Filomat 35:5 (2018), 1765–1776
https://doi.org/10.2298/FIL1805765Q

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Mining Negative Sequential Patterns from Frequent and Infrequent Sequences Based on Multiple Level Minimum Supports

Ping Qiu[a], Xiaoqi Jiang[a], Feng Hao[a], Tiantian Xu[a], Xiangjun Dong[a]

[a] *School of Information, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China*

Abstract. Negative sequential patterns (NSP) are critical and sometimes much more informative than positive sequential patterns (PSP) in many intelligent systems and applications. However, the existing NSP algorithms do not allow negative items being contained in an element except the NegI-NSP algorithm, which can obtain many meaningful sequences with negative items in an element. NegI-NSP, however, hasn't considered the following problems: (1) it uses a single minimum support to all size sequences, which is unfair to a long size sequence; (2) it only mines NSP from PSP, not from infrequent positive sequences (IPS), which also contain many useful NSP. So we propose an efficient algorithm, named MLMS-NSP, to mine NSP based on multiple level minimum supports (MLMS) from PSP and IPS. Firstly, MLMS scheme is proposed by assigning different minimum supports to sequences with different sizes. Secondly, IPS are constrained by combining MLMS, and then the NSP is obtained from these IPS. Finally, experimental results show that the MLMS-NSP algorithm can effectively mine NSP from IPS, and the time efficiency is higher than using single minimum support.

1. Introduction

Behavioral research concerns all aspects of our lives and has attracted more and more attention. How to understand behaviors, especially the non-occurring behaviors (NOB) is very important in behavioral research [1][2]. Mining negative sequential patterns (NSP) is one of few methods that can well understand NOB [3]. Different from only considering occurring (positive) behaviors in positive sequential patterns (PSP)[4], NSP take into account both non-occurring (negative) and occurring behaviors. NSP play an irreplaceable role in many intelligent systems and applications. For example, $s =< xy\neg zF >$ is a NSP, where x, y and z stand for drug codes, and F stands for disease status. The NSP s shows that patients who usually take drugs x and y but NOT z are likely to have disease status F. Such situation cannot be expressed by using PSP alone.

However, most of the existing NSP algorithms do not allow negative items to be contained in elements [5][6][7][8][9][10][11][12][13], which makes many meaningful sequences impossible to be obtained. For

2010 *Mathematics Subject Classification*. Primary 68T10
Keywords. negative sequential patterns, infrequent positive sequences,multiple level minimum supports
Received: 19 October 2017; Accepted: 7 December 2017
Communicated by Hari M. Srivastava
Corresponding authors are Tiantian Xu and Xiangjun Dong
Research supported by National Natural Science Foundation of China (71271125, 61502260), Shandong Natural Science Foundation, China (ZR2018MF011)
Email addresses: qiupcl@163.com (Ping Qiu), 704595335@qq.com (Xiaoqi Jiang), hf_mails@163.com (Feng Hao), xtt-ok@163.com (Tiantian Xu), d-xj@163.com (Xiangjun Dong)

example, for customers' purchase sequences database in a supermarket, a customer's purchase behavior is an element, and the item in the element is a commodity. In real life, we cannot ignore this purchase because the customer has not bought certain goods at one time. NegI-NSP algorithm [14] was proposed for the problem. Firstly, it formally introduced two loose constraints as much as possible. Secondly, it proposed negative containment definition based on items. Finally, it proved equations for calculating the supports of negative sequential candidates (NSC), so that the NSCs supports can be calculated only by searching the information of NSCs corresponding PSP, which improves the efficiency of algorithm time effectively.

Although NegI-NSP can efficiently mine NSP based on items, it cannot consider the following problems.

(1) It uses a single minimum support to all size sequences, which is unfair to a long size sequence. For a sequence that contains k elements ($k = 1, 2, , m$), i.e., the size of the sequence is k, the bigger the k is, the smaller its support is[15]. So using a single minimum support (ms) is unfair to the long size sequence. If the support was too high, a small number of long frequent sequences would be discovered; if the support was too low, a large number of short frequent sequences would be discovered, which would increase the difficulties for users to choose actionable sequential patterns [16]. To solve the problem, Apriori_MLMS [15] used the multiple level minimum supports (MLMS) to constrain infrequent itemsets and frequent itemsets by giving different ms to itemsets with different lengths; E-msNSP [17] and CPNFSP [18] used multiple minimum supports (MMS) to mine NSP by setting different ms to different items. However, these methods are either not used for mining sequential patterns, or do not consider the influence of the sequences size. So we use MLMS scheme, i.e., assign different minimum supports to sequences with different sizes to mine NSP, which is different form the existing works.

(2) It only mines NSP from PSP, not from infrequent positive sequences (IPS), which also contain many useful NSP. Just like many useful negative association rules or negative frequent itemsets can be mined from infrequent itemsets (inFIS) [17][19][20][21][22], there are many useful NSP in IPS. However, how to discover IPS is still an open problem [23]. E-NSPFI [13] is the only existing algorithm to mine NSP from IPS. But its constraint is too strict to IPS because it requires the supports of all (k-1)-size subsequences of ips are not less than minimum support threshold. For example, given ms=2, a dataset is as follows:{10 :< $abcad$ > ;20 :< $acad$ >;30 :< bcd >;40 :< acb >;50 :< $adcd$ >}. According to the existing PSP mining algorithms, s_1 =< abc > and s_2 =< $abcd$ > are infrequent sequences because the supports of s_1 and s_2 are both 1, denoted as $sup(s_1) = 1$ and $sup(s_2) = 1$. The sequence s_1 is the 3-size subsequence of s_2. So s_2 does not satisfy the infrequent constraint of e-NSPFI. But s_2 can also generate NSP, such as < $a\neg bc\neg d$ >. In fact, a large number of IPS whose (k-1)-size subsequences are infrequent contains useful NSP.

To solve the two problems, we propose an efficient algorithm, named MLMS-NSP, to mine NSP from PSP and IPS based on MLMS. We summarize the significant contributions of this paper as follows:

Firstly, MLMS scheme is proposed by assigning different ms to sequences with different sizes.

