DOI:10.3233/JIFS-17092
IOS Press

Maximal fuzzy supplement frequent pattern mining based on advanced pattern-aware dynamic search strategy and an effective FSFP-array technique

Haiqing Zhang[a,*], Tao Wang[b], Daiwei Li[a,c], Abdelaziz Bouras[d], Xi Xiong[e] and Shaojie Qiao[e]
[a]*School of Software Engineering, Chengdu University of Information Technology, Chengdu, China*
[b]*DISP Laboratory, INSA Lyon, UJM-Saint Etienne, France*
[c]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China*
[d]*Department of Computer Science, Qatar University, ictQATAR, Doha, Qatar*
[e]*School of Cybersecurity, Chengdu University of Information Technology, Chengdu, China*

Abstract. The proper expression of the potentially useful but hidden information in large-scale datasets via using proper structure is vital important in both theory and applications of advanced pattern mining. The fundamental challenges are how to alleviate the mining combinatorial explosion problem and ensure the efficiency of mining results. However, most of the existing algorithms have not been entirely capable of solving these issues due to the fact that enormous number of candidate patterns has been generated and the weight constraints of items were only considered in crisp values. In order to generate more practical patterns in the new proposed Fuzzy Supplement Frequent Pattern (FSFP), base-(second-order-effect) pattern structure is proposed and new pruning strategies including pattern-aware dynamic base pattern search strategy and FSFP-array technique are given. Thus, the proposed maximal FSFPs mining algorithm guarantees efficient mining performance by scanning the dataset only once, preventing overheads of pattern extraction based on the pruning strategies, and adopting fuzzy weight conditions to enhance the dependability of mining results. The extensive experimental results obtained from nine benchmark datasets indicate that our algorithm has outstanding performance in comparison to PADS and FPMax* algorithms.

Keywords: Frequent pattern mining, fuzzy weight conditions, pattern-awareness, dynamic base pattern search

1. Introduction

Mining potentially useful but hidden information from large-scale databases is one of the main goals of advanced pattern mining. The fundamental pattern mining methods, including Apriori [1] and FP-growth [2], and the properties of these two algorithms have been widely applied in other research works. In order to address the challenges of large data growth, more extensively studied algorithms have been proposed, including, but not limited to, sequential frequent patterns [3], top-k frequent patterns [4], weighted frequent patterns [5], and high-dimensional patterns [6]. The above mentioned frequent pattern mining methods are based on the downward closure property of frequent patterns that all of the sub-patterns are frequent for any frequent pattern. However, in accordance with the practical experience of medical field,

[*]Corresponding author. Haiqing Zhang, School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China. E-mail: zhanghq@cuit.edu.cn.

the information and knowledge discovery for diverse patients tend to span several departments that generate a large number of relative frequent items and lower frequent items. In order to elaborate the complexity of diseases emergence, the combination of relative frequent items and relative lower frequent items should be analysed to perceive meaningful hidden patterns from large-scale medical datasets. Thus, this study aims to discover the diseases that are closely related with current diseases or those that would most likely be induced or pulled out diseases by common diseases instead of only discovering the association rules among common diseases.

In order to achieve the aim of reducing the collection number of frequent itemsets and meet the requirements of low runtime and low memory usage for large amount of data, research works have been proposed to mine maximal sequential patterns (MSP) [7] and closed sequential patterns(CSP) [8].

Although MSP and CSP can partly alleviate the mining expansion problem, MSP and CSP mining cannot fundamentally resolve the issue of low mining effectiveness [9]. For instance, for the given sample transaction database in Table 1 and the corresponding practical meaning for each item is shown in Table 2. It generated 181 association rules under the constraints of *minimum support*$\geq$4, *minimum confidence*$\geq$0.6, and a new added parameter that is used to guarantee positive association: *lift*$\geqslant$1. Based on the analysis of extracted MSP and CSP for Table 1, the longest frequent pattern is: <pgn>, and the association rules are occurred in patterns: <g,n,e>, <g,n,s>, <p,g,n>, <p,g,n,a>, <p,g,e,a>, <p,n,a,e>, <g,n,a,e>, and <p,n,a,e,g>. But according to practical meaning, more meaningful but second frequent patterns should be extracted. For instance, the meaningful pattern <b,c,m,h> (<*Fatty Liver Disease*, ***Overweight BMI***, *Dyslipidemia* (Elevated Triglycerides), ***Obesity***) should be extracted, it is because that ***Obesity*** and ***Overweight BMI*** tend to have close relationship with *Dyslipidemia* and *Fatty Liver Disease*. The obtained association results show that the mined frequency patterns from MSP and CSP are not sufficient to represent all practical frequent patterns. Meanwhile, mining the significant practical patterns is usually not to extract the most frequent patterns. The reason is that most of the mining results can be easily known for many skilled professional people, which caused the mining results are not that useful.

Table 1
Sample transaction database

TID	Items	Traditional Frequent Items
T_1	b,h,i,o,p,j	o,p,b,h
T_2	c,d,g,i,m,n,e,s	g,n,c,e,m,s
T_3	a,b,l,m,o	o,a,b,m,l
T_4	b,h,m,o	o,b,m,h
T_5	b,c,h,p,s,r,x,y,j	p,b,c,s,h
T_6	a,c,e,l,m,n,p,g,q,s,d	g,n,p,a,c,e,m,s,l
T_7	a,r,x,g,e,n,q,p	g,n,a,e,p
T_8	b,c,h,j,m	b,c,m,h
T_9	a,e,n,g,s,p,q,o,r,y	g,n,o,p,a,e,s
T_{10}	a,c,e,l,n,o,p,g	g,n,o,p,a,c,e,l
T_{11}	y,p,n,g,s	g,n,p,s
T_{12}	o,l,d,i,z	o,l

Table 2
The corresponding practical meaning of each item in Table 1

Item	Disease	Item	Disease
a	Dyslipidemia (elevated cholesterol)	m	Dyslipidemia (elevated triglycerides)
b	Fatty Liver Disease	o	Chronic pharyngitis
c	Overweight BMI	p	Visual Impairment
d	Increased Percentage of Lymphocytes	x	Elevated Blood Pressure
e	Thyroid	r	Kidney Stone
g	External hemorrhoid	s	Increased Platelet
h	Obesity	q	Chronic Cervicitis
i	ECG detection: sinus tachycardia	n	Breast lobular hyperplasia
j	Elevated Alanine Aminotransferase	y	Elevated Blood Glucose
l	Trachoma	z	Urine Occult Blood: +

Based on above analysis, in the studied Medical dataset, the patients tend to have several main certain diseases (base items) and several induced diseases (second order effect items) during a defined period, and the changes of the structure of the base-(second-order-effect) patterns depend on the changes of the related fuzzy weight. Thus, in this paper, a novel structure called FSFP-Tree is proposed to extract the relations among different items based on the integrated fuzzy weights gained from different constraints. The new FSFP-tree conducts pattern analysis and pattern extraction more accurately by selecting core patterns due to pattern-aware dynamic base pattern search strategy. In order to reduce the traversing time and searching space, an advanced probing mechanism and FSFP-array technique is adopted to enhance the mining performance and efficiency. The clearly noticeable advantage is that the invalid subpatterns can be pruned sharply to reduce excessive combinations. The proposed algorithm significantly outperforms the efficient algorithms of PADS [10] and FPMax* [11] based on experimental analysis in Section 5.

