select actionable positive or negative sequential patterns

Xiangjun Dong[a,*], Chuanlu Liu[a,*], Tiantian Xu[a] and Dakui Wang[b,*]
[a]*School of Information, Qilu University of Technology, Jinan, China*
[b]*International School of Software, Wuhan University, Wuhan, China*

Abstract. Negative sequential patterns (NSP) refer to sequences with non-occurring and occurring items, and can play an irreplaceable role in understanding and addressing many business applications. However, some problems occur after mining NSP, the most urgent one of which is how to select the actionable positive or negative sequential patterns. This is due to the following factors: 1) positive sequential patterns (PSP) mined before considering NSP may mislead decisions; and 2) it is much more difficult to select actionable patterns after mining NSP, as the number of NSPs is much greater than PSPs. In this paper, an improved method of pruning uninteresting itemsets to fit for a selecting actionable sequential pattern (ASP) is proposed. Then, a novel and efficient method, called SAP, is proposed to select the actionable positive and negative sequential patterns. Experimental results indicate that SAP is very efficient in the selection of ASP. To the best of our knowledge, SAP is the best method for the selection of actionable positive and negative sequential patterns.

Keywords: Actionable, negative sequential patterns, positive sequential patterns

1. Introduction

Negative sequential patterns (NSP) refer to sequences with non-occurring and occurring items, while sequential patterns that contain only occurring items are called positive sequential patterns (PSP) [19, 24, 26]. For instance, assume that $p_1 = <abcX>$ is a PSP; $p_2 = <abcY>$ is an NSP, where a, b and c represent medical service codes that a patient has received in a healthcare setting, and X and Y each stand for a disease status. p_1 indicates that a patient who usually receives medical services a, b and then c is likely to have disease status X, whereas p_2 indicates that patients receiving treatment of a and b but not c have a high probability of disease status Y. It is increasingly recognized that such NSPs can play an irreplaceable role in understanding and addressing many business applications, such as associations between treatment services and illnesses. However, some urgent problems occur after mining NSP. These urgent problems include: 1) difficult in selecting actionable positive or negative sequential patterns; 2) the need to improve efficiency of the NSP mining algorithm; 3) how to decrease the number of negative sequential candidates (NSC). Among these problems, the selection problem is the most urgent.

One reason for this is that the PSPs being mined before considering NSP may mislead decisions. For example, a PSP <*abc*> was mined and used to make decisions before considering NSP. However, based on <*abc*>, NSPs such as <*abc*>, <*abc*>, <*abc*> and <*abc*> may be obtained. Obviously, not all of these

*Corresponding author. X.J. Dong and C.L. Liu, School of Information, Qilu University of Technology, Jinan 250353, China. Tel.: +86 13853166812; Fax: +86 531 89631256; E-mail: d-xj@163.com (X.J. Dong); Tel.: +86 18765832249; Fax: +86 531 89631518; E-mail: chuanluliu@163.com (C.L. Liu) and D.K. Wang, International School of Software, Wuhan University, Wuhan 430079, China. Tel.: +86 18515130815; Fax: +86 27 59238813; E-mail: dakui1988@foxmail.com

ISSN 1064-1246/15/$35.00

patterns are actionable. If <*abc*> is not actionable now, the previous decisions that are made based on it would be incorrect.

Another reason is that it is much more difficult to select actionable patterns after mining NSP, because the number of NSPs is much greater than PSPs. To a k-size of PSP p, the number of potential negative sequential patterns based on p, $|NSP_p|$, according to a NSP mining algorithm e-NSP [24], is $\sum_{m=1}^{\lceil k/2 \rceil} \frac{(k-m+1)!}{m!*(k-2m+1)!}$, where $\lceil k/2 \rceil$ is a minimum integer that is no less that k/2. For instance, $|NSP_p| = 54$ when $k = 8$. Not only are some of those patterns entirely actionable, but it is very difficult to select actionable one(s) from among them.

Although there have been some reports of actionable knowledge discovery [8, 9, 10, 16] and selecting actionable patterns/rules or interestingness measures in association rule mining [1, 2, 4, 15, 17], none of the previous research considers how to select actionable positive and negative sequential patterns. Very few papers study NSP mining [12, 19, 20, 21, 24, 26, 27], and most primarily focus on how to design a mining algorithm and how to improve the algorithm's efficiency. Therefore, this paper investigates association rule mining. In this area, some methods have been proposed to prune uninteresting itemsets, to select actionable patterns, to mine positive and negative association rules, and so on [2, 4, 15, 17]. Among these methods, the one proposed by X.D. Wu, et al. is one of the most suitable methods, referred to as Wu's method for convenience [23]. Wu's method can prune the itemsets that are not of interest before mining positive and negative association rules. However, due to the differences between the association rule and sequential pattern, this method cannot be directly used to solve the selection problem and must be improved before use.

No studies shave yet applied Wu's method to positive or negative sequential patterns, and Wu's method can only deal with itemsets, not actionable sequential patterns. Therefore, in this paper, we first improve Wu's method to fit a selecting actionable sequential pattern (ASP), then propose a novel and efficient method, called SAP, to select actionable positive and negative sequential patterns based on the improved method. Experimental results indicate that SAP is very efficient in the selection of ASP. To the best of our knowledge, this is the best method to select actionable positive and negative sequential patterns.