Secondly, IPS is constrained by combining MLMS, and then the NSP is obtained from these IPS.

Finally, the experimental results show that the MLMS-NSP algorithm can effectively mine NSP from IPS, and the time efficiency is higher than using single minimum support.

The remainder of the paper is organized as follows. Section 2 discusses the related work. Section 3 is preliminary. Section 4 proposes MLMS-NSP algorithm. Section 5 is experimental results. Conclusions and future work are discussed in Section 6.

2. Related work

This section consists of two main parts. Firstly, we summarize the status of NSP mining. Secondly, we discuss the status of mining useful information from infrequent patterns.

2.1. The Status of NSP Mining

NegGSP [24] generates NSC by seed sets and calculates the NSC's supports by re-scanning database. This is very time consuming [25]. Furthermore, it generates NSP by comparing the supports of NSC with single minimum support. PNSP is another algorithm to mine NSP in the form of $< (abc)\neg(de)(ijk) >$ [5]. It generates NSC by joining iteratively positive and negative itemsets. NSC's supports are also calculated by re-scanning database, and then NSP are generated by single minimum support. GA algorithm still uses the single minimum support, and the difference is that it avoids generating NSC [12]. The method in [26]only identifies three forms of NSP, i.e., $(\neg A, B)$, $(A, \neg B)$ and $(\neg A, \neg B)$ and requires $A \cap B = \emptyset$. Although this requirement is common in association rules mining, it is very strict in NSP mining. NSPM only deals with the last element in the NSP [27].

E-NSP is the most time efficient algorithm for mining NSP [3]. It generates NSC by using a conversion strategy and calculates the supports of NSC only by equations, thus avoiding re-scanning database. This effectively improves the efficiency of e-NSP. f-NSP use bitset structure to effectively improve the time efficiency of e-NSP algorithm[28] and e-RNSP mine repetitive negative sequence patterns[29]. SAPNSP [30], SAP[31], SAPSD [6] and SAPBN [6] first mine NSP by e-NSP algorithm, and then select actionable NSP by different methods. E-msNSP [16] and CPNFSP [18] use multiple minimum supports to mine NSP. E-msNSP uses the same idea of e-NSP to aviod re-scanning database and CPNFSP only identified three forms of NSP. Furthermore, they use the actual frequencies of a single item sequence in the dataset as the basis for minimum item support assignments.HUNSPM can mine high utility NSP[32]. NegI-NSP [14] is the only existing algorithm to mine NSP based on items. In NegI-NSP, the smallest negative unit of NSC is an item. NegI-NSP not only proposes the definition of negative containment based on items, but also proves the formulas for the supports of NSC to avoid re-scanning database. This paper uses the definition of negative containment and the formulas to calculate the NSC's supports.

2.2. The Status of Mining Useful Information from Infrequent Patterns

Apriori_MLMS [15] used the MLMS to mine inFIS and frequent itemsets (FIS). It assigned different ms to itemsets with different lengths. Let *ms(k)* be the *ms* of *k*-itemsets ($k = 1, 2, , n$), $ms(1) \geq ms(2) \geq, \ldots, \geq ms(n) \geq ms > 0$. For any *k*-itemset X, if $sup(X) \geq ms(k)$, then X is a FIS; and if $ms(k) > sup(X) \geq ms$, then X is an inFIS. Apriori_IMLMS [23] algorithm is an extended version of Apriori_XMMS. It used MLMS model and interesting metric to select inFIS and FIS. Apriori_XMMS [33] extended the MMS model to adapt to mining inFIS by adding a constraint to inFIS. The literature [34] used correlation coefficient instead of interest to improve the performance of IMLMS model.

PNAR_MLMS [35] is a corresponding algorithm to mine positive and negative association rules from inFIS and FIS discovered by MLMS model. The algorithm in literature [36] proposed the concept of 2-level supports model to discover inFIS and FIS. 2-level supports model uses two level supports ms_FIS and ms_inFIS ($ms\_FIS \geq ms\_inFIS > 0$) to constrain the inFIS and FIS respectively. For any itemset A, if $sup(A) \geq ms\_FIS$, then A is a FIS; and if $ms\_FIS > sup(A) \geq ms\_inFIS$, then A is an inFIS. The literature [37] proposed 2-level XMMS model, which is based on MMS model, to improve the efficiency of 2-level supports model. E-NSPFI [13] uses the idea of e-NSP to avoid re-scanning database for mining NSP from IPS. But it requires that any subsequence of IPS should be frequent. This is very strict for IPS.

3. Preliminary

3.1. Positive Sequential Patterns-PSP

Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items. An itemset is a subset of I. A sequence is an ordered list of itemsets. An itemset does not allow repeated items, and sequences allow repeated items in different elements. A

sequence s is denoted by $< e_1e_2...e_l >$, where e_j is an element of s, $e_j \subseteq I(1 \leq j \leq l)$. The element e_j can be denoted as $(i_1i_2...i_m)$, where i_k is an item, $i_k \in I$ $(1 \leq k \leq m)$. For simplicity, the bracket is omitted if an element only contains one item, i.e., element (i) is coded i.

The total number of elements in s is the size of sequence s, denoted as $size(s)$. When $size(s) = l$, s is called a l-size sequence. The total number of items in s is the length of sequence s, denoted as $length(s)$. When $length(s) = m$, s is a m-length sequence. For example, a sequence $s =< (ab)(cd)(ef) >$ is composed of three elements (ab), (cd) and (ef). Therefore, s is a 3-size and 6-length sequence, i.e., $size(s) = 3$ and $length(s) = 6$.

Sequence $s_\alpha =< \alpha_1\alpha_2...\alpha_n >$ is a sub-sequence of sequence $s_\beta =< \beta_1\beta_2...\beta_m >$ and s_β is a super-sequence of s_α, if $1 \leq j_1 < j_2 < \ldots < j_n \leq m$, $\alpha_1 \subseteq \beta_{j1}, \alpha_2 \subseteq \beta_{j2}, \ldots, \alpha_n \subseteq \beta_{jn}$, denoted as $s_\alpha \subseteq s_\beta$. We also say that s_β contains s_α. For example, the sequence $< bf >$ is subsequences of $< (ab)(cd)(ef) >$. The element b of $< bf >$ is contained the first element (ab) of $< (ab)(cd)(ef) >$,i.e., $j_1 = 1$; the element f of $< bf >$ is contained the third element (ef) of $< (ab)(cd)(ef) >$,i.e., $j_2 = 3$.