This remainder of this paper is structured in the following way. In Section 2, the preliminary concepts and the problem statement of mining structure are given. In Section 3, the details of FSFP-Tree including structure, construction, and the corresponding pruning strategies that contain pattern-aware dynamic base pattern search strategy and FSFP-array technique are described. On the basis of Section 3, maximal FSFP mining algorithm is given in Section 4. In Section 5, extensive experimental results in terms of mined base pattern analysis, runtime analysis, and memory usage analysis are presented to demonstrate the performance of the proposed algorithm. The paper is concluded in Section 6.

2. Preliminary concepts and problem statement

This paper addresses the problem of mining appropriate and efficient maximal frequent patterns based on the characteristics analysis of medical transaction datasets. In previous works [9, 12], the weights are only defined by comparing with the differences among items, and the weight values are only assigned in crisp values. Being different from these works, the weighted support of each item is composed by four parts: *item weight in itemset*, *itemset weight in database*, *item weight in transaction*, and *itemset weight in transaction*. In order to resolve the uncertain and inaccurate problems embedded in medical items, we represent the weights by using fuzzy set to handle these uncertainness problems since fuzzy logic is a sophisticated approach to tackle uncertain issues [13, 14].

The transaction database is denoted as T, where $\mathrm{T} = \{\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_n\}$, T_i is a transaction, and $\mathrm{T}_i = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_m\}$ means one transaction composed by multiple items, where $i_j \in \mathrm{I}$, $\mathrm{I} = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_k\}$ ($\mathrm{m} \leqslant \mathrm{k}$), I is the set of all items. For each item i_i in the itemset I, i_i has a weight based on its hazard level. For a given $\mathrm{I} = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{i,} \ldots \mathrm{i}_k\}$, the fuzzy itemset weight for I is defined as $\widetilde{V} = \{\widetilde{v_1}, \widetilde{v_2}, \cdots, \widetilde{v_\mathrm{i}}, \cdots, \widetilde{v_k}\}$ *(global weight)*. For the same item i_i, currently i_i is in transaction $\mathrm{T} = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{i,} \ldots \mathrm{i}_m\}$ ($\mathrm{m} \leqslant \mathrm{k}$), the comparison weights of items in the transaction are defined as: $\widetilde{W} = \{\widetilde{w_1}, \widetilde{w_2}, \cdots, \widetilde{w_i}, \cdots, \widetilde{w_m}\}$ *(local weight)*. The overall value of global and local weight is used to express the differences of the severity level for the same item to different patients.

Based on the characteristics analysis of medical data, the expected mined itemsets are separated into two groups in terms of the items that belong to the base pattern and second order effect items. Given a base pattern, *BP*, it is the core part of any transaction *T* and consisted by the core elements in *I*. Given a pattern *SOP*, which is the second order effect item in *I* that pulled by the items in the base pattern.

Definition 1. The Severity Level of Pattern (P) (SL_P). Given a pattern $\mathrm{P} = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{i,} \ldots \mathrm{i}_n\}$, the weights for each item i_i in P for transaction T_i is $\widetilde{\mathrm{W}}(\mathrm{T_i}) = \{\widetilde{\mathrm{w}_1}(\mathrm{T}_1), \widetilde{\mathrm{w}_2}(\mathrm{T}_2), \cdots, \widetilde{\mathrm{w}}_\mathrm{i}(\mathrm{T_i}), \cdots, \widetilde{\mathrm{w}_\mathrm{n}}(\mathrm{T_n})\}$, the inherent weight for each item i_i from itemset I is $\widetilde{\mathrm{V}} = \{\widetilde{\mathrm{v}}_1, \widetilde{\mathrm{v}}_2, \cdots, \widetilde{\mathrm{v}}_\mathrm{i}, \cdots, \widetilde{\mathrm{v}}_\mathrm{n}\}$. Where, the range of each value in $\widetilde{\mathrm{W}}$ and $\widetilde{\mathrm{V}}$ is in a fuzzy scale. The severity level of P is defined as Equation (2.2).

$$\widetilde{\mathrm{w}}_\mathrm{i} = \frac{\sum_{j=1}^{|T_i|} \widetilde{\mathrm{w}}_\mathrm{j}(T_j)}{|T_j|} \tag{2.1}$$

$$SL_P = \frac{\sum_{i=1}^{n} \widetilde{\mathrm{w}}_\mathrm{i} \bigotimes \widetilde{\mathrm{v}}_\mathrm{i}}{n} \tag{2.2}$$

The severity levels of pattern P are defined with assigning weights lie in [0, 10]. The sets of items are categorized into five levels ($\widetilde{\mathrm{V}}$) based on the harm extent of diseases. The fuzzy range is defined similar with reference [13]. The fuzzy scale is defined with five severity levels, which is represented by Degree (+), Degree (++), Degree (+++), Degree (++++), and Degree (+++++) with the related assigned values are [0 1 3], [1 3 5], [3 5 7], [5 7 9], and [7 9 1], respectively. The range of the comparison weights of items in one transaction ($\widetilde{\mathrm{W}}$) is the same with $\widetilde{\mathrm{V}}$. Some items (diseases) are defined as the base objects which may lead to other items (diseases) or cause deadly hazard.

Definition 2. The Severity Level of Transaction ($\mathrm{SL}(\mathrm{T}_i)$). Given a transaction $\mathrm{T}_i = \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_m\}$, the weights among transactions are different since some transactions may contain high priority items while others do not.

$$SL(T_i) = \frac{\sum_{i=1}^{m} (\widetilde{w_{i(T_i)}} \otimes \widetilde{v_i})}{m} \tag{2.3}$$

Where, m is the number of items in one transaction.

Definition 3. Support of Pattern (P) (SUP(P)) Based On Severity Level.

$$Calculate(P, T_\mathrm{i}) = \begin{cases} 1, & \text{if } \mathrm{P} \in \mathrm{T} \\ 0, & \text{if } \mathrm{P} \notin \mathrm{T_i} \end{cases} \tag{2.4}$$

$$SUP(P) = \frac{\sum_{i=1}^{|T_i|} Calculate(P, T_i) \times SL_p}{\sum_{i=1}^{|T|} SL(T_i)} \tag{2.5}$$

The frequency of pattern P in transaction T_i is calculated based on Equation (2.4), where $|T_i|$means the length of T_i. SUP(P) is a triangular fuzzy number that can be equivalently expressed by a triple of real numbers. The defuzzification method in this paper follows the fuzzy set operation shown in research work [15].