The remainder of the paper is organized as follows. Section 2 discusses the related work. In Section 3 reviews an e-NSP algorithm and Wu's method, and then proposes the improved method and SAP method. Section 4 presents the experiment results, and conclusions and future work are presented in Section 5.

2. Related work

Some literature has proposed a few measures to mine interesting association rules, such as correlation coefficients, chi-squared tests, interestingness, Laplace, the Gini-index, Piatetsky-Shapiro method, conviction and so on [2, 4, 15, 17]. M.L. Antonie, O.R. Zaiane, X.J. Dong, Z.D. Niu, X.L. Shi, X.D. Zhang and D.H. Zhu used correlation coefficient to mine positive and negative association rules [11, 25]. B. Liu, W. Hsu and Y.M. Ma used chi-squared test to prune non-actionable rules or rules at lower significance levels [3].

Some literature has discussed the selection of actionable knowledge or actionable patterns. L.B. Cao, C.Q. Zhang, et al. developed a new data mining framework, referred to as Domain-Driven In-Depth Pattern Discovery (DDID-PD) [8]. L.B. Cao, Y.C. Zhao, et al. formally defined the actionable knowledge discovery (AKD) concepts, processes, the action ability of patterns, and operable deliverables. With such components, four types of AKD frameworks are proposed to handle various business problems and applications [9]. L.B. Cao discussed a paradigm shift in knowledge discovery from data to actionable knowledge discovery and delivery. In post-analysis [10], a key component is to extract actions from learned rules [23]. A typical method applied to learning action rules is to split attributes into "hard/soft" [23] to extract actions that may improve the loyalty or profitability of customers. G. Adomavicius and A. Tuzhilin proposed a method of discovery of actionable patterns, which first builds an action tree for the specific application, and then assigns actionable patterns to the corresponding nodes of the tree by data mining queries [5]. K. Kavitha and E. Ramaraj proposed an efficient transaction reduction algorithm, TR-BC, to mine the frequent patterns based on bitmap and class labels [6]. K. Wang, Y. Jiang and A. Tuzhilin introduced the notion of "action" as a domain-independent method to model the domain knowledge. This presents several pruning strategies to reduce the search space and algorithms for mining all actionable patterns, as well as mining the top k actionable patterns [7]. P. Kanikar and D.K. Shah extracted actionable association rules from multiple datasets [14].

Next, the literature on mining NSP is discussed.

S.C. Hsueh, M.Y. Lin and C.L. Chen proposed a PNSP approach for mining positive and negative

sequential patterns in the form of $< (abc)\neg(de)\ (ijk) >$ [19]. Z.G. Zheng, Y.C. Zhao, Z. Zuo and L.B. Cao proposed an NegGSP approach to mine NSP. The general idea of NegGSP is to mine PSPs by GSP first, then to generate and prune NSCs. After that, it counts the support of NSCs by re-scanning databases to identify negative patterns [26]. N.P. Lin, H.J. Chen and W.H. Hao only mined NSPs whose final element is negative and other elements are positive [12]. W.M. Ouyang and Q.H. Huang only identified NSP in the form of $(\neg A, B)$, $(A, \neg B)$ and $(A, \neg B)$ [21], which is similar to mine negative association rules [13, 23].

This requires $A \cap B = \emptyset$, which is a usual constraint to association rule mining but is represents a very strict constraint to sequential pattern mining. V.K. Khare and V. Rastogi mined NSP in the same form as W.M. Ouyang and Q.H. Huang in incremental transaction databases [12, 21]. Z.G. Zheng, Y.C. Zhao, Z. Zuo and L. Cao proposed an approach based on genetic algorithms to mine NSPs. The approach borrows the idea of gene evolution, to obtain candidates by crossover and mutation, and proposed a dynamic fitness function to generate as many candidates as possible and to avoid population stagnation [27].

X.J. Dong, Z.G. Zheng, L.B. Cao, Y.C. Zhao, et al. proposed an efficient e-NSP algorithm to mine NSP. In this paper, e_NSP is used to mine PSPs and NSPs. The details of e_NSP will be introduced in Section 3 [24].

3. SAP method

3.1. Basic concept

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items; an itemset is a subset of I and a sequence is an ordered list of itemsets. A sequence s is denoted by $< s_1, s_2 \ldots s_1 >$, where $s_j \subseteq$ I for $1 \leq j \leq 1$. s_j is also designated as an element of the sequence, and denoted as $(x_1, x_2 \ldots x_m)$, where x_k is an item, $x_k \in I$ for $1 \leq k \leq m$. For brevity, the brackets are omitted if an element has only one item; for example, element (x) is written as x. The order number j of element s_j is called order_id of s_j, denoted as $id(s_j)$, i.e., $id(s_j) = j$. An item can occur once, at most, in an element of a sequence, but can occur multiple times in different elements of a sequence. The length of sequence s, denoted as length(s), is the total number of items in all the elements in s; sequence s is a k-length sequence if length(s) = k. The size of sequence s, denoted as size(s), is the total number of elements in s; sequence s is a k-size sequence if size(s) = k. For example, $s_1 =< abc >$ is a 3-length, 3-size sequence; $s_2 =< (ab)c >$ and $s_3 =< a(bc) >$ are both 3-length, 2-size sequences.