A *sequence database* D is a set of tuples $< sid, ds >$, where sid is the *sequence_id* and ds is the *data sequence*. The number of tuples in D is denoted as |D|. The set of tuples containing sequence s is denoted as $\{< s >\}$. The support of s, denoted as $sup(s)$, is the number of $\{< s >\}$, i.e.,$sup(s) = |\{< s >\}| = |\{< sid, ds >, < sid, ds > \in D \wedge (s \subseteq ds)\}|$.

3.2. Negative Sequential Patterns-NSP

This section introduces the negative containment definition of NegI-NSP algorithm. The definition will be used in this paper. Let's introduce an advance definition.

Definition 1. *Positive Partner*. The positive partner of a negative element$(\neg ab)$ is (ab), which is denoted as $p(\neg ab)$, i.e., $p(\neg ab) = (ab)$; the positive partner of positive element (ab) is (ab) itself, i.e., $p(ab) = (ab)$. The positive partner of a negative sequence $ns =< s_1...s_k >$ can be obtained by converting all negative elements to their positive partners, which is denoted as $p(ns)$, i.e., $p(ns) = \{< s'_1...s'_k > | s'_i = p(s_i), s_i \in ns\}$. For instance, $p(< (\neg(abc)\neg c(\neg de) >) =< (bc)e >$.

To determine whether the ds contains ns, we first need to determine whether the ordered sequence of all the positive items in ns $(MPS(ns))$ is a subsequence of ds. Only ds contains $MPS(ns)$, ds may contains ns. Secondly, we need to determine whether the ordered sequence of any negative item's partner and all positive items in ns is a subsequence of ds. The ordered sequence of any negative item's partner and all positive items in ns is called the $1 - neg - lengthMaximumSub - sequence$ denoted as $1 - neglMS_{ns}$. For any $1 - neglMS_{ns}$, only ds does not contain $1 - neglMS_{ns}$, ds contains ns. Hence we propose the Definition 3 and Definition 4.

Definition 2. *Maximum Positive Subsequence*. The sequence of all positive items in ns is called *the Maximum Positive Subsequence* devoted as $MPS(ns)$.

Definition 3. *1-neg-size Maximum Sub element*. For a negative element s_i, its sub element includes all positive items and one negative item e $(\forall e \in s_i)$ is called a *1-neg-size maximum sub element*, denoted as $1 - negMPSE$.

Definition 4. *1-neg-length Maximum Subsequence*. For a negative sequence ns, its subsequence that includes *MPS(ns)* and $1 - negMPSE$ is called a $1 - neg - lengthmaximumsubsequence$, which is denoted as $1 - neglMS_{ns}$. The subsequence set that includes all 1-neg-length maximum subsequences of ns is called $1 - neg - lengthmaximumsubsequenceset$, which is denoted as $1 - neglMSS_{ns}$.

Definition 5. *Negative Containment Definition*. Let $ds =< d_1d_2 \ldots d_t >$ be a data sequence, $ns =< s_1s_2 \ldots s_m >$ be an m-size and n-neg-size negative sequence (1) if $m > 2t + 1$, then ds does not contain ns; (2) if $m \geq 1$ and n=1, then ds contains ns when $\forall p(1 - neglMS) \nsubseteq ds$ and $MPS(ns) \subseteq ds$; (3) otherwise, ds contains ns if, $\forall 1 - neglMS_i \in 1 - neglMSS_{ns}$, $p(1 - neglMS_i) \nsubseteq ds$ and $MPS(ns) \subseteq ds(1 < i \leq n)$.

4. MLMS-NSP Algorithm

This section consists of three parts. First, the definition of multiple level minimum supports (MLMS), including the scope of infrequent sequential patterns (IPS) and frequent sequential patterns, is proposed. Second, the steps of MLMS-NSP algorithm are given. Final, the pseudo code is given.

4.1. Multiple Level Minimum Supports-MLMS

Constraint 1 (*1-length-neg element format constraint*). NSC only does not allow continuous 1-length negative elements.

For instance, a sequence $< (\neg xy)\neg z >$ satisfies the constraint, but a sequence $< \neg x\neg y >$ does not. This is because we are unable to determine the order in which the elements, i.e., $\neg x$ and $\neg y$, occur.

Definition 6. *Multiple Level Minimum Supports Definition*. Let $ms(k)(k = 0, 1, 2, \ldots, m)$ indicates the minimum support of a k-size sequence, $ms(1) \geq ms(2) \geq, \ldots, \geq ms(m) \geq ms(0) > 0$,for k-size sequence s,

(1) if $sup(s) \geq ms(k)$, then s is a frequent sequential pattern;

(2) if $sup(s) < ms(k)$ and $sup(s) \geq ms(0)$, the s is an infrequent sequential pattern.

Both $ms(0)$ and $ms(k)$ are specified by users or researchers. Among them, $ms(0)$ is the minimum support threshold for mining IPS.

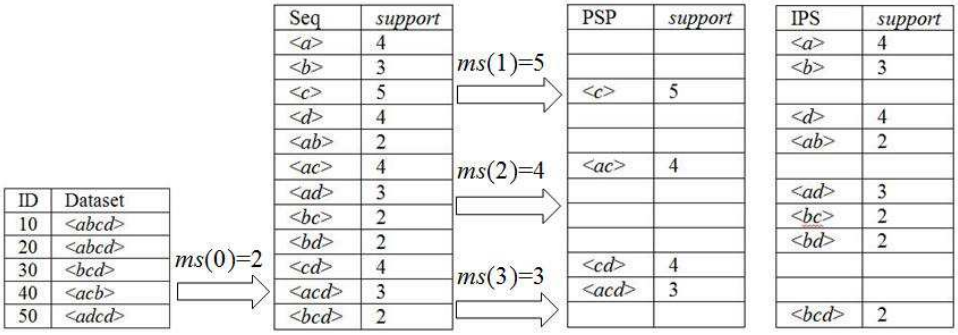

Figure 1: The PSP and IPS are mined by using MLMS.

