Journal of Universal Computer Science, vol. 22, no. 6 (2016), 802-821
submitted: 30/11/15, accepted: 28/5/16, appeared: 1/6/16 © J.UCS

Using Soft Set Theory for Mining Maximal Association Rules in Text Data

Bay Vo
(Faculty of Information Technology, Ho Chi Minh City University of Technology
Ho Chi Minh, Viet Nam
bayvodinh@gmail.com)

Tam Tran
(Tuy Hoa Industrial College, Tuy Hoa, Viet Nam
tranthidangtam@tic.edu.vn)

Tzung-Pei Hong
(National University of Kaohsiung, Kaohsing City, Taiwan, R.O.C
and
National Sun Yat-sen University, Kaohsiung City, Taiwan, R.O.C
tphong@nuk.edu.tw)

Nguyen Le Minh[*]
(Division of Data Science, Ton Duc Thang University, Ho Chi Minh, Viet Nam
Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh, Viet Nam
School of Information Science, Japan Advanced Institute of Science and Technology
nguyenleminh@tdt.edu.vn)

Abstract: Using soft set theory for mining maximal association rules based on the concept of frequent maximal itemsets which appear maximally in many records has been developed in recent years. This method has been shown to be very effective for mining interesting association rules which are not obtained by using methods for regular association rule mining. There have been several algorithms developed to solve the problem, but overall, they retain weaknesses related to the use of memory as well as mining time. In this paper, we propose an effective strategy for maximal rules mining based on soft set theory that consists of the following steps: 1) Build tree Max_IT_Tree where each node contains maximal itemsets *X*, the category of *X*, the set of transactions in which *X* is maximal, and the support of the maximal itemsets *X* for each category. 2) From the tree Max_IT_Tree built in previous steps, build a tree Max_Item_IT_Tree so that each maximal itemset has child nodes where each node contains items with categories different from the category of maximal itemsets. 3) Generate maximal association rules which satisfy predefined minimum *M-support* (min M-sup) and minimum *M-confidence* (min M-conf) thresholds.

Keywords: association rule, data mining, maximal association rule, soft set, text mining
Categories: I.2, I.2.1, I.2.7, I.2.8, M.1

[*] Corresponding author

1 Introduction

In data mining, association rule mining has been successfully applied in many fields. However, there are many complex problems in economics, engineering, environment, social science, medical science, etc., involving data that are not always all crisp. There are three theories: the theory of probability, the theory of fuzzy sets [Zadeh, 65], and interval mathematics [Goralazayny, 87], which we can consider as mathematical tools for dealing with uncertainties. However, all these theories have their own difficultie. The theory of probabilities can deal only with stochastically stable phenomena. Without going into mathematical details, we can say, e.g., that for a stochastically stable phenomenon, there should exist a limit of the sample mean in a long series of trials. To test the existence of the limit, we must perform a large number of trials. We can do it in engineering, but we cannot do it in many economic, environmental, or social problems. Currently, fuzzy set theory is progressing rapidly. However, there is a difficult issue which involves determining a method by which to set the membership function of each particular case. Interval mathematics methods are not sufficiently adaptable for problems with different uncertainties [Molodtsoy, 99]. Therefore, Molodtsov [Molodtsoy, 99] initiated the concept of soft set theory as a mathematical tool to deal with uncertainty.

In recent years, studies on soft set theory have achieved significant progress, including using the foundation of soft set theory [Maji, 03], soft set theory applied to support decision making [Maji, 02], parameterization reduction of a soft set and its application [Chen, 05]. In soft set theory, the initial description of an object has an approximate nature, and we do not need to introduce the notion of an exact solution. Because there are not any restrictions on the approximate description in soft set theory, this theory is very convenient and easily applicable in practice.

While common association rules are based on the notion of frequent itemsets or frequent closed itemsets [Han, 00] [Lucchese, 06] [Pasquier 99] [Agrawal, 94] [Vo, 11] [Vo, 13] [Vo, 14] [Zaki, 04], set of attributes which appear in many records and maximal association rules are based on frequent maximal itemsets. Frequent maximal itemsets are a set of attributes that appears alone or maximally in many records. Here, what is new that is maximal association rule mining allows the mining of interesting association rules that will not be obtained using the method for regular association rule mining [Han, 00] [Lucchese, 06] [Pasquier, 99] [Agrawal, 94] [Vo, 11] [Vo, 13] [Vo, 14].

There have been many authors who have studied maximal association rule mining and its application. Bi et al. applied rough set theory for maximal association rule mining in a collection of text documents and proposed alternative strategies for assuming taxonomy - a taxonomy existing for collections of labeled documents [Bi, 03]. Guan et al. proposed methods for maximal association rule mining using rough set theory [Guan, 12] [Guan, 13]. Their proposed approach is based on a partition on the set of all attributes in a transaction database, a so-called taxonomy and category of items. Herawan and Deris [Herawan, 11] directly applied a soft set on the Boolean valued information system for association rule mining, and based on the concept of co-occurrence and maximal co-occurrences of parameters in a transaction, these authors also defined the concept of regular association rules and maximal association rules between two set of parameters as well as a regular support, regular

confidence and maximal support and maximal confidence of the rules, respectively. The authors also pointed out that rules based on this method are similar to those methods in [Bi, 2003] [Guan, 03] [Guan, 05]. However, the algorithm suggested by authors still splits the problem maximal rule mining into two phases:

1) Find all frequent maximal itemsets (left-hand side) and sub-databases consisting of the projection on category T_i of the transactions M-support *X*.
2) Find the right hand side of maximal association rules based on the above sub-databases, and generate maximal association rules. In this paper, we suggest travering the database once to build the tree Max_Item_IT_Tree, then, find the frequent itemsets and generate maximal association rules in the tree Max_Item_IT_Tree at the same time.