Sequence $\alpha =< \alpha_1, \alpha_2 \ldots \alpha_n >$ is a subsequence of sequence $\beta =< \beta_1, \beta_2 \ldots \beta_m >$ and β is a super sequence of α, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ such that β contains α. For example, $< a(cd) >$ is a subsequence of $< ab(cde) >$, $< ab(cde) >$ is a super subsequence of $< a(cd) >$ also denoted as $< ab(cde) >$ contains $< a(cd) >$.

A sequence database D is a set of tuples $< sid, ds >$, where sid is a $sequence\_id$, and ds is a data sequence. The number of tuples in D is denoted as $|D|$. The set of tuples containing sequence α is denoted as $\{< \alpha >\}$. The support of α, denoted as $s(\alpha)$, is the ratio of the number of $\{< \alpha >\}$ to the total number of tuples in D, i.e., $s(\alpha) = |\{< \alpha >\}|/|D| = |\{< sid, ds > | < sid, ds > D\ (\alpha\ ds)\}|/\ |D|$. The ms is a minimum support threshold given by users or experts. Sequence α is called a (positive) sequential pattern, or α is frequent, if $s(\alpha) \geq ms$; α is infrequent, if $s(\alpha) < ms$. The task of positive sequential pattern mining is to discover the set of all positive sequential patterns.

The concepts of length and size for positive sequences are also suitable for negative sequences. The neg-size of sequence ns, denoted as neg-size(ns), is the total number of negative elements in ns; ns is a k-neg-size negative sequence if neg-size(ns) = k. For example, $ns =< \neg a\ (ab)ca\neg c >$ is a 5-size and 2-neg-size negative sequence because size(ns) = 5 and neg-size(ns) = 2.

3.2. Review of e_NSP

3.2.1. Three constraints in e_NSP

An e_NSP adds three constraints to negative sequences in order to decrease the number of NSCs and discover only meaningful NSPs.

Constraint 1 is the element negative constraint. That is, the minimum negative unit in am NSC is an element; not only one part of the items in the element (if the element includes more than one item) is negative. For example, $< a\neg(ab)ca\neg c >$ satisfies this constraint, but not the $< a(\neg ab)ca\neg c >$ because only a is negative in element $(\neg ab)$.

Constraint 2 is formation constraint. There are no contiguous negative elements in an NSC because right order of two contiguous negative elements cannot be determined if there are no positive elements between them. For example, $< a\neg(ab)ca\neg c >$ satisfies constraint 2, but not the $< a\neg(ab)c\neg a\neg c >$.

Constraint 3 is frequency constraint. The e_NSP only mined the NSP whose positive partner is a PSP. The positive partner of a negative sequence changes all negative elements in ns to their corresponding elements, denoted as $p(ns)$. For example, $p(< a\neg(ab)ca\neg c >) = < a\ (ab)cac >$

3.2.2. The main steps of e_NSP

There are four steps in e_NSP.

- The first step is to mine all PSPs according to the classical PSP mining algorithms. To each PSP p, save its support and all sid of the data sequence that contain the p to a set {p}.
- Based on these PSPs, generate all NSCs according to the following method.
- For a k-size PSP, its NSCs are generated by changing any m non-contiguous element to its negative element, $m = 1, 2, \ldots, [k/2]$, where $[k/2]$ is a minimum integer that is no less than $k/2$. For example, the NSCs based on $< abcde >$ include: $m = 1$, $< abcde >$, $< abcde >$, $< abcde >$, $< abcde >$, $< abcde >$; $m = 2$, $< abcde >$, $< abcde >$, $< abcde >$, $< abcde >$, $< abcde >$, $< abcde >$; $m = 3$, $< abcde >$.
- Convert the negative containment to a positive containment.
- For example, if a data sequence ds contains negative sequence $< \neg ab\neg cd\neg e >$, then ds must contain (positive) sequence $< bd >$ and must not contain (positive) sequence $< abd >$, $< bcd >$ and $< bde >$.
- Calculate the support of each NSC and determine whether it is a NSP by comparing its support with the minimum threshold ms.

For example, $s(<abcde>) = s(<bd>) - |\{<abd>\} \cup \{<bcd>\} \cup \{<!bde>\}|/|D|$, where $\{<abd>\}$, $\{<bcd>\}$ and $\{<bde>\}$ are the *sid* set that contain $< abd >$, $< bcd >$ and $< bde >$ respectively, and $|\{<abd>\} \cup \{<bcd>\} \cup \{<bde>\}|$ is the *sid* number of the union set.

3.3. The original Wu's method

Wu's method defines an interestingness function $interest(X, Y) = |s(X \cup Y)| - s(X)s(Y)|$ and a threshold mi (minimum interestingness) [23]. If interest (X, Y) mi, the rule of X and Y is of potential interest, and X and Y are referred to as a potentially interesting itemset. The details are as follows.

I is a *frequent itemset of potential interest* if $fipi(I) = s(I) \geq ms \wedge$

$$(\exists X, Y : X \cup Y = I) \wedge \tag{1}$$

where

$$fipis(X, Y) = (X \cap Y = \Phi) \wedge \tag{2}$$

$$(f(X, Y, ms, mc, mi) = 1), f(X, Y, ms, mc, mi) = \frac{s(X \cup Y + c(X \Rightarrow Y) + interest(X, Y) - (ms + mc + mi) + 1)}{|s(X \cup Y) - ms| + |c(X \Rightarrow Y) - mc| + |interest(X, Y) - mi| + 1} \tag{3}$$

where f() is a constraint function concerning the support, confidence, and interestingness of X and Y; and ms, mc, mi are the minimum support, minimum confidence, and minimum interestingness provided by users or experts.