The useful pattern discovery in this paper focus on the combination items from the BP and SOP, which is called Fuzzy Supplement Frequent Pattern (FSFP). The FSFP is generated from two cases based on the functionalities of items: 1) All of the specific base items simultaneously appear together with partly second order effect items. The specific base items have very high severity level and ensure these items have the ability to attract the second order effect items with lower support counts. 2) Part of the specific base items simultaneously appear together with partly second order effect items. This means that only a part of the specific base items can be treated as core and able to pull the lower support items into a transaction. The association rules should also consider the influence from not occurred base items on full transaction since these base items may reduce or change the adsorption ability of core items and change the activeness of adsorbed items. The definition of FSFP is given in Definition 4.

Definition 4. (Fuzzy Supplement Frequent Pattern). Base on the above discussion, the set of FSFP is defined in Equation (2.6) and the constraints of generating FSFP is shown in Equation (2.7).

$$\mathrm{FSFP} = \begin{cases} (\bigcup_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{bp_i}) \cup (\bigcup_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{sop_i}),\ \ (\text{where } \mathrm{bp_i} \in \mathrm{BP},\ \ \mathrm{sop_i} \in \mathrm{SOP}); \\ (\bigcup_{\mathrm{i}=1}^{x} \mathrm{bp_i}) \cup (\bigcup_{\mathrm{i}=1}^{n-x} \neg bp_i) \cup (\bigcup_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{sop_i}) \end{cases} \tag{2.6}$$

$$\begin{cases} SUP(FSFP) = (SUP^L(FSFP), SUP^M(FSFP), SUP^U(FSFP)) \\ \sigma \leq \theta \leq SUP^L(\bigcup_{i=1}^{n} bp_i) \\ \sigma \leq SUP^L(\bigcup_{i=1}^{n} sop_i) \leq SUP^U(\bigcup_{i=1}^{n} sop_i) \leq \theta \\ \sigma \leq SUP^L((\bigcup_{i=1}^{n} bp_i) \cup (\bigcup_{i=1}^{m} sop_i)) \leq SUP^U((\bigcup_{i=1}^{n} bp_i) \cup (\bigcup_{i=1}^{m} sop_i)) \leq \theta \\ \varepsilon \leq SUP^L((\bigcup_{i=1}^{x} bp_i) \cup (\bigcup_{i=1}^{n-x} \neg bp_i) \cup (\bigcup_{i=1}^{m} sop_i)) \leq \theta\ (x \leq n) \end{cases} \tag{2.7}$$

Where, SUP(FSFP) is a triangular fuzzy element, *SUP^L(FSFP)* is the lower number, *SUP^M(FSFP)* is the middle number, *SUP^U(FSFP)* is the upper number, and it has: *$SUP^L(FSFP) \leqslant SUP^M(FSFP) \leqslant SUP^U(FSFP)$.* If *$SUP^L(FSFP) = SUP^M(FSFP) = SUP^U(FSFP)$,* and then SUP (FSFP) is the traditional crisp number. Symbol '¬' means not occurring together with other items, such as $(\bigcup_{\mathrm{i}=1}^{x} \mathrm{bp_i}) \cup (\bigcup_{\mathrm{i}=1}^{n-x} \neg bp_i) \cup (\bigcup_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{sop_i})$ means all the items of $(\bigcup_{\mathrm{i}=1}^{n-x} \neg bp_i)$ not occurring together with itemset $(\bigcup_{\mathrm{i}=1}^{x} \mathrm{bp_i}) \cup (\bigcup_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{sop_i})$ in a transaction. The minimum support threshold of base items is defined as ***minsup***, which is θ (where θ is the defined minimum overall weight of an item in BP); the second threshold ***min_connect_sup*** is defined to determine the boundary of the severity level between BP and SOP, which is σ (where $\sigma \leqslant \theta, \sigma$ is the defined minimum overall weight of an item in SOP); and ε is an extra parameter that used to adjust the range of support. The value of ε is obtained based on experiment test.

3. Fuzzy supplement frequent pattern mining with fuzzy weight conditions

3.1. Pattern-aware dynamic base pattern search strategy

Pattern-aware search conception is firstly proposed by Zeng et al. [10], which has improved the dynamical ordering strategy used in Mafia [16], GenMax [17], and FPMax* [11]. According to the conception of pattern-aware dynamic search, the subtree is firstly constructed with a method of searching the potential max-patterns that are scheduled into some branches, and then the previous found patterns can be pruned as they were the subsets of the max-patterns. However, base patterns mining is different from finding the maximum pattern mining, which requires considering the items' fuzzy support value based on its severity level. Moreover, the proposed multiple-probing process in pattern-aware dynamic search cannot ensure the newly obtained longer maximal frequency patterns to have a better pruning capability, meanwhile the expanding of the probing process would significantly increase the time complexity. Therefore, in order to discover the longest and optimal base pattern, a novel advanced probing mechanism is given to reduce traversing time and searching space.

In order to better explain the proposed probing mechanism, we first give some primary notations. Set

I be an itemset, *P* node is the head of current search in the set enumeration tree. ∅ is an empty itemset. Since different ordering of the tail can affect the computation efficiency [10], we focus on utilizing the order of items in *Tail* to fully utilize the newly obtained base patterns and avoiding the cost of projecting database construction. Thus, we give the definitions of key search elements of *Tail, untrimmed Tail*, and *the new generated tail* as follows.

Definition 5. ***Tail(T)***. *Tail(T)* is the tail of an itemset T, which is also used to deposit the candidate base pattern of itemset T.

Definition 6. ***UnTrim(Q)***. Suppose Q is the child of node P in the set enumeration tree, $Q = P \cup I'$, *UnTrim(Q)* is defined as the untrimmed tail of itemset Q, suppose the last item in *P* is *i*, and I' is the set of items ordered after item *i* in *Tail(P)*, then it has the relation that:

$$UnTrim(Q) = \{i' | i' \in Tail(P), \forall\, i' \in Q,\; orderof(i') \prec orderof(i)\} \quad (3.1)$$

where *orderof(i)* means the position of item *i* based on the defined order, and $orderof(i') \prec orderof(i)$ denote the order of item *i'* follow item *i* in current order.