The contribution of this paper is as follows: use soft set theory for maximal association rule mining in text data by scanning the database only once to build a tree Max_Item_IT_Tree for which the structure of the tree is as described in Section 3; then, find frequent itemsets on this tree based on diffset strategy. To do this, we perform the following steps: 1) scan database to build the first tree Max_IT_Tree (we call level 1 of the tree Max_Item_IT_Tree), where each node contains maximal itemset X, category of X, the set of transactions where X is maximal, and the support of X for each category, and the second tree Item_IT_Tree, where each node contains item Y, a category of Y, a set of transactions that contain Y, and support of Y. This means that we have found the left-hand side of maximal association rules (the set of the maximal itemsets which are contained in the tree Max_IT_Tree). 2) Find the right-hand side of the rules by using the two trees Max_IT_Tree and Item_IT_Tree built in the previous step, and then build a tree Max_Item_IT_Tree (we call level 2 of the tree Max_Item_IT_Tree); then, find frequent itemsets on the tree Max_Item_IT_Tree and generate rules which satisfy the min M-Sup and min M-Conf thresholds. Tree traversal is D*epth-first* traversal. Every branch is traversed, and then the frequent itemsets are updated; rules are generated, and the branch is deleted. This saves memory when building the tree Max_Item_IT_Tree.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 presents the proposed method consisting of the tree structure, details for the main algorithm, procedures used in the main algorithm, and an example to illustrate the algorithm. Section 4 describes the experimental results. Section 5 presents conclusions and suggestions for future work.

2 Related Work

2.1 Association Rules and Maximal Association Rules

2.1.1 Association Rules

Let $I = \{i_1, i_2, \dots, i_{|A|}\}$ for $|A| > 0$ refers to the set of literals called a set of items, and the set $D = \{t_1, t_2, \dots, t_{|U|}\}$, for $|U| > 0$, refers to the transaction database, where each transaction $t \in D$ is a set of distinct items $t = \{i_1, i_2, \dots, i_{|M|}, 1 \leq |M| \leq |A|\}$, and each transaction can be identified by a distinct identifier TID. An itemset with

$k-item$ is called a $k-itemset$. The support of an itemset X, denoted $sup(X)$, is defined as a number of transactions contain X. An association rule between sets X and Y is an implication of the form $X \rightarrow Y$, where $X \cap Y = \emptyset$. The itemsets X and Y are called the antecedent and consequent, respectively. The support of an association rule $X \rightarrow Y$, denoted $\sup(X \rightarrow Y)$, is defined as a number of transactions in D that contain $X \cup Y$. The confidence of an association rule $X \rightarrow Y$, denoted $cfi(X \rightarrow Y)$, is defined as a ratio of the number of transactions in D that contain $X \cup Y$ to the number of transactions in D that contain X. Thus, $cfi(X \rightarrow Y) = \frac{\sup(X \rightarrow Y)}{\sup(X)}$.

A large number of association rules can be found from a transaction database. To find the interesting association rules in a transaction database, we must define a specified minimum support (called $minsup$) and specified minimum confidence (called $minconf$). The itemset $Y \subseteq I$ is called a frequent itemset if $sup(X) \geq \min sup$. It is known that a subset of any frequent itemset is a frequent itemset; a superset of any infrequent itemset is not a frequent itemset. Finally, the association rule $X \rightarrow Y$ holds if $conf(X \rightarrow Y) \geq \min conf$.

2.1.2 Taxonomy and Category [Herawan, 11]

Let $I = \{i_1, i_2, \ldots, i_{|A|}\}$ be a set of items. A taxonomy T of I is a partition of I into disjoint subsets, i.e., $T = \{T_1, T_2, \ldots, T_n\}$. Each member of T is called a category. For an item i, we denote $T(i)$ as the category that contains i. Similarly, if X is an itemset all of which are from a single category, then we denote this category by $T(X)$.

Example 1. There is a database consisting of the 10 transactions [Herawan, 11]; 2 articles refer to Countries ‘‘Canada, Iran, USA” and topics ‘‘crude, ship”; 1 article refers to ‘‘USA” and ‘‘earn”; 2 articles refer to ‘‘USA” and ‘‘jobs, cpi”; 1 article refers to ‘‘USA” and ‘‘earn, cpi”; 1 article refers to ‘‘Canada” and ‘‘sugar, tea”; 2 articles refer to ‘‘Canada, USA” and ‘‘trade, acq”, and 1 article refers to ‘‘Canada, USA” and ‘‘earn”. The transactions are shown in Table 1.

TID	Items
1	Canada, Iran, USA, crude, ship
2	Canada, Iran, USA, crude, ship
3	USA, earn
4	USA, jobs, cpi
5	USA, jobs, cpi
6	USA, earn, cpi
7	Canada, sugar, tea
8	Canada, USA, trade, acq
9	Canada, USA, trade, acq
10	Canada, USA, earn

Table 1: The transaction database from [Herawan, 11]

We can create a taxonomy based on Table 1, which contains two categories: ‘‘countries” and ‘‘topics”, i.e., T = {countries, topics}, where countries = {Canada; Iran; USA} and topics = {crude; ship; earn; jobs; cpi; sugar; tea; trade, acq}.

2.1.3 Maximal Association Rules

The concept of maximal association rules was introduced by [Feldman, 97]. This method is a variant of association rules, which are designed to mining many interesting association rules hidden in a database that cannot be obtained by using the regular association rules. It allows the discovery of association rules relating to items that most often do not appear alone, but rather appear together with closely related items, and hence associations relevant only to these items tend to obtain low confidence when use regular association rules.