Using this method, Wu's method can establish an effective pruning strategy for efficiently identifying all frequent itemsets. However, due to the differences between itemsets and sequential patterns, this method can't be directly used to solve the selection problem and must be improved before use.

3.4. The improved Wu's method

The improvements cover three aspects; the third improvement is the key improvement.

- Remove the confidence measure from Equations 1–3.

The confidence measure c() and mc are used to discover association rules. If only itemsets/ patterns are discovered, this measure can be removed, as in Wu's method.

- Replace the interestingness function with correlation coefficient function.

X.J. Dong explains that interest (X, Y) is not enough to prune uninteresting itemsets because it is greatly influenced by the value of $s(X \cup Y)$, $s(X)$ and $s(Y)$. For example, interest of (X, Y) is no more than 0.1 if

$s(X \cup Y)$, $s(X)$ and $s(Y)$ are less than 0.1; interest of (X, Y) is no more than 0.01 if $s(X \cup Y)$, $s(X)$ and $s(Y)$ are less than 0.01. $s(X \cup Y)$, $s(X)$ and $s(Y)$ are variable with different datasets or different itemsets, so that it is very difficult for users to designate mi a suitable value [22]. X.J. Dong analyzed this shortcoming and proposed a good substitute: the correlation coefficient [22]. Correlation coefficients can also measure the relationships between X and Y, and change with the degree of linear dependency between X and Y. More details can be found in [22]. Here, we simply adopt this result and use a correlation coefficient instead of interest (X, Y).

The correlation coefficient between X and Y, $\rho(X,\ Y)$, can be calculated as follows:

$$\rho(X,\ Y) = \frac{s(X \cup Y) - s(X)s(Y)}{\sqrt{s(X)(1 - s(X))s(Y)(1 - s(Y))}} \tag{4}$$

where $s() \neq 0, 1$.

$\rho(X,\ Y)$ has the following three possible cases:

* If $\rho(X,\ Y) < 0$, then X and Y are positively correlated. The more X events that occur, the more Y events occur.
* If $\rho(X,\ Y) = 0$, then X and Y are independent (for binary variables). The occurrence of event X has nothing to do with the occurrence of event Y.
* If $\rho(X,\ Y) > 0$, then X and Y are negatively correlated. The more X events occur, fewer Y events occur

The range of $\rho(X,\ Y)$ is between –1 and +1, and the positive correlation is between the absolute value of $\rho(X,\ Y)$, and represents the correlation strength of A and B. Therefore, we can set a threshold ρmin to prune patterns with small correlation strength.

Correspondingly, we replace interest (X, Y) with $\rho(X,\ Y)$ and remove the confidence measure. Then, Equations (2) and (3) are simplified to Equations (5) and (6).

$$fipis(X,\ Y) = (X \cap Y = \Phi) \wedge \tag{5}$$

$$(f(X,\ Y,\ ms,\ \rho_{\min}) = 1) f(X,\ Y,\ ms,\ \rho_{\min}) = \frac{s(X \cup Y) + \rho(X,\ Y) - (ms + \rho_{\min}) + 1}{s(X \cup Y) - ms| + |\rho(X,\ Y) - \rho_{\min}| + 1} \tag{6}$$

– Adapt these equations to fit sequential patterns.

So far, the above discussions are based on mining interesting itemsets, not sequences. The differences between them are that, in a sequence, the itemsets are in an order, and an item can occur at multiple times in different elements of a sequence. Therefore, the conditions $\exists X, Y : X \cup Y = I$ and $X \cap Y = \Phi$ in Equations 4-5 must be changed; how they are changed is the key technique for selecting ASP. Our methods test whether any 2-size sequence constructed by two contiguous elements in a pattern is actionable. That is, if a k-size (k > 1) PSP or NPS $P =< ele2 \ldots ek >$ is actionable, then we require that $< ele2 >, < e2e3 >, \ldots, < ek - 1ek >$ be actionable too. Formally, we have the following definitions.

Definition 1. Actionable sequential pattern.

A k-size (k > 1) PSP or NSP $P =< e1\ e2 \ldots ek >$ is an actionable sequential pattern if $\forall i \in \{2 \ldots k\}$,

$$asp(e_{i-1},\ e_i) = s(< e_i - 1e_i >) \geq ms \wedge (f(e_{i-1},\ e_i,\ ms,\ \rho_{\min}) = 1), \tag{7}$$

where

$$f(e_{i-1},\ e_i,\ ms,\ \rho_{\min}) = \frac{s(< e_{i-1}e_i >) + \rho(e_{i-1},\ e_i) - (ms + \rho_{\min}) + 1}{|s(< e_{i-1}e_i >) - ms| + |\rho(e_{i-1},\ e_i) - \rho_{\min}| + 1} \tag{8}$$

$$\rho(e_{i-1},\ e_i) = \frac{s(< e_{i-1}e_i >) - s(< e_{i-1} >)s(< e_i >)}{\sqrt{s(< e_{i-1} >)(1 - s(< e_{i-1} >)s(< e_i >)(1 - s(< e_i >))}} \tag{9}$$

According to Definition 1, we can obtain a corollary as follows.