Definition 7. New generated *Tail(Q)* based on *Tail(P)*. As it is based on Definitions 5 and 6, *Tail(Q)* is extracted from *UnTrim(Q)*, and it is used to deposit the tail of itemset Q and to save the potential base pattern that is generated by the elements in itemset Q. Thus, we have the following equation to define the new generated *Tail(Q)*:

$$Tail(Q) = \{t \in UnTrim(Q) | Q \bigcup \{t\}\} \quad (3.2)$$

The proposed advanced dynamic base pattern search strategy has the following steps. 1) Searching the subtree rooted in an itemset I, if the *I*∪*Tail(I)* is not a subset of any base pattern and *Tail(I)* $\neq \emptyset$, and then we apply the advanced probing mechanism to find the longest base pattern BP_1 in the current focused subtree. 2) Selecting one longer base pattern LBP_1 as the key pattern from all of the found longer base patterns. 3) Reordering the items in *Tail(I)* in defined order, and constructing the I-projected nodes in FP-tree structure. All the children of itemset I in the FP-tree structure can be divided into two categories: a) the potential children, which may contain new base patterns in its descendant nodes; and b) the impossible potential children, which is impossible to contain new base patterns in its descendant nodes. For each item $i \in Tail(I)\text{-}LBP_1$, $I \cup \{i\}$ is a potential children for *I*. For each item $i \in Tail(I) \cap LBP_1$, $I \cup \{i\}$ is an impossible potential children for I. We only recursively search the potential children to determine the final base pattern. We improved the probing process to reduce the traversing time and searching space by comparing with research work Zeng et al. [10].

The probing processes work as follows: 1) Determine the analyzed items and input into set *UnTrim(o)*. For each item *i* in *UnTrim(o),* we need to check whether $o \cup \{i\}$ can satisfy the conditions of base pattern by the following steps. 2) Search the FP-tree structure and determine the paths that contain the current focused item *o*; 3) Start from the first item in the first path, if $o \cup \{i\}$ can satisfy the conditions of base pattern, and then input item *i* into set *E*; 4) If the length of *E* is greater than or equal to the current minimal length of all paths or the number of common items among paths is greater than or equal to the defined lower limit, we calculate the combination set and determine the basic base pattern, and then update the value of minimal length of all paths, the node link, and set *E*. 5) We repeat the steps from 2)–4) till all items in set *UnTrim(o)* have been examined.

We select the first five transactions in Table 1 as an example, and update fuzzy support value and the related count number of each item. Set the minimum support value $\theta = 0.2$ and the minimum base frequency *core_count_number=2*. Then the potential base pattern is $P_1 = \{j,m,h,b,p,i,o\}$, which is sorted based on the corresponding fuzzy support value. Set the initial P= ∅, the initial Tail(P) = P_1. Set Q = {m} is the child of itemset P, then the untrimmed tail of item *m* in the set enumeration tree is *UnTrim(Q)* = {h,b,p,i,o}. For each item *t*∈ *UnTrim(Q)*, if Q∪{t} satisfy the base pattern condition in Definition 5, then insert new item *t* into *Tail(Q)*.

3.2. The FSFP-Tree construction algorithm

A Fuzzy Supplement Frequent Pattern Tree (FSFP-Tree) with the corresponding transaction database is a compressed tree that is used to mine FSFPs. FSFP-Tree structure is adopted to manage and maintain the information regarding continually added transactions and the related overall weights. All of the base items are not deleted, if the supports of base items are lower than a given minimum support threshold, then this base items with symbol '¬' are inserted into FSFP-Tree. The reason is that the invalid base items still have influence on the current transaction due to the special functionalities of base items. Thus, all of the base items are contained in the head table of FSFP-Tree. The other items are determined based on the count number of appearing times and the corresponding severity level.

The key elements of constructing the structure of FSFP-Tree is given as follows:

- It has a root node labeled with "Root";
- Each node in FSFP-Tree has seven attributes: **item-name**, **branch-level**, **parent**, **children**, **node-link**, **fuzzy support, count number**, and **node-link-core**. All nodes with the same item-name are linked together by node-link, and all branches containing current core-itemset are linked together by node-link-core which points to the first node in bottom-up order. Note that the fuzzy support of current pattern is calculated based on the overall severity level of each item and its count number (The specific calculation method is shown in Definition 3). In order to show the frequency of each item, count number is displayed in FSFP-tree as an attribute.
- **coreItems** parameter is used to record the current base items that consists of current items, current un-happened items, **count number**, and **fuzzy support**, and head of **node-link-core**.
- A **header table** is constructed for items. The items are in the descending order of the transaction database. Each entry in the header table consists of two attributes: **item-name** and **head** of the **node-link** marked by the same item-name.

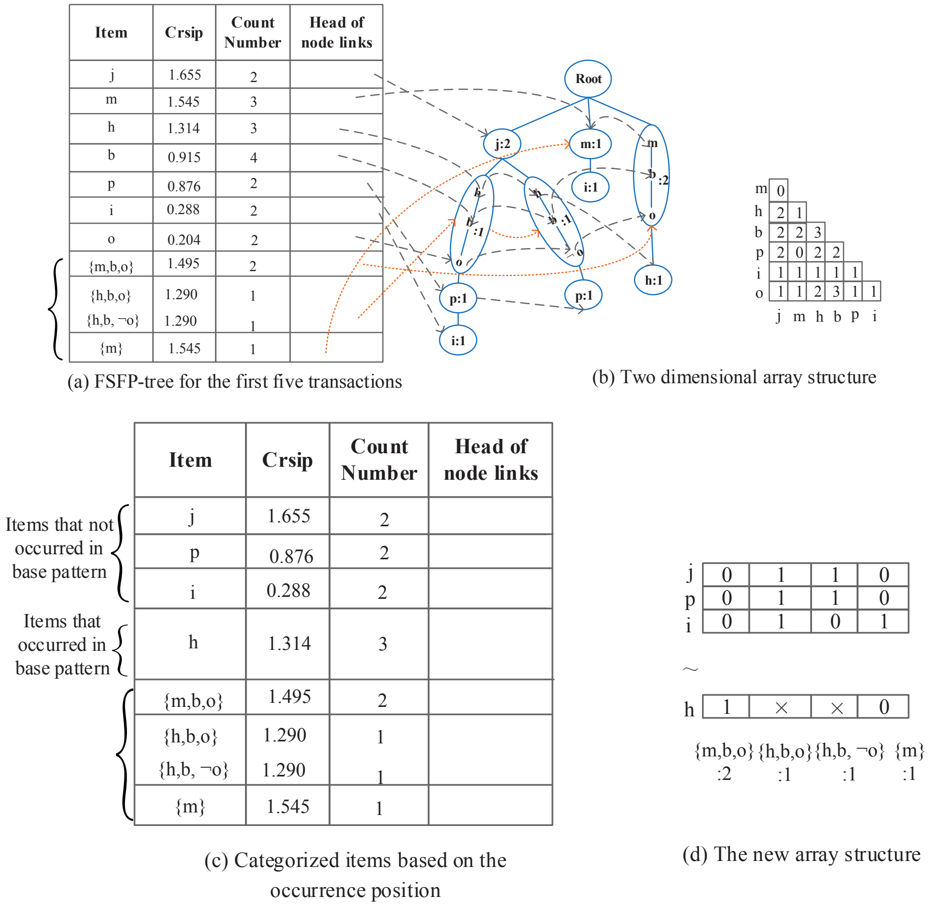
Remark 1. Some notations that are used in Algorithm 1 are given as follows: Transaction Database is remarked as ***TDs***; the allowed minimum frequency of connected items is remarked as ***connect_count_number***; the allowed minimum frequency of base items is noted as ***core_count_number***; θ is used to express the minimum fuzzy support of base items; the minimum fuzzy support of the connected items is σ; p is noted as the first item in a transaction, and q is the remaining item in addition to coreItems.