Feldman et al. (1997) noted that maximal association rules are not designed to replace regular association rules, but rather to complement them. Every maximal association rule is also a common association, but the support and confidence can vary [Feldman, 97]. While association rules are based on the notion of frequent itemsets which appear in many records, maximal association rules are based on frequent maximal itemsets which appear maximally in many records [Feldman, 97]. Using only maximal association rules, many interesting common associations can be lost.

The initial step to discover maximal rules is a partition on the set of items from a transaction database using a so-called taxonomy and categorization of items. To illustrate the notion of maximal association rules, let us consider the ideas which are quoted directly from [Amir, 05]. Maximal association rule $X \overset{max}{\longrightarrow} Y$, that is, whenever X appears alone then Y also appears. For this, we must first define the notion for each category $T_i \in T$ as follows:

For a transaction t, a category T_i and an itemset $X \subseteq T_i$, we say that X is alone in t if t $\cap\, T_i = X$. That is, X is alone in t if X is the largest subset of T_i which is in t. In this case, we can also say that X is maximal in t and that $t\ M-supports\ X$. For a database D, the $M-support$ of X in D, denoted $S_D^{max}(X)$ is the number of transaction $t \in D$ that M-support X.

A maximal association rule or M-association rule is a rule of the form $X \overset{max}{\longrightarrow} Y$, where $X \subseteq T(X)$, and $Y \subseteq T(Y)$. The $M-support$ of the M-association rule $X \overset{max}{\longrightarrow} Y$, denoted by $S_D^{max}(X \overset{max}{\longrightarrow} Y)$ is defined as

$$S_D^{max}\left(X \overset{max}{\longrightarrow} Y\right) = |\{t: t\ M-supports\ X\ and\ t\ supports\ Y\}|$$

That is, $S_D^{max}\left(X \overset{max}{\longrightarrow} Y\right)$ is the number of transactions in D that $M-support\ X$ and also support Y in the regular sense. The intuitive meaning of the M-association rule $X \overset{max}{\longrightarrow} Y$ is that whenever a transaction M-supports X, then Y also appears in the transaction, with some probability. However, in measuring this probability, we are only interested in those transactions where some elements of $T(Y)$ (the category of Y) appear in the transaction. Accordingly, the maximal confidence is defined as follows:

Let $D(X, T(Y))$ be the subset of the database D consisting of all the transactions that M-support X and contain at least one element of $T(Y)$. The confidence of the M-association rule $X \overset{max}{\longrightarrow} Y$, denoted by $C_D^{max}(X \overset{max}{\longrightarrow} Y)$ is defined as:

$$C_D^{max}\left(X \overset{max}{\longrightarrow} Y\right) = \frac{S_D^{max}(X \overset{max}{\longrightarrow} Y)}{|D(X,T(Y))|}.$$

As in regular association rule, to find the interesting association rules in a transaction database, we must define a specified $min\ M - sup$ and specified $min\ M - conf$. The maximal itemset $X \subseteq I$ is called a frequent maximal itemset if $S_D^{max}(X) \geq min\ M - sup$. In this case, a subset of any frequent maximal itemset is not necessarily a frequent maximal itemset.

Note that in the definition of maximal association rule where the antecedent is maximal, the consequent need not be maximal. Thus, a maximal rule $X \overset{max}{\longrightarrow} Y$ says that if X appears alone, then Y also appears, not necessarily alone. We note that alternative definitions are also possible, i.e. requiring maximality for both sides, or just for the consequent [Feldman, 97] . Any of these alternative definitions would constitute a mathematically valid definition. In this paper, we chose association rules where the antecedent is maximal but where the consequent need not be maximal.

2.2 Soft Set Theory

Molodtsov [Molodtsoy, 99] defined a soft set as follows: Let U be an initial universe set, and E be the set of parameters. Set $P(U)$ be the power set of U, and $A \subset E$.

Definition 1 [Herawan, 11]:

A pair *(F,E)* is called a soft set over U, where F is a mapping given by

$$F: E \longrightarrow P(U).$$

In other words, a soft set over U is a parameterized family of subsets of the universe U. For $e \in E$, $F(e)$may be considered as the set of $e - elements$ of the soft set (F, E) or as the set of $e - approximate$ elements of the soft set. Clearly, a soft set is not a (crisp) set. To illustrate this idea, let we consider the following example:

Example 2: Suppose that
U is the set of houses which is being considered.
E is a set of parameters.
E = {Expensive, beautiful, wooden, cheap, in green surroundings}.

In this case, to define a soft set means to point out expensive houses, beautiful houses, and so on. A soft set *(F, E)* describes the attractiveness of the houses which Mr. X is going to buy.

Suppose that there are six houses which are being considered in the universe U,

$$U = \{h_1, h_2, h_3, h_4, h_5, h_6\}, \text{ and}$$
$$E = \{e_1, e_2, e_3, e_4, e_5\}$$

is a set of decision parameters, where e_1 stands for the parameters "expensive", e_2 stands for the parameters "'beautiful", e_3 stands for the parameters ''wooden", e_4 stands for the parameters ''cheap", e_5 stands for the parameters ''in green surroundings".

Suppose that:

$$F(e_1) = \{h_2, h_4\}; F(e_2) = \{h_1, h_3\};$$
$$F(e_3) = \{h_3, h_4, h_5\}; F(e_4) = \{h_1, h_3, h_5\}; F(e_5) = \{h_1\}.$$

Therefore, $F(e_1)$ means ''houses (expensive)", whose functional value is the set $\{h_2, h_4\}$. Thus, we can view the soft set $(F,)$ as a collection of approximations as below:

$$(F,E) = \left\{ \begin{array}{c} expensive\ houses = \{h_2, h_4\} \\ beautiful\ houses = \{h_1, h_3\} \\ wooden\ houses = \{h_3, h_4, h_5\} \\ cheap\ houses = \{h_1, h_3, h_5\} \\ houses\ in\ green\ surroundings = \{h_1\} \end{array} \right\}$$

Each approximation has two parts; a predicate p and an approximate value set v. For example, for the approximation "'$expensive\ houses = \{h_2, h_4\}$", we have the predicate name of $expensive\ houses$, and the approximate value set or value set is $\{h_2, h_4\}$.