Corollary 1. *A k-size* $(k > 1)$ *PSP or NSP* $P =< e_1\ e_2 \ldots e_k >$ *is not an ASP if* $\exists i \in \{2 \ldots k\}$, $< e_{i-1}\ e_i >$ *is not an ASP.*

Proof. From Definition 1, we can see that if $P =< e_1\ e_2 \ldots e_k >$ is an ASP, then $\forall i \in \{2 \ldots k\}$, $< e_{i-1}\ e_i >$ must be an ASP. Otherwise, $P =< e_1\ e_2 \ldots e_k >$ will not be an ASP. This satisfies corollary 1, so corollary 1 is correct.

Definition 1 and correlary 1 are used to implement the proposed SAP method

3.5. SAP method

The primary steps of SAP method are as follows.

Step 1: all PSP and NSP are mined by e-NSP.

Step 2: each 2-size pattern such $s < e_1e_2 >$ are tested by Definition 1. If $< e_1e_2 >$ is not an ASP, then $< e_1e_2 >$ and all the patterns containing $< e_1e_2 >$ are removed.

Step 3: longer sized patterns are tested in a similar way as step 2, until all patterns are tested.

Algorithm SAP.

Input: D: a sequential database; ms: minimum support; ρmin: a threshold defined by user;

Output: ASP: set of actionable sequential patterns;

– let ASP = Φ:

– use e-NSP to mine all PSPs and NSPs and store them in the set {PNSP};

– for k = 2 to maximum length of PSP in {PNSP} do {

– for each k-size pattern $P < e_1 e_2 \ldots e_k >$ in {PNSP} do {

– test P with definition 1;

– if P is an actionable sequential pattern then

– ASP = ASP∪{P};

– else

– remove P and all the patterns contain P from PNSP;

– end if

– }

– k++;

– }

– return ASP;

An example is illustrated below.

The sample database is shown in Table 1, given ms = 0.4, and $\rho_{\min} = 0.3$.

Step 1: Use e-NSP algorithm to mine all PSP and NSP. We obtain:

PSP = {<a>, <b>, <c>, <d>, <e>, <f>, <aa>, <(ab)>, <ab>, <ba>, <ac>, <ca>, <ad>, <ea>, <af>, <(bc)>, <bc>, <cb>, <bd>, <db>, <eb>, <bf>, <fb>, <cc>, <dc>, <ec>, <fc>, <ef>, <(ab)c>, <(ab)d>, <(ab)f>, <aba>, <eab>, <a(bc)>, <abc>, <aca>, <eac>, <acb>, <acc>, <(bc)a>, <adc>, <ebc>, <fbc>, <dcb>, <ecb>, <fcb>, <bdc>, <efb>, <efc>, <(ab)dc>, <a(bc)a>, <eacb>, <efcb>}.

NSP = $\{< \neg f >, < \neg aa >, < a\neg a >, < \neg(ab) >, < \neg ba >, < b\neg a >, < \neg ca >, < c\neg a >, < \neg ad >, < a\neg d >, < \neg ea >, < e\neg a >, < a\neg f >, < \neg(bc) >, < \neg bd >, < b\neg d >, < \neg db >, < d\neg b >, < \neg eb >, < e\neg b >, < b\neg f >, < \neg fb >, < \neg ec >, < e\neg c >, < \neg fc >, < e\neg f >, < \neg(ab)c >, < \neg(ab)d >, < ab\neg a >, < \neg eab >, < a\neg(bc) >, < a\neg bc >, < ab\neg c >, < ac\neg a >, < \neg eac >, < \neg(bc)a >, < a\neg dc >\}$.

Table 1
Sample database

Sid	Data sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>
50	<(de)>

Step 2: test all actionable sequential patterns according to Definition 1 and corollary 1. Some examples are listed below.

For <ab>, s(<ab>) = 0.8, s(<a>) = 0.8, s(<b>) = 0.8

$$\rho(a,\ b) = \frac{0.8 - 0.8 * 0.8}{\sqrt{0.8 * (1 - 0.8) * 0.8 * (1 - 0.8)}} = 1$$

$$f(a,\ b,\ ms\ \rho\min) = \frac{0.8 + 1 - (0.4 + 0.3) + 1}{|0.8 - 0.4| + |1 - 0.3| + 1} = 1$$

Sequence <ab> meets asp(a, b) = s(<ab>) $\geq$ 0.4 $\wedge$ (f (a, b, ms, ρmin) = 1), so it is an asp.

For <cb>, s(<cb>) = 0.6, s(<c>) = 0.8, s(<b>) = 0.8

$$\rho(c,\ b) = \frac{0.6 - 0.8 * 0.8}{\sqrt{0.8 * (1 - 0.8) * 0.8 * (1 - 0.8)}} = -0.25$$

$$f(c,\ b,\ ms,\ \rho\min) = \frac{0.6 - 0.25 - (0.4 + 0.3) + 1}{|0.6 - 0.4| + |-0.25 - 0.3| + 1} = 0.37 \neq 1$$

Therefore <cb> is not an asp.