Algorithm 1 FSFP-Tree (*TDs*, *core_count_number, connect_count_number*, θ, σ)

```
Input: TDs, core_count_number,
       connect_count_number, $\theta$, $\sigma$.
Output: FSFP-Tree
    BEGIN
Scan TDs once and Determine the connected and base items based on Definition 4.
Create the root node of FSFP-Tree: 'Root':
for each transaction $T_i$ in TDs do
    Determine the related base pattern (coreItems) for each transaction based on section 3.1.
end for
Create header table. And then updated FSFP-Tree. The sorted list for each transaction $T_i$ is [p|coreItems |q].
if p$\in$q then
    if T has children n, and has n.item-name=p.item-name then
        countNumber(n)++; calculate SUP(n)
    else
        Create new node n, set countNumber(n)=1, recalculate SUP, link its parent node, and link the node to the same item-name through node-link.
    end if
end if
if p$\in$coreItems then
    Pick out the coreItems in the current branch, and note it as p'.
    if T has children n', and has n'.item-name=p'.item-name then
        countNumber(n')++; adjust the branches with the same coreItems; and this [coreItems] is the parent node for the remaining nodes.
    else if Transaction T has child n", has n"$\cap$p' $\neq$Null then
        This [coreItems] is the parent node for the remaining nodes in this path.
    end if
end if
if q$\neq$Null then
    Recursively update FSFP-Tree.
end if
    END
```

3.3. FSFP-array technique for reducing traverse time and memory usage

In order to reduce the tree-traverse time, an array technique is proposed by FPMax* [11] and widely used in several research works [9, 10]. Based on the array technique, FP-tree structure is associated with an array. The array structure stores the frequency of every two pair itemsets for each FP-tree. For every two items i and j in the head table of FP-tree, the array technique is adopted to store the frequency (traditional support) of each pair $\{i,j\}$ in the item-projected database. For each item i in the header table of the projected FP-tree, the examination of the existed frequent set with new added item i can be generated by reading the row of i in the array structure and collecting the item j such that the frequency of cell



Item	Crsip	Count Number	Head of node links
j	1.655	2	
m	1.545	3	
h	1.314	3	
b	0.915	4	
p	0.876	2	
i	0.288	2	
o	0.204	2	
{m,b,o}	1.495	2	

{h,b,o}
{h,b, ¬o}
	1.290
	1.290
		1
		1	

{m}	1.545	1	

(a) FSFP-tree for the first five transactions

m	0					
h	2	1				
b	2	2	3			
p	2	0	2	2		
i	1	1	1	1	1	
o	1	1	2	3	1	1
	j	m	h	b	p	i

(b) Two dimensional array structure

	Item	Crsip	Count Number	Head of node links
Items that not occurred in base pattern	j	1.655	2	
	p	0.876	2	
	i	0.288	2	
Items that occurred in base pattern	h	1.314	3	
	{m,b,o}	1.495	2	

	{h,b,o}
	{h,b, ¬o}
		1.290
		1.290
			1
			1	

	{m}	1.545	1	

(c) Categorized items based on the occurrence position

j	0	1	1	0
p	0	1	1	0
i	0	1	0	1
~				
h	1	×	×	0
	{m,b,o} :2	{h,b,o} :1	{h,b, ¬o} :1	{m} :1

(d) The new array structure

Fig. 1. Pruned counting time based on new proposed array structure.

$\{i, j\}$ is larger than the defined threshold, instead of by scanning the header table of the projected FP-tree.

We focus on improving the version used by Zeng et al. [10] and Yun et al. [9] in order to improve the efficiency in reducing traverse time. Let *BP* be the base pattern set that is selected based on pattern-aware dynamic search strategy. Since the unique base pattern *BP* exists in each path of FSFP-Tree, the maximal FSFPs can be obtained by the base pattern and the associated items in each path, thus the array structure is changed into two dimensions that contain base patterns and the other items. Moreover, according to Definition 4, the potential maximal FSFPs are generated by base pattern and second order effect pattern. The items occurred in second order effect patterns have two categories, namely the items already exist in base patterns and the items do not exist in base patterns. Since the items already occurred in base patterns that currently belong to second order effect pattern are impossible to occur in the base pattern of the current focused path, thus, for any item i_j in base pattern set *BP* appearing in path j, and the current base pattern bp_j in path j, must satisfy the following equation:

$$\{i_j \in BP, bp_j \subset BP,\ i_j \bigcap bp_j = \emptyset\} \qquad (3.3)$$

Consequently, the cell between i_j and bp_j in the array structure do not need to be generated. We give Example 1 to demonstrate the working mechanism for the new proposed array structure.

Example 1. The FSFP-tree for the first five transactions in Table 1 is built based on Algorithm 1, which is shown in Fig. 1(a). The significant feature of FSFP-tree that distinguishes from FP-tree is the defined coreItems structure that contains count number, fuzzy support, and head of node-link-core. The array structure based on the previous technique is shown in Fig. 1(b). The items are categorized into three groups based on the definition of base pattern and second order effect pattern, which is shown in Fig. 1(c). The new proposed array structure based on the identified three groups is shown in Fig. 1(d), where the symbol "×" means that we do not generate this cell.

4. Maximal FSFPs mining algorithm

The mining procedure of maximal FSFPs is given in this section to show how the steps of maximal FSFPs mining are conducted and how the proposed pattern-aware dynamic base pattern search strategy and FSFP-array technique are applied in the mining procedure. The maximal FSFP mining algorithm is represented in Algorithm 2.

Algorithm 2 Algorithm MFSFP Mining(*TDs*, *minmum_count_number,* σ)

```
Input:  Transaction Database TDs,
        minmum_count_number, $\sigma$.
Output: Maximal FSFPs: MFSFPs
        BEGIN
Adopt pattern-aware dynamic base pattern search strategy to determine base pattern set BP.
Construct FSFP-tree for TDs (Algorithm 1).
Construct FSFP-array based on new proposed array structure and conditional database CDB.
if Path $p_i$ is a single-path then
    Generate a new pattern $np_i$ by checking the base pattern in current path.
    if (SUP($np_i$)$\geqslant\sigma$ and superset_check($np_i$) is false) then
        MFSFP=MFSFP$\cup np_i$;
    else
        Record MFSFP=MFSFP$\cup bp_i$;
    end if
else{For multiple paths}
    for each item $a_i$ in TDs.header do
        Generate a new set of frequent items sfi in $a_i$'s conditional pattern base based on FSFP-array structure;
        Sort sfi in descending order based on the corresponding support value;
        Call MFSFP Mining(sfi, minmum_count_number, $\sigma$);
    end for
end if
        END
```