Thus, a soft set $(F,)$ can be viewed as a collection of approximations below:

$$(F,E) = \{p_1 = v_1, p_2 = v_2, p_3 = v_3, \dots, p_n = v_n\}.$$

2.3 Soft Set Theory for Association Rule Mining

2.3.1 Taxonomy and Categorization Using Soft Set Theory [Herawan, 11]

Let $(F,)$ be a soft set over the universe U. A taxonomy T of E is a partition of E into disjoint subsets, i.e., $T = \{E_1, E_2, \dots, E_n\}$. Each member of T is called a category. For an item i, we denote $T(i)$ the category that contains i. Similarly, if X is an itemset all items from a single category, then we denote this category by $T(X)$.

2.3.2 Maximal Association Rule Mining

Herawan and Deris (2011) used an approach for association rule mining [Herawan, 11]; this approach is started by a transformation of a transaction database into a soft set; then they define the notions of support, confidence of regular association rules and maximal support, maximal confidence of maximal association rules by using the concept of parameter co-occurrence in a transaction. The advantage of this method is that the execution is faster than the method proposed in [Amir, 05], the weakness of the approach is that it only obtains association rules with both the left and right hand side being maximal and is time-consuming in regard to transforming a transaction database into a soft set.

Rajpoot et al. (2012) proposed an efficient approach for association rule mining based on a soft set [Rajpoot, 12]. In this approach, the authors used constraint support that can filter out rarely occurring items. Due to the deletion of these items, the structure of the database is improved, and the result is produced more quickly, more accurately and uses less memory than the previous approach [Herawan, 11]. After the deletion of these items, the improved database is transformed into a Boolean-valued information system. Since the ''standard" soft set deals with such information system, a transaction database can be represented as a soft set. Using the concept of parameters co-occurrence in a transaction, they defined the notion of common

association rules between two sets of parameters, as well as their support and confidence by using soft set theory. The weakness of their approach is that it is time consuming to traverse the database to calculate the support for each item and to delete items which rarely occur, and to transform a transaction database into a soft set database.

3 The Proposed Method

IT-Tree-based method is an efficient method for frequent itemset mining [Zaki, 04], based on the intersection of the tidset to determine the support of frequent itemsets fast. Therefore, the algorithms based on IT-tree only scan the database once. Besides, we may also use the diffset to reduce the storage space (compared to tidset). The algorithms do not generate candidates, so mining efficiency is usually higher than the algorithms that generate candidates.

Applying the diffset strategy, in this section, we propose a method for association rule mining based on the tree Max_Item_IT_Tree with a tree structure as follows:

- Level 1 of the tree contains the nodes in which each node contains the following information: maximal itemsets *X*, the category of *X*, the set of transactions in which *X* is maximal, the support of X for each category. Because each node of this level contains maximal itemsets, we denote this level by the tree Max_IT_Tree.
- Level 2 of the tree contains item *Y* which has a category different from the category of this maximal itemset, category and the set of transactions that contain *X* and *Y*. We denote this level by the tree Max_Item_IT_Tree.
- The other levels of the tree contain frequent itemsets which are generated by the diffset strategy with a few changes.

In summary, each node in the tree contains the following information that depend on the level of the tree:

- The maximal itemsets (max) or itemset (itemset).
- The list of transactions that contain maximal itemsets or the list of transactions that contain items (Tidset).
- Maximal itemset category or itemset category (Category).
- An array contains support of the maximal itemsets for each category or the support of the itemsets.
- List of child nodes (children).

The approach is both intended to find frequent itemsets and to generate rules, which satisfies the min M-sup and min M-conf thresholds. Tree traversal is D*epth-first* traversal. Every branch is traversed then the frequent itemsets are updated; the rules are generated, and the branches are deleted. This saves memory when building the tree Max_Item_IT_Tree.

3.1 Algorithm

The algorithm (see Figure 1) uses the following procedures:

Build_Max_IT_Tree(D, min_M_Sup): Build two trees Max_IT_Tree and Item_IT_Tree to support the tree building of the Max_Item_IT_Tree in the next step. Each node of the tree Max_IT_Tree contains maximal itemset X (max), the category of maximal itemset X (category), a set of transactions in which X is maximal and a support of maximal itemset X for each category. Each node of the tree Item_IT_Tree contains an item (item), an item category, a list of transactions (tidset), and support of this item (sup). At that time, level 1 of the tree Max_Item_IT_Tree is completed.

Input: database D, min_M_Sup, min_M_Conf threshold, taxonomy T, Categories T_i
Output: maximal association rules.
Approach to implement:
1. Root_Max={};
2. Roo_Item={};
3. Build_Max_IT_Tree(D, min_M_Sup);
4. Build_Max_Item_IT_Tree(min_M_Sup, min_M_Conf);
5. Find_Frequent_2Item(Root_Max, min_M_Sup, min_M_Conf);
6. Find_ Frequent_nItem(Root_Max, min_M_Sup, min_M_Conf);

Figure 1: Algorithm for maximal association rule mining.