4.2. Steps of MLMS-NSP Algorithm

Step1: improve the GSP algorithm to mine PSP and IPS based on MLMS.

First, we use different *ms* to mine PSP and output the number of PSP of different sizes. In particular, the supports of sequences are obtained by re-scanning the database. Second, we compare the results of each ms to set MLMS. How to set the MLMS is introduced in section 5.2. Finally, we use the MLMS and $ms(0)$ to select PSP and IPS.

Step2: use the method in NegI-NSP algorithm to generate NSC from the above PSP and IPS as follows. For a k-size PSP, its NSC are generated by changing any m elements to their negative ones, $1 \leq m \leq k$, for each element that contains h items, change any w, $1 \leq w < h$. In particular, if there are continuous single elements $h = 1$, they cannot be changed continuously.

For instance, the NSC based on $< (xy)zx >$ include:

m=1, $< (\neg xy)zx >, < (x\neg y)zx >, < (xy)\neg zx >, < (xy)z\neg x >$;

m=2, $< (\neg xy)\neg zx >, < (x\neg y)\neg zx >, < (\neg xy)z\neg x >, < (x\neg y)z\neg x >$.

Step3: use the formulas in NegI-NSP algorithm to calculate the NSC's supports.

Given an m-size and n-neg-size negative sequence ns, among the n negative elements, for $\forall 1 - neglMS_i \in 1 - negsMSS_{ns}(1 \leq i \leq n)$, the support of ns is:

$$sup(ns) = |\{ns\}| = |\{(MPS(ns)\} - \cup_{i=1}^{n}\{p(1 - neglMS_i)\}| \tag{1}$$

Because $\cup_{i=1}^{n}\{p(1 - neglMS_i)\} \subseteq \{MPS(ns)\}$, equation (1) can be converted to:

$$sup(ns) = |\{(MPS(ns)\}| - |\cup_{i=1}^{n}\{p(1 - neglMS_i)\}|sup(MPS(ns)) - |\cup_{i=1}^{n}\{p(1 - neglMS_i)\}| \tag{2}$$

In particular, for negative sequence $< \neg e >$,

$$sup(< \neg e >) = |D| - sup(< e >) \tag{3}$$

Step4: compare the NSC's supports with MLMS to generate NSP.

Table 1: The Pseudo Code of MLMS-NSP Algorithm

Algorithm: MLMS-NSP Algorithm.

Input: Sequence dataset D and Parameter $ms(k)$;

Output:IPS, PSP and NSP;

(1) C=imporGSP()_Re-scanning(D);
(2) For(c: C){
(3) If ($c.support \geq ms(0)$){
(4) $IPS.add(c)$;
(5) $L.add(c)$;
(6) If($c.support \geq ms(k)$){
(7) $PSP.add(c)$;
(8) }}
(9) while ($L.size() > 0$){
(10) $C = genCandidate(L)$;
(11) For($int i = 0; i < C.size(); i++$){
(12) $c = C.get(i)$; (13) If ($sup(c) \geq ms(0)$){
(14) $IPS.add(c)$;
(15) $L.add(c)$;}
(16) If($c.support \geq ms(k)$){
(17) $PSP.add(c)$; (18) }}}}
(19) For ($eachspinPSP \cup IPS$){
(20) $NSC = NegI - NSP\_Candidate\_Generation(psp)$;
(21) For($nsc : NSC$){
(22) Calculate the support of nsc by using the formulas(1),(2) and (3);
(23) If ($sup(nsc).support \geq ms(k)$){
(24) $NSP.add(nsc)$;
(25) }}
(26) Return $PSP, IPSandNSP$;

4.3. Pseudo Code of MLMS-NSP Algorithm

Line (1) to (18) is the improving GSP algorithm to mine PSP and IPS;
Line (1) to (7) generate the 1 size PSP and IPS;
line (9) to (18) generate the size of PSP and IPS larger than 1;
Line (18) to (26) generate the NSP from the PSP and IPS;
Line (20) generates NSC by step2;
Line (22) calculates the supports of NSC by step3;
Line (23) to line (25) compare the supports of NSC with $ms(k)$ to generate NSP;
Line (26) returns the results and ends the program.

5. Experiment and Results

We conduct experiments to verify the performance of MLMS-NSP, including the effect of IPS on NSP and the effect of MLMS on NSP. For the first experiment, we change *ms(0)* to mine PSP and IPS based on the same MLMS by applying the improved GSP algorithm, and then mine NSP from these PSP and IPS. For the second experiment, we compare the efficiency of MLMS-NSP with NegI-NSP. All experiments are

Table 2: Summary of datasets

Dataset	sequence Numbers	distinct item Numbers	file size
DS1	5,269	Around 4K	5.1M
DS2	10K	100	12.8M
DS3	1K	100	1.9M
DS4	100	100	52.6M

performed on Windows 7 PC with 16GB memory, Inter Core i5 2.5GHz CPU, all the programs are written in Java. In this section, all supports (*ms* and MLMS) are calculated in terms of the percentage of the frequency $|< s >|$ of a pattern s compared to $|D|$.

Table 3: PSP and IPS are generated by MLMS on DS1

DS1	PSP	k=1	k=2	k=3	k=4	k=5	k=6	k=7
ms(*)=0.09	PSP	51	243	383	165	50	3	2
ms(*)=0.1	PSP	47	209	270	121	26	3	2
ms(*)=0.11	PSP	44	178	180	90	6	2	0
ms(*)=0.12	PSP	40	156	122	68	3	0	0
ms(*)=0.13	PSP	37	127	89	50	1	0	0
ms(*)=0.14	PSP	36	105	63	38	0	0	0
ms(*)=0.15	PSP	31	94	46	24	0	0	0
ms(*)=0.16	PSP	28	73	42	16	0	0	0
ms(*)=0.17	PSP	27	60	36	10	0	0	0
ms(1)=0.17,. . .,	PSP	27	73	46	38			
ms(4)=0.14,ms(0)=0.09	IPS	24	170	337	127			
ms(1)=0.17,. . .,	PSP	27	73	46	38			
ms(4)=0.14,ms(0)=0.13	IPS	10	54	43	12			

5.1. Datasets

We use one real-life and three synthetic datasets for the experiments. The synthetic datasets are generated by IBM data generator.