In Algorithm 2, The mining operations should firstly provide the values of *fuzzy support value*, *base patterns*, *FSFP-tree*, and *FSFP-array*. Thus, the fuzzy support value for each item is calculated by Definition 3, base pattern set *BP* is obtained based on pattern-aware dynamic base pattern search strategy, FSFP-tree is constructed based on Algorithm 1, and FSFP-array is built according to new proposed array structure. Note that in base pattern and FSFP-tree construction, the fuzzy severity level and frequency of items should be regarded as the pruning conditions. If the path is a single-path, a new pattern np_i is generated by combining the current base pattern with checking of all of the items in the single path and calculating the new support value. If the new support value is greater than or equal to the minimum threshold and also has no superset for this pattern, then the pattern MFSFP should be regarded as valid maximal FSFPs and inserted into the set of maximal FSFPs. Otherwise, if the new MFSFP generated by the base pattern and the new itemset cannot satisfy the above maximal conditions, we only consider the base pattern with strong absorption abilities for the other items as the maximal FSFP in the current path. For the multiple paths, we generate the corresponding conditional database based on FSFP-array structure and sort the new set by using support value in descending order, and then recursively calls itself after assigning the new set of count number of the corresponding base pattern with each item in the header table until a single-path occurs.

5. Experiments and discussion

5.1. Experimental environment

Performances study including runtime and memory usage analysis is conducted to evaluate the effectiveness of proposed FSFP-tree mining algorithm, pattern-aware dynamic base pattern search strategy, FSFP-array structure, and MFSFP mining algorithm. The experimental results are represented on seven real datasets and two synthetic datasets. The real datasets including Chess, Mushroom, Pumsb, Connect, Accidents, Pumsb_star and two synthetic datasets including T10I4D100K and T40I10D100K have been extensively used in the previous studies as the benchmark datasets, where they can be gained at http://fimi.ua.ac.be/data/. We provide a new dataset called Medical that has a sparse feature and contains real patients suffering from different type diseases. This dataset can be downloaded at http://medical.witaction.com:808/medical/. Table 3 shows the features of the analyzed datasets.

Table 3
The features of benchmark datasets

Data set		No. of tuples	No. of items	Avg. trans. len.	Max. trans. len.
Dense	Chess	3196	76	37	37
	Mushroom	8124	120	23	23
	Pumsb	49046	2113	74	74
	Connect	67577	150	43	43
Sparse	Accidents	2125	268	34	45
	Pumsb_star	49046	2088	50	63
	T10I4D100K	100000	1000	10	29
	T40I10D100K	100000	943	39	77
	Medical	6341	698	6	24

All the experiments were evaluated in a PC computer by running in win7 system, with a 2.20 GHz Pentium i7-3632QM CPU, 8 GB RAM, and a 700 GB hard disk. The programs were written in C++ by using Microsoft Visual Studio 2010. Based on the above analysis, the existing algorithms still cannot efficiently obtain effective maximal frequent patterns, and runtime, memory usage, and combinatorial growth are still the bottleneck of frequent pattern mining. Therefore, these parameters will be the key monitoring objects.

5.2. Base pattern analysis based on dynamic base pattern search strategy

The proposed dynamic base pattern search strategy is applied to extract the base patterns for eight datasets under the parameters of *core_count_number*, *connect_count_number*, θ, and σ. The initial fuzzy global and local weights are assigned with (1,1,1) for the other eight datasets except for Medical dataset. In this paper, we mainly study the sparse datasets and the mined base patterns are shown in Fig. 2.

In Fig. 2, the lines with the same type marker are conducted under the same parameters and the dotted lines are the obtained base patterns. For Medical dataset, the larger the space between parameters of θ and σ is, the more significantly the obtained number of base patterns will be increased. Meanwhile, the number of the generated initial candidate base pattern peaked when $\theta = 0.6$ and $\sigma = 0.15$, meaning that dynamic base pattern search strategy should avoid searching candidate base patterns to reduce running time and memory consumption since more "false" base patterns occurred when the value of θ is larger. What's more, the number of the generated base pattern reached to the highest point when $\theta = 0.348$ and $\sigma = 0.05$, indicating that this value is the optimum

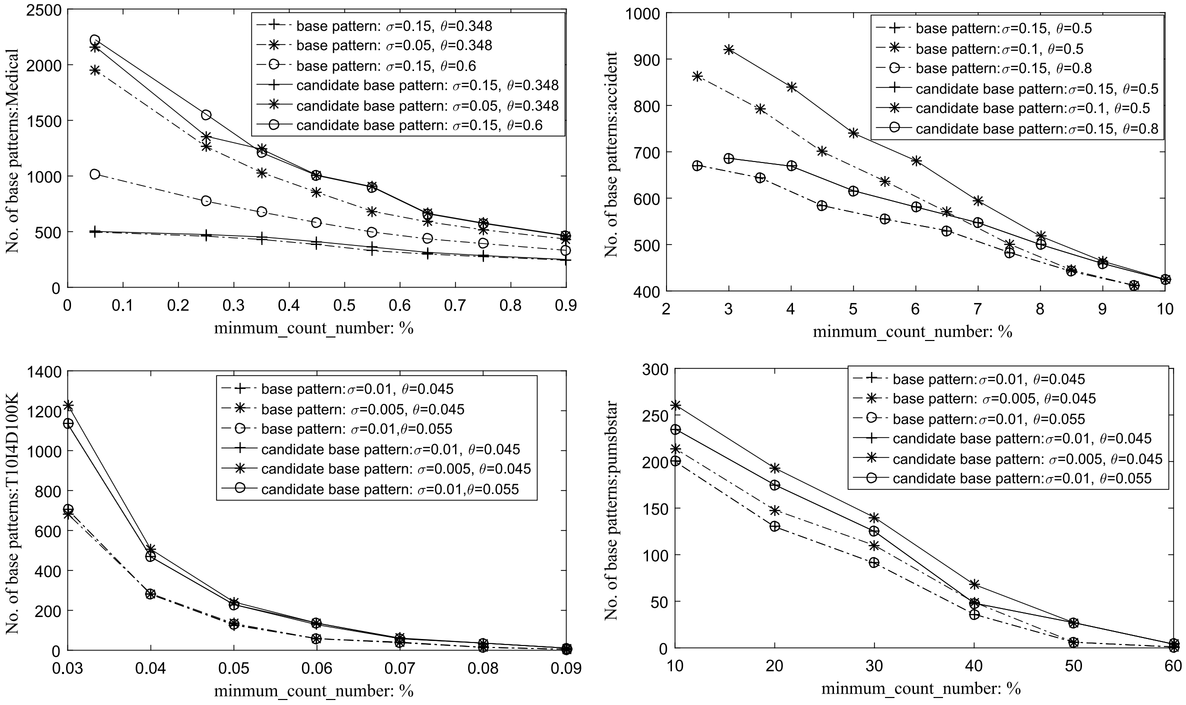


Fig. 2. Base patterns analysis for sparse dataset based on the proposed dynamic base pattern search strategy.