Build_Max_Item_IT_Tree(min_M_Sup, min_M_Conf): From the trees Max_IT_Tree and Item_IT_Tree built in the previous step, we build a tree Max_Item_IT_Tree (level 2 of the tree Max_Item_IT_Tree) such that each maximal itemset has child nodes and for which each node contains:

- Items different from the maximal itemsets in the category.
- A set of transactions contains the maximal itemsets and the item.
- The support of the item.

Find_Frequent_2Item (Root_Max, min_M_Sup, min_M_Conf) and Find_ Frequent_nItem(Root_Max, min_M_Sup, min_M_Conf): From the trees Max_IT_Tree and the Item_IT_Tree built in the previous step, we perform both find frequent itemsets and generate rules based on the diffset strategy of IT-Tree with a few changes.

Details of the procedures are shown on Figures 1-5.

3.2 Illustration Example

Example 3: Consider the database as shown in Table 1, in which we have T={countries, topics}, T_1 = Countries = {Canada, Iran, USA} và T_2 = topics = {crude, ship, earn, jobs, cpi, sugar, tea, trade, acq}.

To make calculation convenient, we change parameters into the following symbols:

A: Canada; B: Iran; C: USA; D: crude; E: ship; F: earn; G: jobs; H: cpi; I: sugar; J: tea; K: trade; L: acq;

The database is converted into Table 2.

Let min $M_{Sup} = 2$; min $M_{Conf} = 75\%$, we have done the following steps:

Step 1: procedure **Build_Max_IT_Tree** builds two trees: Max_IT_Tree (level 1 of tree Max_Item_IT_Tree) and Item_IT_Tree to use for building the tree Max_Item_IT_Tree in next step.

- Consider transactions t_1 = {A, B, C, D, E}
  + $t_1 \cap T_1$ = {A,B,C} $\Rightarrow$ {A,B,C} is maximal in t_1.
  + {A,B,C} belong to category 1 $\Rightarrow$ sup[1] = 0
  + do $t_1 \cap T_2$= {D,E} $\neq \varnothing \Rightarrow$ sup[2] =1

```
Build_Max_IT_Tree(D, min_M_Sup)
1. For all t_i ∈ D do
2.    For all T_j∈ T do //build tree Max_IT_Tree
3.       X=t_i∩T_j;
4.       If X ≠ ∅ then
5.          If (X does not exist in the tree) then
6.             For all T_k ∈ T (k≠j) do
7.                Y = t_i∩T_k;
8.                If Y ≠ ∅ then a[k]=1;
9.                   Root_Max.AddChild(New node(X,i,j,a));
10.               Else
11.                  Find node p contains maximal itemsets X in Root_Max;
12.                  p.Tidset = p.Tidset ∪ i;
13.                  For all T_k ∈ T(k≠j) do
14.                     Y=t_i∩T_k;
15.                     If Y ≠ ∅, then p.Sup[k] = p.Sup[k]+1;
16.   For each item ∈ t_i do //build tree Item_IT_Tree
17.      If (item does not exist in the tree) then
18.         Find category l of item
19.         Root_Item.AddChild(new node(item,i,l,1);
20.      Else
21.         Find node q contains item in Root_Item;
22.         q.Tidset = q.Tidset ∪ i;
23.         q.Sup = q.Sup +1;
24. End.
```

Figure 2: The algorithm for building the trees Max-IT-Tree and Item_IT_Tree

```
Build_Max_Item_IT_Tree(min_M_Sup, min_M_Conf)
1. For each node p ∈ Root_Max do
2.     If (p.Tidset.Count ≥ min_M_Sup) then
3.         For each node q ∈ Root_Item do
4.             If (p.Category !=q.Category) then
5.                 intersection = p.Tidset∩q.Tidset;
6.                 If (intersection.Count ≥ min_M_Sup) and
                     ((intersection.Count/p.Sup[q.Category])≥ min_M_Conf) then
7.                   p.AddChild(new node(q.Itemset, intersection,q.Category,
                       new int[] {intersection.Count}));
8.                   Generate rule (p.Max → q.Itemset);
9.     Else delete p;
```

Figure 3: The Algorithm for tree building Max_Item_IT_Tree

```
Find_Frequent_2Item(Root_Max, min_M_Sup, min_M_Conf)
1. For each node p ∈ Root_Max do
2.     For each node q_i ∈ p do
3.         For each node q_j ∈ p (with j>i) do
4.             If (q_i.Category = q_j.Category), then
5.                 subtract = q_i.Tidset – q_j.Tidset;
6.                 Itemset = q_i.Itemset ∪ q_j.Itemset;
7.                 Sup = q_i.Sup[0] – subtract.Count;
8.                 Conf = Sup/p.Sup[q_i.Category];
9.                 If (Sup≥Min_M_Sup) and (Conf ≥Min_M_Conf) then
10.                    q_i.AddChild(Itemset, subtract,qi.Category,new int[]{sup});
11.            Generate rule (p.Max → Itemset) with Sup and Conf.
12. End.
```

Figure 4: The algorithm for finding frequent itemsets that have two items.

```
Find_ Frequent_nItem(Root_Max, min_M_Sup, min_M_Conf)
1. If (Root_Max = ∅) then return;
2. For each node ∈ Root_Max do
3.     Sup_Parent = node.Sup[node.Category];
4.     Find_Frequent(node, Sup_Parent, min_M_Sup,Min_M_Conf);
5.     Delete node;
Find_Frequent (node, Sup_Parent, min_M_Sup,Min_M_Conf)
6. For each node1 ∈ node do
7.     For each node q_i ∈ node1 do
8.         For each node q_j ∈ node1 do (with j>i) do
9.             If (p_i.Category = p_j.Category) then
10.                subtract =p_j.Tidset – p_i.Tidset;
11.                Itemset= p_i.Itemset ∪ p_j.Itemset;
12.                Sup = p_i.Sup[0] – subtract.Count;
13.                If (Sup≥min_M_Sup) and (Sup/Sup_Parent≥min_M_Conf) then
14.                    p_i.AddChild(Itemset, subtract,p_i.Category,new int[] {Sup});
15.                    Generate rule (node.Max →Itemset);
16.    Find_Frequent (node1, Sup_Parent, min_M_Sup, Min_M_Conf);
17. End;
```

Figure 5: The algorithm for mining frequent itemsets that have n-items (n > 2).