Dataset 1 (DS1) is a dataset of health insurance claim sequences. The dataset contains 5269 sequences. The average number of elements in per sequence is 21. The maximum number of elements in a sequence is 144, and the minimum number is 1.

Dataset 2 (DS2), C6_T16_S8_I10_DB10k_N100.

Dataset 3 (DS3), C13_T8_S14_I8_DB1k_N100.

Dataset 4 (DS4), C6_T6_S8_I6_DB100k_N100.

5.2. Assign MLMS

Tables 2, 3, 4 and 5 represent the number of PSP that can be mined from DS1 to DS4 by using single minimum support, where *ms(*)* represents any minimum support threshold and k is the size of PSP. From table 2, when $ms(*) \geq 0.14$ and $k \geq 5$, the number of PSP is 0. That is when *ms(0)*=0.13 and k=5, the number

Table 4: PSP and IPS are generated by MLMS on DS2

DS1	PSP	k=1	k=2	k=3	k=4	k=5	k=6	k=7
ms(*)=0.19	PSP	820	9953	14408	2712	19	0	0
ms(*)=0.2	PSP	709	8043	10704	1752	7	0	0
ms(*)=0.21	PSP	645	6560	7949	1152	3	0	0
ms(*)=0.22	PSP	570	5379	5988	747	0	0	0
ms(*)=0.23	PSP	502	4519	4576	503	0	0	0
ms(*)=0.25	PSP	397	3076	2675	217	0	0	0
ms(*)=0.26	PSP	354	2586	2054	135	0	0	0
ms(*)=0.27	PSP	320	2167	1574	94	0	0	0
ms(*)=0.28	PSP	291	1868	1243	58	0	0	0
ms(1)=0.28,. . .,	PSP	291	2167	2054	217			
ms(4)=0.25,ms(0)=0.19	IPS	529	7786	12354	2495			
ms(1)=0.28,. . .,	PSP	291	2167	2054	217			
ms(4)=0.25,ms(0)=0.23	IPS	211	2352	3422	286			

Table 5: PSP and IPS are generated by MLMS on DS3

DS1	PSP	k=1	k=2	k=3	k=4	k=5	k=6	k=7
ms(*)=0.2	PSP	609	14855	83032	131104	63232	8134	98
ms(*)=0.21	PSP	550	12641	66586	97577	43364	4870	41
ms(*)=0.22	PSP	511	11025	55269	76183	31534	3158	76
ms(*)=0.23	PSP	467	9471	44691	57635	21774	1905	5
ms(*)=0.24	PSP	422	8373	37430	45283	16064	1251	3
ms(*)=0.25	PSP	388	7217	30408	34616	11276	738	0
ms(*)=0.26	PSP	357	6398	25605	27474	8301	477	0
ms(*)=0.27	PSP	320	5621	21063	21076	5941	276	0
ms(*)=0.28	PSP	303	5010	17804	16872	4380	167	0
ms(*)=0.29	PSP	340	7307	23840	18408	3024	14	0
ms(*)=0.3	PSP	257	3980	12613	10472	2317	68	0
ms(1)=0.30,. . .,	PSP	257	7307	17804	21076	8301	738	
ms(6)=0.25,ms(0)=0.2	IPS	352	7548	65228	110028	54931	7396	
ms(1)=0.30,. . .,	PSP	257	7307	17804	21076	8301	738	
ms(6)=0.25,ms(0)=0.24	IPS	165	1066	19626	24207	7763	513	

Table 6: PSP and IPS are generated by MLMS on DS4

DS1	PSP	k=1	k=2	k=3	k=4	k=5	k=6	k=7
ms(*)=0.07	PSP	332	1635	778	16	0	0	0
ms(*)=0.08	PSP	269	1160	446	31	0	0	0
ms(*)=0.09	PSP	221	872	275	1	0	0	0
ms(*)=0.1	PSP	180	679	181	0	0	0	0
ms(*)=0.11	PSP	148	524	118	0	0	0	0
ms(*)=0.12	PSP	125	414	72	0	0	0	0
ms(*)=0.13	PSP	113	321	48	0	0	0	0
ms(*)=0.14	PSP	104	250	33	0	0	0	0
ms(1)=0.14,ms(2)=0.13,	PSP	104	321	72				
ms(3)=0.12,ms(0)=0.07	IPS	228	1314	706				
ms(1)=0.14,ms(2)=0.13,	PSP	104	321	72				
ms(3)=0.12,ms(0)=0.11	IPS	44	203	46				

of 5-size IPS is 0. This is meaningless. Therefore, when testing the effect of IPS on NSP mining, we do not consider the sequences (*size*>4). Furthermore, in order for easy set MLMS, for a few sequences with great size, we set the same minimum support, like using single minimum support to mine NSP. Therefore, when testing the effect of MLMS on NSP mining, we do not consider the sequences (*size*>4) either. In summary, for DS1, we can set the minimum support of the 1-size PSP is 0.17, denoted as *ms*(1)=0.17, *ms*(2)=0.16, *ms*(3)=0.15 and *ms*(4)=0.14, the number of PSP corresponding to the shadowed portions are (27 + 73 + 46 + 38 =)184, and the number of IPS is (24 + 170 + 337 + 127 =)658 when *ms*(0)=0.09 and the number of IPS is (10 + 54 + 43 + 12 =)119 when *ms*(0)=0.13. This is the MLMS for DS1.

Similar to set MLMS for DS1, we can set MLMS from DS2 to DS4 by analyzing tables 3 to 5. The shadowed part represents the number of PSP. For DS2, we set *ms*(1)=0.28, *ms*(2)=0.27, *ms*(3)=0.26 and *ms*(4)=0.25; For DS3, we set *ms*(1)=0.30, *ms*(2)=0.29, *ms*(3)=0.28, *ms*(4)=0.27, *ms*(5)=0.26 and *ms*(6)=0.25; For DS4, we set *ms*(1)=0.14, *ms*(2)=0.13 and *ms*(3)=0.12.