For $< \neg eab >$, $s(< \neg ea >) = 0.4$, $s(< \neg e >) = 0.2$, $s(< a >) = 0.8$

We require test sequence $< \neg ea >$ and sequence <ab>; if each of these sequences is an ASP, $< \neg eab >$ is also an ASP.

$$\rho(\neg e,\ a) = \frac{0.4 - 0.2 * 0.8}{\sqrt{0.2 * (1 - 0.2) * 0.8 * (1 - 0.8)}} = 1.5$$

$$f(\neg e,\ a,\ ms,\ \rho\min) = \frac{0.4 + 1.5 - (0.4 + 0.3) + 1}{|0.4 - 0.4| + |1.5 - 0.3| + 1} = 1$$

Sequence $< \neg ea >$ meets asp$(\neg e, a) = s(< \neg ea >) \geq 0.4 \wedge (f(\neg e, a, ms, \rho min) = 1)$; test sequence <ab> is an ASP, therefore $< \neg eab >$ is an ASP.

All ASPs are obtained according to the above procedure.

ASP = $\{< ab >, < ac >, < (ab)c >, < (ab)d >, < (ab)f >, < a(bc) >, < (bc)a >, < a(bc)a >, < \neg aa >, < a\neg a >, < \neg ba >, < b\neg a >, < \neg ca >, < c\neg a >, < \neg ad >, < a\neg d >, < \neg ea >, < e\neg a >, < a\neg f >, < \neg bd >, < b\neg d >, < \neg db >, < d\neg b >, < \neg eb >,$

$< e\neg b >, < b\neg f >, < \neg fb >, < \neg ec >, < e\neg c >, < \neg fc >, < e\neg f >, < ab\neg a >, < \neg eab >, < ac\neg a >, < \neg eac >\}$.

4. Experimental results

Experiments were performed on a Pentium 4 Celeron 2.1 G PC with 2 G main memory, running on Microsoft Windows7. All programs were written in MyEclipse 10.

4.1. Datasets

To describe and observe the impact of data characteristics on algorithm performance, the following data factors are used: C, T, S, I, DB and N, which are defined to describe characteristics of sequence data [18].

C: Average number of elements per sequence;

T: Average number of items per element;

S: Average size of maximal potentially large sequences;

I: Average size of items per element in maximal potentially large sequences;

DB: Number of sequences (=size of Database); and

N: Number of items.

Four source datasets are also used in these experiments [24]. They include both real data and synthetic datasets generated by an IBM data generator [18].

Dataset 1 (DS1), C8_T4_S6_I6_DB100k_N0.1k.

Dataset 2 (DS2), C10_T8_S20_I10_DB10k_N0.2k.

Dataset 3 (DS3) is obtained from UCI Datasets. There are 989,818 records; average number of elements in a sequence is 4; and each element only has one item.

Dataset 4 (DS4) is real-application data from the financial service industry. It contains 5,269 customers/sequences; the average number of elements in a sequence is 21; the minimum number of elements in a sequence is 1; and the maximum number is 144.

4.2. Experimental results

In the experiments, different ms and $\rho_{\min}$ values were set to reflect differences in the four datasets. The results of the experiment are as follows:

4.3. Experimental analysis

Figures 1–4 demonstrate that with the increment of $\rho_{\min}$, the number of actionable sequential patterns decreases based on the same minimum support. As

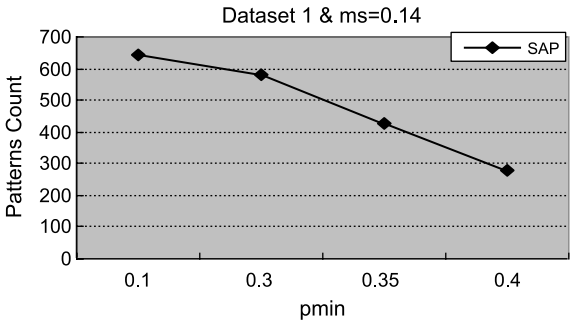

Fig. 1. The count of actionable sequential patterns.

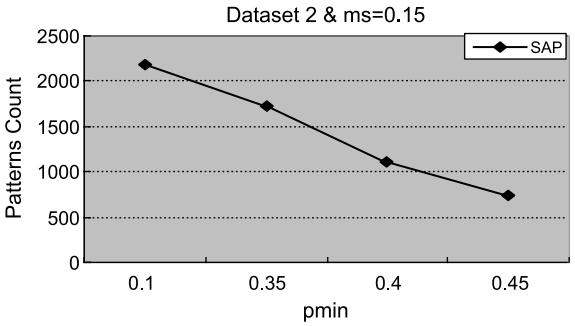

Fig. 2. The count of actionable sequential patterns.

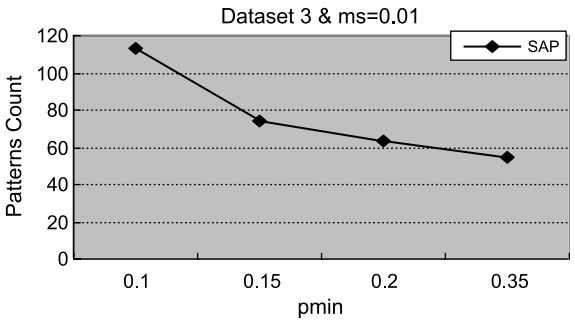

Fig. 3. The count of actionable sequential patterns.