Add node which contains informations such as: maximal itemset {A,B,C}, category of {A,B,C} and support of {A,B,C} for each category in tree Max_IT_Tree.

+ $t_1 \cap T_2$={D,E} ⇒ {D,E} is maximal in t_1
+ {D,E} belong to category 2 ⇒ sup[2] = 0
+ do $t_1 \cap T_1$= {A,B,C} ≠ ∅ ⇒ sup[1] =1

Add node that contains maximal itemset {D,E}, category of {D,E} and support of {D,E} for each category in the tree Max_IT_Tree.

TID	Items
1	A,B,C,D,E
2	A,B,C,D,E
3	C,F
4	C,G,H
5	C,G,H
6	C,F,H
7	A,I,J
8	A,C,K,L
9	A,C,K,L
10	A,C,F

Table 2: Converted database

- Consider transaction t_2= {A,B,C,D,E}
  + $t_2 \cap T_1 = \{A,B,C\} \Rightarrow \{A,B,C\}$ is maximal in t_2.
  + do $t_2 \cap T_2 = \{D,E\} \neq \varnothing$ và {A,B,C} already exists in the tree ⇒ sup[2] +=1
  + $t_2 \cap T_2 = \{D,E\} \Rightarrow \{D,E\}$ is maximal in t_2
  + do $t_2 \cap T_1 = \{A,B,C\} \neq \varnothing$ và {D,E} already exists in the tree ⇒ sup[1] +=1

- Do the same with other transactions, we have tree Max_IT_Tree with root Root_Max as it is shown on Figure 6.

The confidence of the M-association rule $X \xrightarrow{max} Y$, denoted by $C_D^{max}(X \xrightarrow{max} Y)$ is defined as:

$$C_D^{max}\left(X \xrightarrow{max} Y\right) = \frac{S_D^{max}(X \xrightarrow{max} Y)}{|D(X,T(Y))|},$$

Where $D(X,T(Y)$ is the subset of the database D consisting of all the transactions that M-support X and contain at least one element of T(Y) (categoryof Y).

Therefore, we must store support of the maximal itemsets for each category. Because there are some cases that transaction is M-support X but does not contain any element of T (Y).

The tree Item_IT_Tree is built similarly to the tree IT-Tree with root Root_Item – see Figure 7.