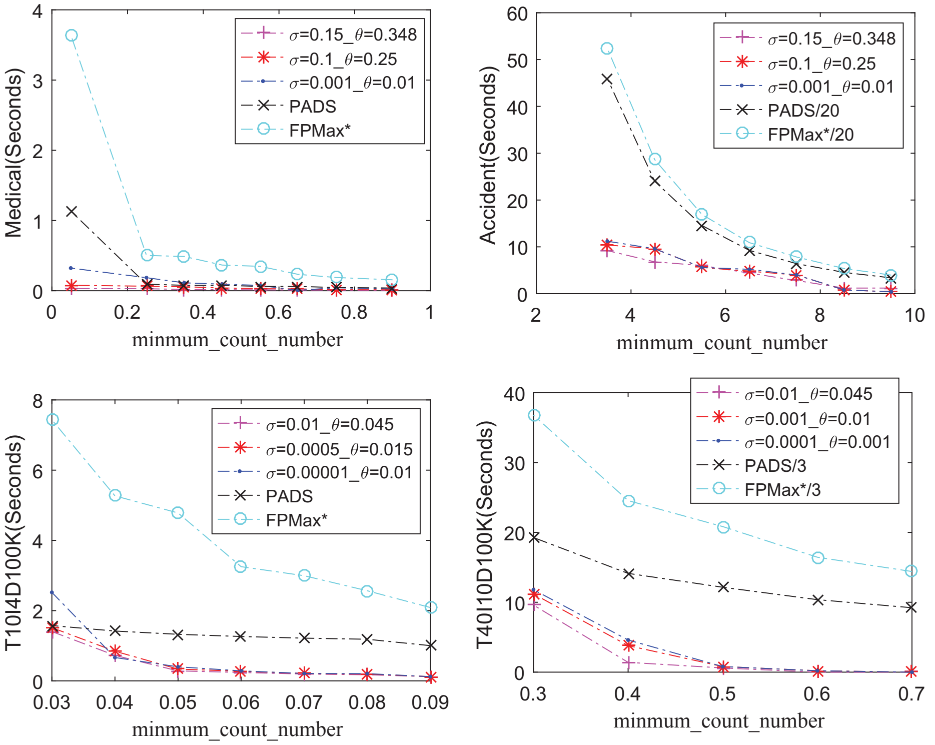


Fig. 3. Runtime results (Sparse Datasets).

suitable point that can be selected to mine MSFPs. Similarly, we can get the boundary line of optimum candidate base patterns extraction and optimum suitable mining point for the other datasets.The distribution of sparse datasets is relatively discrete which caused the fact that multiple probes have to be conducted to determine the base pattern. Moreover, the mining of frequent items, MFP, and CSP for sparse datasets is facing more competitions because of the uncertain regularity and the discrete distribution. The dynamic base pattern search strategy can effectively determine the regular characteristics and be capable to efficiently locate the optimum point of generating candidate base patterns.

5.3. Runtime analysis

In these runtime experiments, the threshold of setting the minimum support of base items (θ), and minimum support of the connected items (σ) for different datasets are selected based on base patterns analysis in Section 5.2. For each dataset, three groups of range are set up to determine the number of base items. The ***minmum_count_number*** is set the same for all algorithms, and the ***core_count_number*** is gradually increased in a different level on the basis of ***minmum_count_number***. The runtime of MFSFP mining takes in weighted frequent patterns extraction, pattern-aware dynamic base pattern searching, and MFSFP searching. The runtime results are shown in Fig. 3.

Generally, based on the experimental performance in Fig. 3, the MFSFP-Tree algorithm presents the fastest runtime performance under all parameters in all cases. Moreover, the proposed algorithm has the lowest runtime increments as the ***minmum_count_number*** been gradually decreased since the proposed dynamic base pattern search strategy and array technique contribute to reduce runtime of the MFSFP-Tree mining, while the existing algorithm show relatively high runtime increases when support value becomes low. When the transaction scale is larger and the support value is lower, the gap between the proposed algorithm and the others becomes larger.

5.4. Memory usage analysis

We have selected four representative datasets to conduct the memory usage experiments, which is

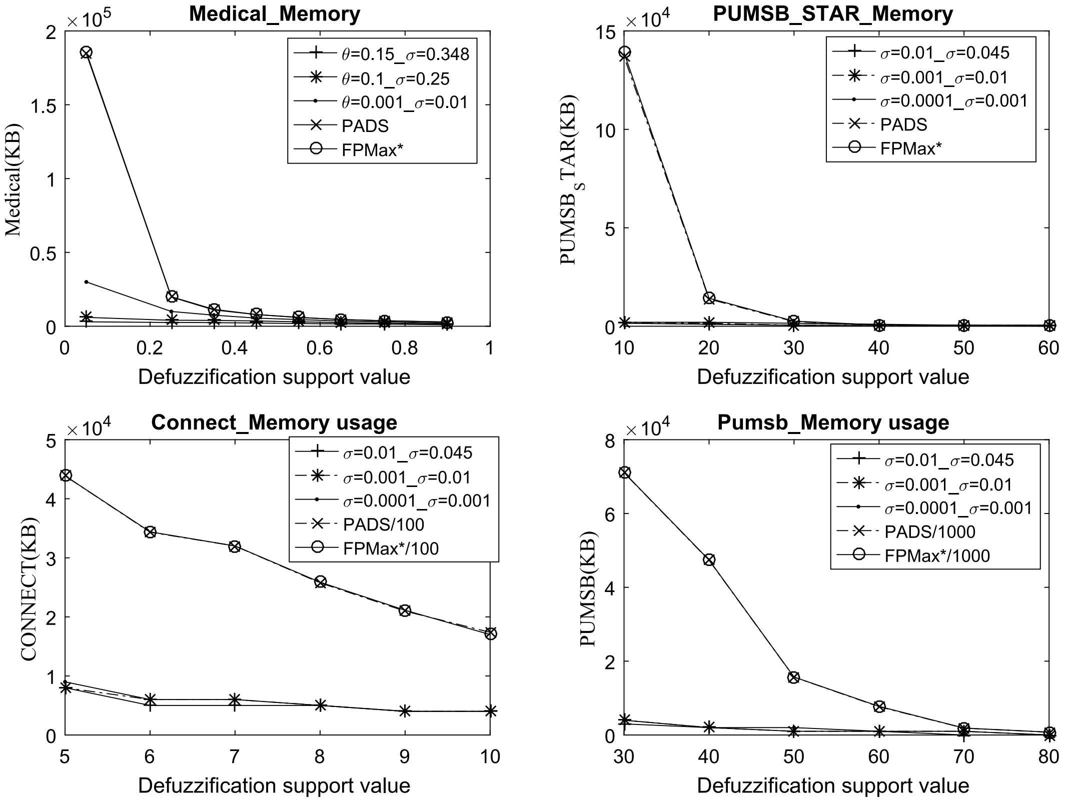


Fig. 4. Memory usage results.

shown in Fig. 4. The proposed dynamic base pattern search strategy and array technique have made a great contribution to reduce memory usage. The proposed algorithm is the most outstanding in memory efficiency based on all of the datasets analysis. The memory usage of PADS and FPMax* are very similar since these two algorithms adopt FP-tree as the major structure. However, the memory usage has an enormous gap among the proposed algorithm, PADS, and FPMAX*, thus, we changed the scale by shrinking the results of PADS and FPMAX* in varying degree to allow the memory usage results emerge together and better represent the comparison results among all algorithms.