5.3. Experimental Results

In section 4.2, we can obtain the number of PSP with different MLMS from DS1 to DS4. All the experiments are based on section 4.2.

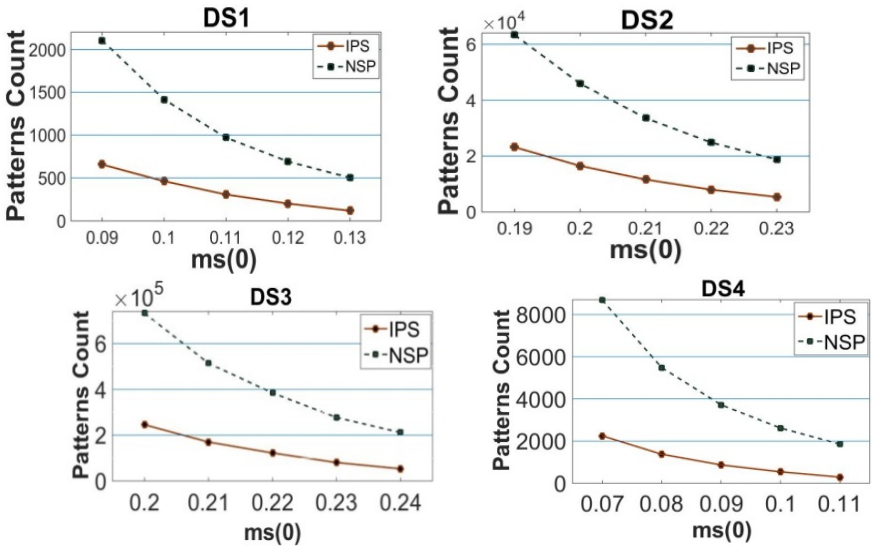

Figure 2: The count of IPS and NSP.

(1)The Effect of IPS on NSP Mining

From Figure 2, we can see that the number of IPS and NSP both decrease, and the number of NSP decreases faster than IPS as *ms*(0) increase. For each dataset, under the same *ms*(0), the number of NSP is consistently more than IPS. This is because with *ms*(0) increasing, the number of long PSP are generated lower than before and long PSP can generate more NSC according to section 3.2.

From figure 3, for DS1, DS2 and DS4, the running time of NSP is longer than IPS and the running time of NSP decreases faster than IPS as *ms*(0) continues to increase. This is proportional to the number of NSP and IPS. For DS3, the running time of NSP is shorter than IPS because the file size of DS3 is around 1.9M and contains only 1000 the data sequences. This is smaller than other datasets. Therefore, when mining IPS, the number of re-scanning data sequences is greatly reduced compared to other datasets.

(2)The Effect of MLMS on NSP Mining

From Figure 4 and 5, ms(1) represents the minimum support of a sequence with 1 size in MLMS-NSP, and also represents the minimum support threshold of NegI-NSP. For DS1, the MLMS of *ms*(1)=0.09 corresponds to *ms*(1)=0.12, *ms*(2)=0.11, *ms*(3)=0.1 and *ms*(4)=0.09. For DS2, the MLMS of *ms*(1)=0.17 corresponds to *ms*(1)=0.2, *ms*(2)=0.19, *ms*(3)=0.18 and *ms*(4)=0.17. For DS3, the MLMS of *ms*(1)=0.2 corresponds to

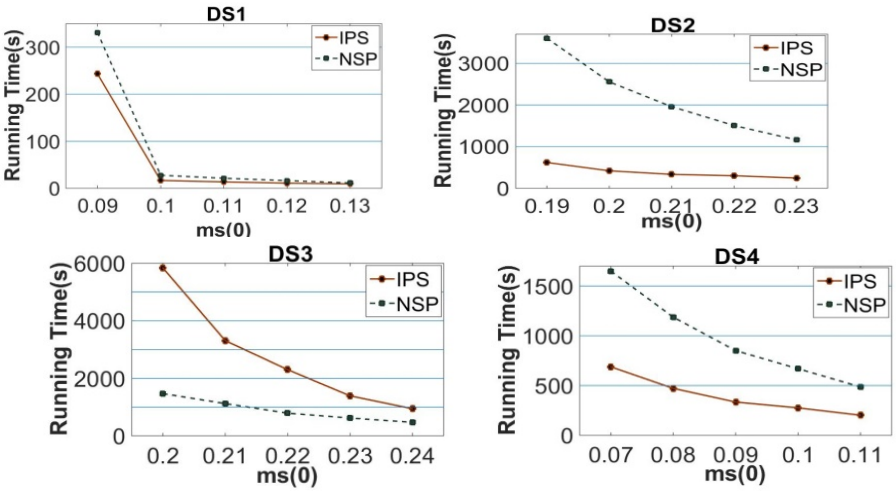

Figure 3: Running time(s).

$ms(1)$=0.25, $ms(2)$=0.24, $ms(3)$=0.23, $ms(4)$=0.22, $ms(5)$=0.21 and $ms(6)$=0.2. For DS4, the MLMS of $ms(1)$=0.07 corresponds to $ms(1)$=0.09, $ms(2)$=0.08 and $ms(3)$=0.07. For the same dataset, the interval between multiple minimum supports remains unchanged.