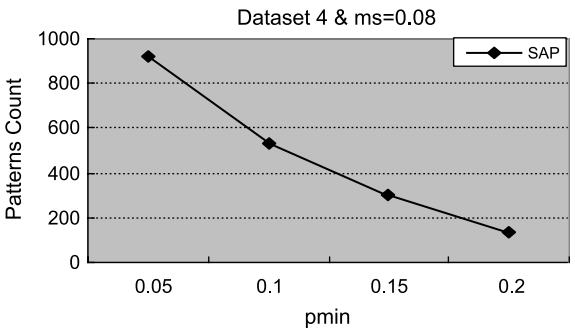

Fig. 4. The count of actionable sequential patterns.

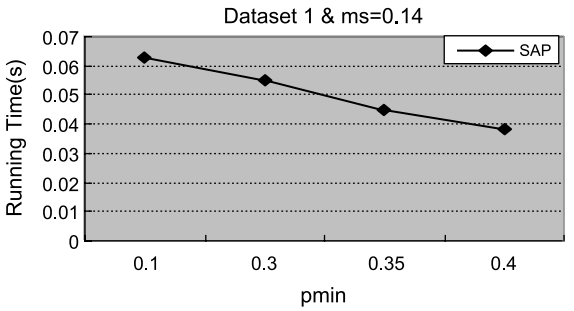

Fig. 5. Running time (s).

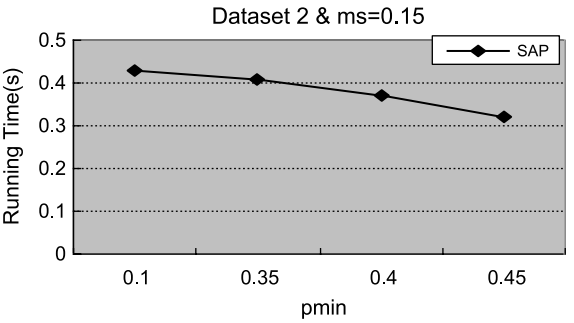

Fig. 6. Running time(s).

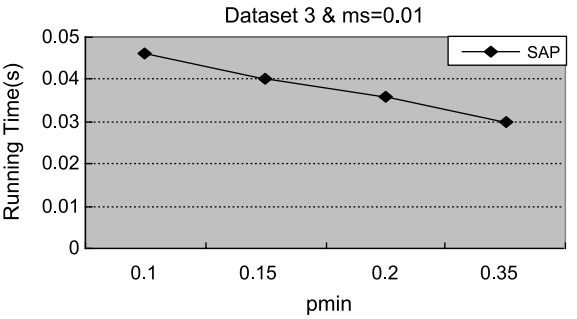

Fig. 7. Running time (s).

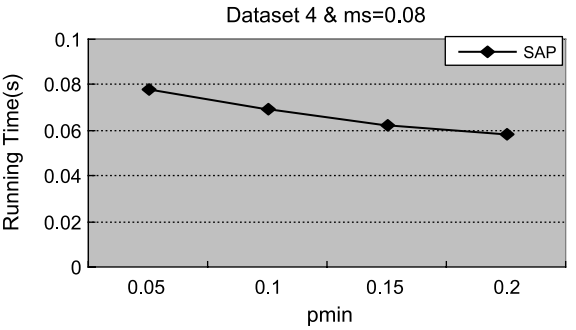

Fig. 8. Running time (s).

shown in Fig. 1, we set ms = 0.14. The experimental data are synthetic data with an increment of ρmin, the number of actionable sequential patterns downward trend is flat, and the running time downward trend is flat. With the same synthetic data, the experimental results in Figs. 1 and 5 are identical to results shown in Figs. 2 and 6. In Fig. 3, ms = 0.01, which is much lower than the value in the synthetic data, is closer to the real-world number. Therefore, the experimental result is close to the real-world result, which proves that the proposed method can be used in businesses applications. As shown in Figs. 3 and 4, with the increase of ρmin, the downward trend in the number of actionable sequential patterns is not a steady decline, but demonstrates some variation. As shown in Figs. 7 and 8, the running time trend is the same as in Figs. 5 and 6, which proves that our proposed method has a stable running time.

Because our method was conducted after the improvement of Wu's method and Wu's method cannot be used for processing sequence, our proposed method cannot be directly compared to Wu's method. However, our data are all sequence data. Further, There is no method can be used to compare.

These experimental results indicate that our proposed method demonstrates a proven stable running result and running time. The problem of selecting the actionable positive and negative sequential patterns is solved through our method. By setting the parameter close to the real-world parameter, SAP determination can efficiently deal with both synthetic and real-world databases.

5. Conclusions and future work

It is increasingly recognized that NSP can play an irreplaceable role in understanding and addressing many business applications. However, some urgent problems occur after mining NSP. These urgent problems include: 1) a selection problem; 2) efficiency problem; 3) candidate problem; and other concerns. Among these problems, the selection problem is the most urgent because the PSP being mined before considering NSP may mislead decisions, and it is much more difficult to select actionable patterns after mining NSP due to the huge number of NSPs. This paper proposes an improvement to Wu's pruning method to

fit for selecting ASP. A novel method is proposed to efficiently use SAP to select ASP. Experimental results indicate that SAP is very efficient.

Future work will find solutions to other urgent problems.

Acknowledgments

This work was supported partly by National Natural Science Foundation of China (71271125, 61173061 and 71201120), Natural Science Foundation of Shandong Province, China (ZR2011FM028) and Scientific Research Development Plan Project of Shandong Provincial Education Department (J12LN10).