Similarly with runtime consumption, the proposed algorithm has the lowest memory usage increments as ***minmum_count_number*** been gradually decreased and yet the memory usage of existing algorithm dramatically increased with this threshold decrease, and more memory consumption is required when the range θ and σ is getting larger in the proposed algorithm. The proposed algorithm possesses high experimental performance that the consumption is significantly lower than the existing algorithms in all cases. The memory consumption of mining global tree is greatly affected by conditional databases and the corresponding terminate steps in mining valid patterns, which reveals that the memory usage has strong relations with the size of datasets and the internal associated structure that represented by the joined percentage of items in each pattern.

Overall, the experimental results clearly show the power of dynamic base pattern search strategy and array technique pruning strategy in the proposed algorithm. The proposed algorithm conducts less subpattern checking by using the proposed pruning strategies in pruning subtrees to ensure better scheduling the candidate patterns.

6. Conclusion

Advanced pattern mining is vitally important in hidden information discovery and information proper representation from datasets. In this paper, we developed a new base-(second-order-effect) pattern structure, conducted FSFP-Tree, and proposed pruning strategies including pattern-aware dynamic base pattern search strategy and FSFP-array technique. The proposed pruning strategies and the new adopted fuzzy weight constraints and properties guarantee that the proposed maximal FSFPs mining algorithm mining is more efficient, which leads to conduct mining operations more quickly and effectively. In order to analyze the environmental performance of MFSFPs mining, we conduct extensive experiments in terms of obtained base patterns analysis, runtime analysis, and memory usage analysis, which clearly show that the proposed algorithm outperforms PADS and FPMax* algorithms. The obtained number and quality of MFSPs has shown that the proposed algorithm is more suitable to handle the combination of relative frequent items and relative lower frequent items.

In the future work, we will compare the discovered diseases extracted by MFSPs that have represented the relations of relative frequent diseases and relative lower frequent diseases with the clinical data to evaluate the effectiveness of the new proposed MFSFP pattern in medical perspective. In knowledge discovery perspective, it is very useful to explore how the proposed base-(second-order-effect) pattern structure and MFSFP mining can be extended to other advanced pattern mining tasks, and it is also interesting to study the inherent properties of the new structure.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (NSFC) (No. 61602064, No. 61772091), Science and Technology Agency Project of Sichuan Province (No. 2017HH0088), and Scientific Research Foundation of CUIT (No. KYTZ201615). We would like to thank the editor and reviewers for their helpful remarks.

References

[1] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, *in: Proc 20th Int Conf Very Large Data Bases, VLDB*, Vol. 1215, 1994, PP. 487–499.

[2] J. Han, J. Pei, Y. Yin and R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, *Data Mining and Knowledge Discovery* **8**(1) (2004), 53–87.

[3] M. Muzammal and R. Raman, Mining sequential patterns from probabilistic databases, *Knowledge and Information Systems* **44**(2) (2015), 325–358.

[4] J. Wang, J. Han, Y. Lu and P. Tzvetkov, TFP: An efficient algorithm for mining top-k frequent closed itemsets, *IEEE Transactions on Knowledge and Data Engineering* **17**(5) (2005), 652–663.

[5] B. Vo, F. Coenen and B. Le, A new method for mining Frequent Weighted Itemsets based on WIT-trees, *Expert Systems with Applications* **40**(4) (2013), 1256–1264.

[6] G. Fang, G. Pandey, W. Wang, M. Gupta, M. Steinbach and V. Kumar, Mining low-support discriminative patterns from dense and high-dimensional data, *IEEE Transactions on Knowledge and Data Engineering* **24**(2) (2012), 279–294.

[7] R.V. Priya, A. Vadivel and R.S. Thakur, Maximal pattern mining using fast CP-tree for knowledge discovery, *International Journal of Information Systems and Social Change (IJISSC)* **3**(1) (2012), 56–74.

[8] L. Chang, T. Wang, D. Yang, H. Luan and S. Tang, Efficient algorithms for incremental maintenance of closed sequential patterns in large databases, *Data & Knowledge Engineering* **68**(1) (2009), 68–106.

[9] U. Yun, G. Lee and K.H. Ryu, Mining maximal frequent patterns by considering weight conditions over data streams, *Knowledge-Based Systems* **55** (2014), 49–65.

[10] X. Zeng, J. Pei, K. Wang and J. Li, PADS: A simple yet effective pattern-aware dynamic search method for fast maximal frequent pattern mining, *Knowledge and Information Systems* **20**(3) (2009), 375–391.

[11] G. Grahne and J. Zhu, Fast algorithms for frequent itemset mining using fp-trees, *Knowledge and Data Engineering, IEEE Transactions on* **17**(10) (2005), 1347–1362.

[12] J. Zhang, Y. Wang and D. Yang, CCSpan: Mining closed contiguous sequential patterns, *Knowledge-Based Systems* **89** (2015), 1–13.

[13] H. Zhang, A. Sekhari, Y. Ouzrout and A. Bouras, Jointly identifying opinion mining elements and fuzzy measurement of opinion intensity to analyze product features, *Engineering Applications of Artificial Intelligence* **47** (2016), 122–139.

[14] Y. Han, Z. Geng, Q. Zhu and Y. Qu, Energy efficiency analysis method based on fuzzy dea cross-model for ethylene production systems in chemical industry, *Energy* **83** (2015), 685–695.

[15] H. Zhang, A. Sekhari, Y. Ouzrout and A. Bouras, Deriving consistent pairwise comparison matrices in decision making methodologies based on linear programming method, *Journal of Intelligent and Fuzzy Systems* **27**(4) (2014), 1977–1989.

[16] D. Burdick, M. Calimlim and J. Gehrke, A Maximal Frequent Itemset Algorithm for Transactional Databases, *in: Proc of the 17th Int'l Conf on Data Engineering*, 2000, pp. 443–452.

[17] K. Gouda and M. Zaki, Efficiently mining maximal frequent itemsets, *in: Data Mining, 2001 ICDM 2001, Proceedings IEEE International Conference on, IEEE*, 2001, pp. 163–170.