From figure 4, although the number of NSP by MLMS-NSP is less than NegI-NSP, the number of long size sequences is not reduced. The MLMS-NSP reduces the number of short size sequences with lower support. This shows that MLMS can not only mine NSP from different levels, but also select NSP more accurately. From figure 5, we can see that the runtime of MLMS-NSP is less than NegI-NSP. This is proportional to

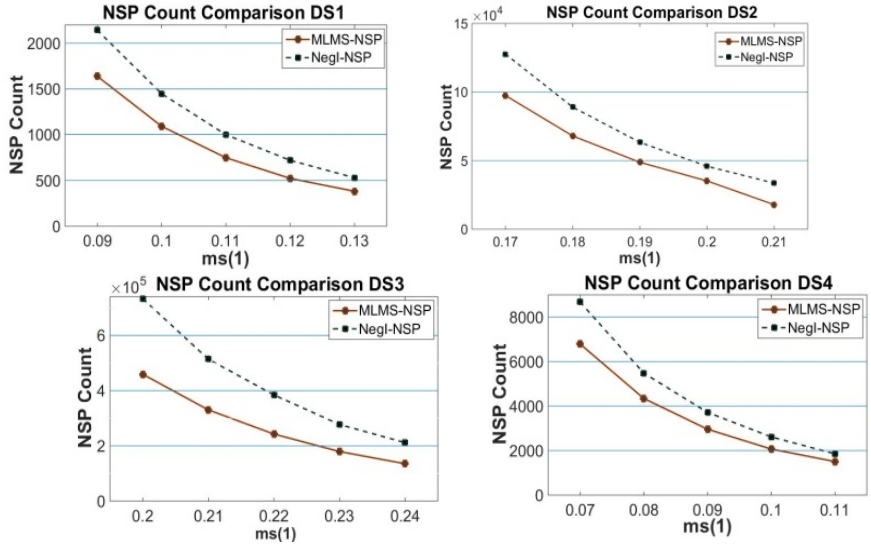

Figure 4: The comparisons of NSP count.

the number of NSP in figure 4. Two algorithms are used for mining NSP based on items. The difference is that MLMS-NSP uses MLMS to select NSP instead of ms. The reduction of the number of NSP leads to the enhancement of MLMS-NSP algorithm mining efficiency. Therefore, the MLMS-NSP algorithm is more effective in mining NSP.

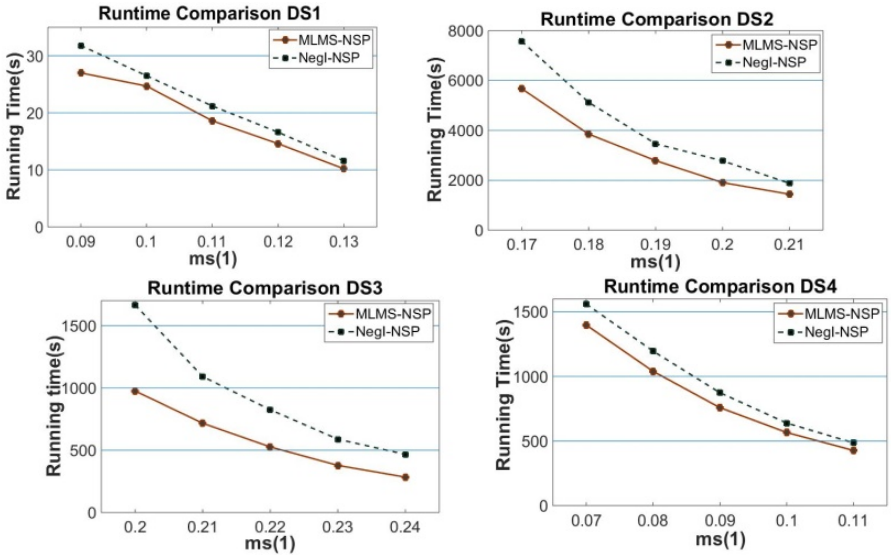

Figure 5: The comparisons of running time.

6. Conclusions and Future Work

NSP provides a special perspective for analysts to capture more valuable information. Although NegI-NSP can efficiently mine NSP based on items, it cannot solve many crucial problems, such as (1) it uses a single minimum support to all size sequences, which is unfair to long size sequence; (2) it only mines NSP from PSP, not from infrequent positive sequences (IPS), which also contained many useful NSP. To solve the problems, we first proposed MLMS scheme, i.e., assign different *ms* to sequences with different sizes. Secondly, we combine MLMS to constrain IPS and successfully mine NSP from these IPS. Finally, we propose an efficient algorithm, named MLMS-NSP, to mine NSP based on MLMS from PSP and IPS. Experimental results show that the MLMS-NSP can effectively mine NSP from PSP and IPS, and the time efficiency is higher than using single *ms*.

In the future, we will focus on selecting actionable NSP and improving the efficiency of NSP mining.

References

[1] L.B. Cao, P.S. Yu, V. Kumar, Nonoccurring behavior analytics: a new area, IEEE Intell. Syst 30(6) (2015) 4–11.
[2] L.B. Cao, Y. Ou, P.S. Yu, Coupled behavior analysis with applications, IEEE Trans. Knowl. Data Eng 24(8) (2012) 1378C-1392.
[3] L.B. Cao, X.J. Dong, Z.G. Zheng, e-NSP: Efficient negative sequential pattern mining, Artificial Intelligence 235 (2016) 156–182.
[4] Y.H. Zhao, G.R. Wang, X. Zhang, et al. Learning phenotype structure using sequence model, IEEE Transactions on Knowledge & Data Engineering 26(3) (2014) 667–681.
[5] S.C. Hsueh, M.Y. Lin, C.L. Chen, Mining negative sequential patterns for e-commerce recommendations, in: APSCC08, IEEE (2008) 1213C1218.
[6] C.L. Liu, G.H. Lv, X.J. Dong, H.N. Yuan, X.J. Dong, Selecting actionable patterns from positive and negative sequential patterns, Journal of Residuals Science & Technology 14(1) (2017) 407-419.
[7] S. Kamepalli, R. Kurra, Frequent negative sequential patterns: a survey, Int. J. Comput. Eng. Technol 5(3) (2014) 15C121.
[8] P. Kazienko, Mining sequential patterns with negative conclusions, in: DaWaK2008 (2008) 423C432.
[9] V.K. Khare, V. Rastogi, Mining positive and negative sequential pattern in incremental transaction databases, Int. J. Comput. Appl. 71(1) (2013) 18C22.
[10] N.P. Lin, H.J. Chen, W.H. Hao, Mining negative sequential patterns, in: WSEAS2007 (2007) 654C658.
[11] Z. Zheng, Y. Zhao, Z. Zuo, L. Cao, Negative-GSP: an efficient method for mining negative sequential patterns, in: Data Mining and Analytics (AusDM09), 101 (2009) 63C67.
[12] Z. Zheng, Y. Zhao, Z. Zuo, L. Cao, An efficient GA-based algorithm for mining negative sequential patterns, in: PAKDD10, in: LNCS, 6118 (2010) 262C273.
[13] Y.S. Gong, T.T. Xu, X.J. Dong, G.H. Lv, e-NSPFI: Efficient mining negative sequential pattern from both frequent and infrequent positive sequential patterns, International Journal of Pattern Recognition and Artificial Intelligence,31(2)(2016) 3-14.