References

[1] A. Tzacheva and Z. Ras, Action rules mining, *International Journal of Intelligent Systems* **20** (2005), 719–736.

[2] B. Liu, W. Hsu, S. Chen and Y.M. Ma, Analyzing subjective interestingness of association rules, *IEEE Intelligent Systems* **15**(5) (2000), 47–55.

[3] B. Liu, W. Hsu and Y.M. Ma, Identifying non-actionable association rules, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001*, San Francisco, CA, USA, pp. 329–334.

[4] E.R. Omiecinski, Alternative interest measures for mining associations, *IEEE Trans Knowledge and Data Eng* **15**(1) (2003), 57–69.

[5] G. Adomavicius and A. Tuzhilin, *Discovery of actionable patterns in databases: The action hierarchy approach*, KDD, 1997, California, USA, pp. 111–114.

[6] K. Kavitha and E. Ramaraj, Efficient transaction reduction in actionable pattern mining for high voluminous datasets based on bitmap and class labels, *International Journal on Computer Science and Engineering* **5**(7) (2013), 664–671.

[7] K. Wang, Y. Jiang and A. Tuzhilin, Mining actionable patterns by role models, *Proceedings of the International Conference on Data Engineering* **2006** (2006), 16–16.

[8] L.B. Cao, C.Q. Zhang, et al., Domain-driven actionable knowledge discovery, *IEEE Intelligent Systems* **22**(4) (2007), 78–88.

[9] L.B. Cao, Y.C. Zhao, et al., Flexible frameworks for actionable knowledge discovery, *IEEE Transactions on Knowledge and Data Engineering* **22**(9) (2010), 1299–1312.

[10] L.B. Cao, Actionable knowledge discovery and delivery, *WIREs Data Mining and Knowledge Discovery* **2**(2) (2012), 149–163.

[11] M.L. Antonie and O.R. Zaiane, Mining positive and negative association rules: An approach for confined rules, *Knowledge Discovery in Databases: PKDD* **3202** (2004), 27–38.

[12] N.P. Lin, H.J. Chen and W.H. Hao, *Minin negative sequential patterns*, WSEAS, Hangzhou, China, 2007, pp. 654–658.

[13] N.P. Lin, H.J. Chen and W.H. Hao, Minin negative fuzzy sequential patterns, *Proceedings of the 7th WSEAS International Conference on Simulation, Modeling and Optimization*, Beijing, China, 2007, pp. 52–57.

[14] P. Kanikar and D.K. Shah, Extracting actionable association rules from multiple datasets, *International Journal of Engineering Research and Applications* **2**(3) (2012), 1295–1300.

[15] P.N. Tan, V. Kumar and J. Srivastava, Selecting the right interestingness measure for association patterns, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alta, Canada, 2002, pp. 32–41.

[16] Q. Yang, J. Yin, C. Ling and R. Pan, Extracting actionable knowledge from decision trees, *IEEE Trans Knowledge and Data Eng* **19**(1) (2007), 43–55.

[17] R.J. Hilderman and H.J. Hamilton, Applying objective interestingness measures in data mining systems, *Proceedings of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'00)*, France, 2000, pp. 432–439.

[18] R. Agrawal and R. Srikant, *Minin sequential patterns*, ICDE, Taipei, 1995, pp. 3–14.

[19] S.C. Hsueh, M.Y. Lin and C.L. Chen, Mining negative sequential patterns for e-commerce recommendations, *Asia-Pacific Services Computing Conference*, Yilan, Taiwan, 2008, pp. 1213–1218.

[20] V.K. Khare and V. Rastogi, Mining positive and negative sequential pattern in incremental transaction databases, *International Journal of Computer Applications* **71**(1) (2013), 18–22.

[21] W.M. Ouyang and Q.H. Huang, Mining negative sequential patterns in transaction databases, *Machine Learning and Cybernetics* **2** (2007), 830–834.

[22] X.J. Dong, Mining interesting infrequent and frequent itemsets based on minimum correlation strength, *Lecture Notes in Computer Science* **7002** (2011), 437–443.

[23] X.D. Wu, C.Q. Zhang and S.C. Zhang, Efficient mining of both positive and negative association rules, *ACM Trans Inf Syst* **22**(3) (2004), 381–405.

[24] X.J. Dong, Z.G. Zheng, L.B. Cao, Y.C. Zhao, et al., E-NSP: Efficient negative sequential pattern mining based on identified positive patterns without database rescanning, *International Conference on Information and Knowledge Management*, Glasgow, UK, 2011, pp. 825–830.

[25] X.J. Dong, Z.D. Niu, X.L. Shi, X.D. Zhang and D.H. Zhu, Mining both positive and negative association rules from frequent and infrequent itemsets, *Advanced Data Mining and Applications* **4632** (2007), 122–133.

[26] Z.G. Zheng, Y.C. Zhao, Z. Zuo and L.B. Cao, Negative-gsp: An efficient method for mining negative sequential patterns, *Data Mining and Analytics* **101** (2009), 63–67.

[27] Z.G. Zheng, Y.C. Zhao, Z. Zuo and L. Cao, An efficient ga-based algorithm for mining negative sequential patterns, *PAKDD* **6118** (2010), pp. 262–273.