[14] P. Qiu, L. Zhao, X.J. Dong, NegI-NSP: Negative sequential pattern mining based on loose constraints, IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society. Proceedings, DOI: 10.1109/IECON.2017.8216579.
[15] X.J. Dong, Z.G. Zheng, Z. Niu, Mining infrequent itemsets based on multiple level minimum supports. International Conference on Innovative Computing, Information and Control. IEEE, (2007) 528-528.
[16] T.T. Xu, X.J. Dong, J.L. Xu, Y.S. Gong, E-msNSP: Efficient negative sequential patterns mining based on multiple minimum supports, International Journal of Pattern Recognition and Artificial Intelligence, 31(2)(2017).
[17] Y.H. Zhao, G.R. Wang, Y. Yin, et al. Improving ELM-based microarray data classification by diversified sequence features selection. Neural Computing & Applications, 27(1) (2016) 155-166.
[18] W. Ouyang, Q. Huang. Mining Positive and Negative Sequential Patterns with Multiple Minimum Supports in Large Transaction Databases, Second Wri Global Congress on Intelligent Systems. IEEE Computer Society, (2010) 190-193.
[19] Y.H. Zhao, G.R. Wang, Y. Li, et al. Finding novel diagnostic gene patterns Based on interesting non-redundant contrast sequence rules. IEEE, International Conference on Data Mining. IEEE, (2012) 972-981.
[20] Y.C. Zhao, H.F. Zhang, L.B. Cao, C.Q. Zhang, H. Bohlscheid, Mining both positive and negative impact-oriented sequential rules from transactional data, in: PAKDD09, in: LNCS, 5476 (2009) 656C663.
[21] Y. Li, Y. Zhao, G. Wang, et al. ELM-Based large-scale genetic association study via statistically significant pattern. IEEE Transactions on Systems Man & Cybernetics Systems, DOI: 10.1109/TSMC.2017.2720702.
[22] Y. Zhao, J.X. Yu, G. Wang, et al. Maximal subspace coregulated gene clustering, IEEE Transactions on Knowledge & Data Engineering, 20(1) 2007 83-98.
[23] X.J. Dong, Z.D. Niu, D.H. Zhu, Z.Y. Zhang, Q.T. Jia, Mining interesting infrequent and frequent itemsets based on MLMS model. The Fourth International Conference on Advanced Data Mining And Applications (ADMA2008), Chengdu, China, Springer-Verlag Berlin Heidelberg, (2008) 444-451.
[24] R. Srikant, R. Agrawal, Mining sequential patterns: generalizations and performance improvements, in: EDBT1996, 1057 (1996) 1C17.
[25] Y.S. Gong, C.L. Liu, X.J. Dong, Research on typical algorithms in negative sequential pattern mining, Open Automation & Control Systems Journal, 7(1) (2015) 934-941.
[26] W.M. Ouyang, Q.H. Huang, Mining negative sequential patterns in transaction databases, in: ICMLC2007, (2007) 830C834.
[27] N.P. Lin, W.H. Hao, H.J. Chen, C.I. Chang, H.E. Chueh, An algorithm for mining strong negative fuzzy sequential patterns, Int. J. Comput. 3(1) (2007) 167C172.
[28] X.J. Dong, Y.S.Gong, L.B.Cao. F-NSP+: A fast negative sequential patterns mining method with self-adaptive data storage, Pattern Recognition, 84 (2018) 13-27.
[29] X.J. Dong; Y.S. Gong; L.B. Cao, e-RNSP: An efficient method for mining repetition negative sequential patterns, IEEE Transactions on Cybernetics, DOI: 10.1109/TCYB.2018.2869907.
[30] C.L. Liu, X.J. Dong, C.Y. Li, L. Li, SAPNSP: Select actionable positive and negative sequential patterns based on a contribution metric, Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on. IEEE, (2015) 811-815.
[31] X.J. Dong, C.L. Liu, T.T. Xu, DK Wang, Select actionable positive or negative sequential patterns,Journal of Intelligent & Fuzzy Systems(fsdm2015), 29(6) (2015) 2759-2767.
[32] T.T. Xu, T.X. Li, X.J. Dong. Efficient High Utility Negative Sequential Patterns Mining in Smart Campus, IEEE Access, 6 (2018) 23839-23847.
[33] X.J. Dong, G. Li, H.G. Wang H, et al. Mining infrequent itemsets Based on extended MMS model. International Conference on Intelligent Computing. Springer Berlin Heidelberg, (2007) 190-198.
[34] X.J. Dong, C.L. Liu. Mining interesting infrequent and frequent itemsets based on multiple level minimum supports and minimum correlation strength. International Journal of Services Technology & Management, 21 (2015) 301–316.
[35] X.J. Dong, Z.D. Niu, X.L. Shi, X.D. Zhang, D.H. Zhu, Mining both positive and negative association rules from frequent and infrequent itemsets. Proceedings of the Third International Conference on Advanced Data Mining and Applications (ADMA 2007), Harbin, China, (2007) 122-133.
[36] X.J. Dong, S.J. Wang, H.T. Song. 2-level support based approach for mining positive & negative association rules. Computer Engineering, 31(10) (2005) 16-18.
[37] X.Q. Han, X.J. Dong, H. Jiang, et al. 2L-XMMS: An efficient method for mining infrequent itemsets with 2-Level multipul minimum supports, Proceedings of the 2012 International Conference on Communication, Electronics and Automation Engineering. Springer Berlin Heidelberg, 2013.