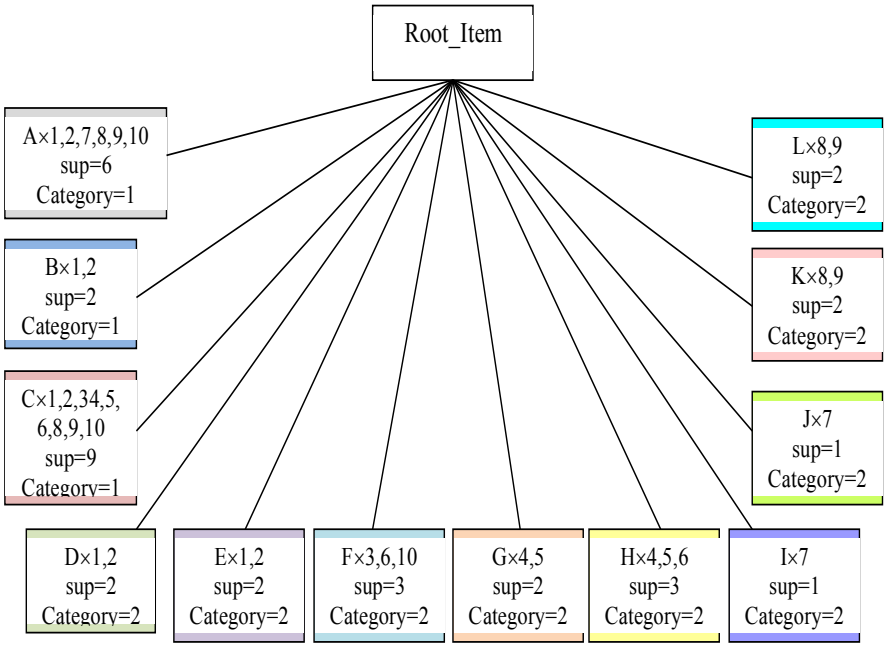

Figure 6: An example illustrates the construction of the tree Max_IT_Tree

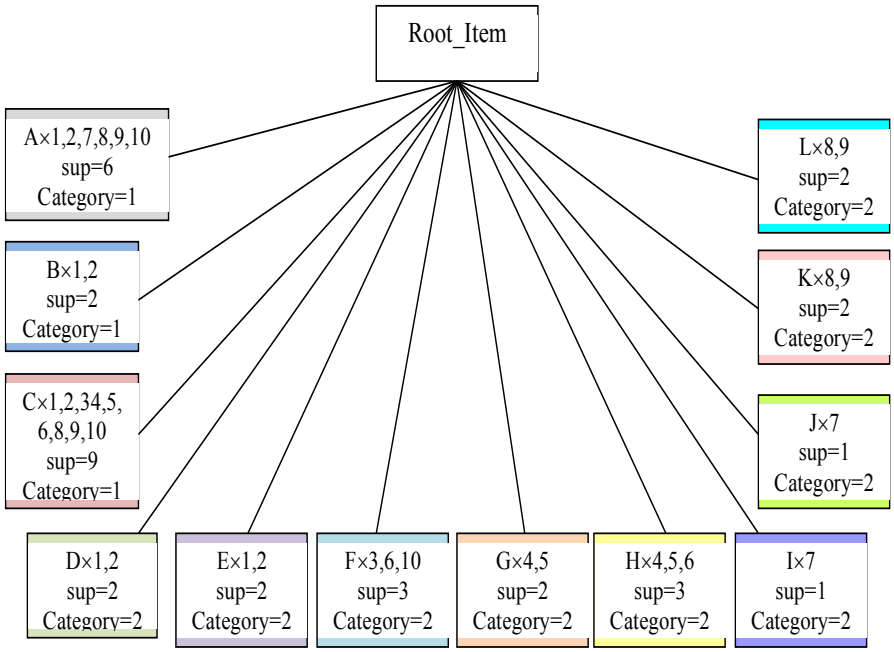

Figure 7: An example illustrates the construction of the tree Item_IT_Tree

Step 2: Build tree Max_Item_IT_Tree as follows:

- Consider maximal itemset {A,B,C}.

  + Traversal of the tree Item_IT_Tree, item {D} that differs from maximal itemsets {A,B,C} in the category.

  + Calculate intersection,
  tidset=tidset({A,B,C})∩tidset({D})={1,2}∩{1,2}={1,2}

  + The number of elements of the intersection is the support of frequent itemset {D} and is the support of rule {A,B,C}→{D}.

  + Add node {D} to the list of child nodes of {A, B, C} because it satisfies the min M-sup and min M-conf thresholds.

  + Do the same with the other items in the tree Item_IT_Tree.

- Do the same with other nodes, which contain other maximal itemset such as {D, E}, {C}, {F}, {G, H}, {A, C}, {K, L}.

In this step, we both build the tree and generate rules; deleted the branches that do not satisfy the min M_sup and min M_Conf thresholds, such as {F,H}, {A}, {I,J}.

We have level 2 of the tree as follows:

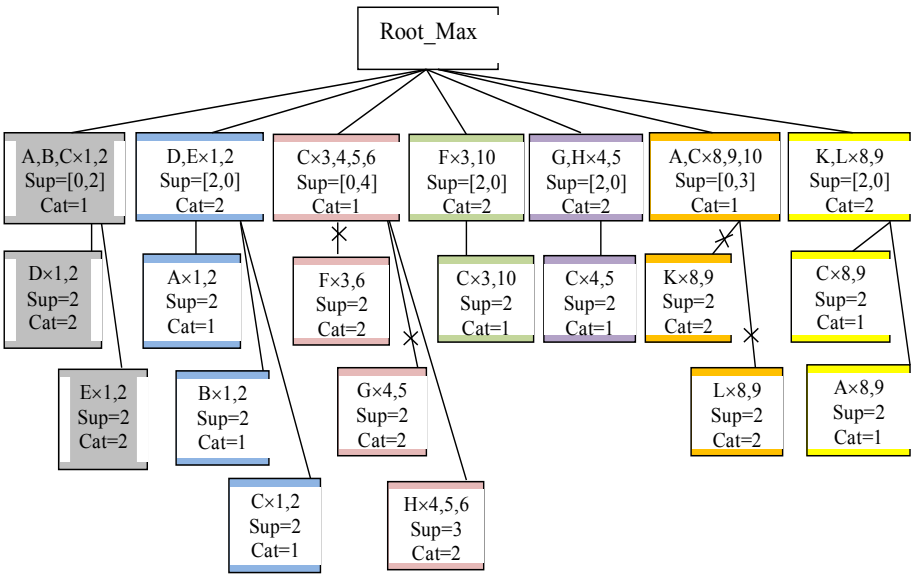

Figure 8: An Example illustrates the construction of the tree Max_Item_IT_Tree

We have the following rules:

$Canada, Iran, USA \rightarrow Crude\ (sup = 2, conf = 2/2 = 100\%)$
$Canada, Iran, USA \rightarrow Ship\ (sup = 2, conf = 2/2 = 100\%)$
$Canada, Iran, USA \rightarrow Crude, Ship\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Canada\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Iran\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Canada, Iran\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Canada, USA\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Canada, Iran, USA\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Iran, USA\ (sup = 2, conf = 2/2 = 100\%)$
$USA \rightarrow cpi\ (sup = 3, conf = 3/4 = 75\%)$
$earn \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$jobs, cpi \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$Trade, acq \rightarrow Canada\ (sup = 2, conf = 2/2 = 100\%)$
$Trade, acq \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$Trade, acq \rightarrow Canada, USA\ (sup = 2, conf = 2/2 = 100\%)$

Step 3: From the tree Max_Item_IT_Tree built in the previous step, we find the frequent itemsets based on the diffset strategy. We have the following results:

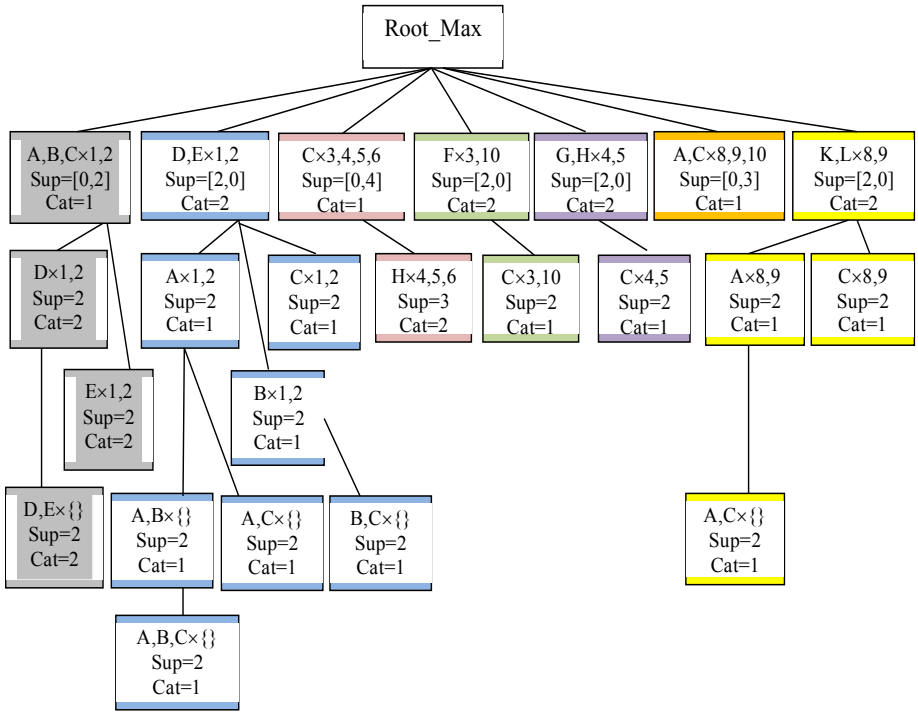

Figure 9: An Example illustrates for findingfrequent itemsets

In the process of finding frequent itemsets, branches are traversed and then deleted. In this step, we both build the tree and generatethe rules. We have the following rules:

$$A,B,C \rightarrow D,E\ (sup = 2, conf = 2/2 = 100\%).$$
$$D,E \rightarrow A,B\ (sup = 2, conf = 2/2 = 100\%).$$
$$D,E \rightarrow A,C\ (sup = 2, conf = 2/2 = 100\%).$$
$$D,E \rightarrow A,B,C\ (sup = 2, conf = 2/2 = 100\%).$$
$$D,E \rightarrow B,C\ (sup = 2, conf = 2/2 = 100\%).$$
$$K,L \rightarrow A,C\ \left(sup = 2, conf = \frac{2}{2} = 100\%\right).$$

So we obtain the following rules:

$$Canada, Iran, USA \rightarrow Crude\ (sup = 2, conf = 2/2 = 100\%)$$
$$Canada, Iran, USA \rightarrow Ship\ (sup = 2, conf = 2/2 = 100\%)$$
$$Canada, Iran, USA \rightarrow Crude, Ship\ (sup = 2, conf = 2/2 = 100\%)$$
$$Crude, Ship \rightarrow Canada\ (sup = 2, conf = 2/2 = 100\%)$$
$$Crude, Ship \rightarrow Iran\ (sup = 2, conf = 2/2 = 100\%)$$
$$Crude, Ship \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$$
$$Crude, Ship \rightarrow Canada, Iran\ (sup = 2, conf = 2/2 = 100\%)$$

$Crude, Ship \rightarrow Canada, USA\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Canada, Iran, USA\ (sup = 2, conf = 2/2 = 100\%)$
$Crude, Ship \rightarrow Iran, USA\ (sup = 2, conf = 2/2 = 100\%)$
$USA \rightarrow cpi\ (sup = 3, conf = 3/4 = 75\%)$
$earn \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$jobs, cpi \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$Trade, acq \rightarrow Canada\ (sup = 2, conf = 2/2 = 100\%)$
$Trade, acq \rightarrow USA\ (sup = 2, conf = 2/2 = 100\%)$
$Trade, acq \rightarrow Canada, USA\ (sup = 2, conf = 2/2 = 100\%)$

4 Experimental Resuls

In this section, we compare the proposed approach with the algorithm for mining maximal association rules of [Amir, 05]. All the algorithms are executed sequentially on a processor Intel core i3, 3×2.27, RAM 2GB and are implemented in the C # programming language (2008). The experimental database is obtained from a collection of labeled documents used for text classification Reuters-21578. The database consists of 21578 transactions (records) [1]. After eliminating empty transactions, 19716 remains transactions are stored in the SQL server 2005.

The initial step to discover maximal rules is a partition on the set of items from a transaction database into so-called taxonomy and categorization of items. In this experiment, we partition the original set of items into taxonomy *T* and four categories specifically:

$$T = \{T_1, T_2, T_3, T_4, T_5\},$$

where: T_1 = Countries (places)

T_2 = Topics
T_3 = People
T_4 = Orgs
T_5 = Exchanges.

Comparison results for the mining time between the algorithm in [Amir, 05] and the proposed algorithm are shown in Table 3. The algorithm in [Amir, 05] obtained all rules for which the left hand side is maximal and the right hand side can either be maximal or not (the algorithm proposed by us also obtained such rules).

The results in Table 3 clearly indicate that the method for tree building the Max_Item_It_Tree is faster than the method in [Amir, 05] with respect to all the values (min M_sup and min M_conf) described in Table 3. Table 3 also shows that both algorithms obtained the same maximal association rules. For the alogirthm described in [Amir, 05], the mining time increases when decreasing the min M_sup. However, the mining time of our proposed algorithm is little changed. Besides, the change of min M_conf does not affect in the mining time.

[1] Source: https://archive.ics.uci.edu/ml/datasets/Reuters21578+Text+Categorization+Collection

min M_sup	min M_conf	number of Maximal association rules	Mining time (s)	
			The algorithm in [Amir et al., 2005]	The proposed algorithm
10	0.8	75	25.06	18.86
10	0.7	92	25	18.82
10	0.6	116	25.05	18.8
7	0.8	111	26.56	18.91
7	0.7	133	26.68	18.87
7	0.6	166	26.66	18.96
5	0.8	181	29.01	19.16
5	0.7	216	29.09	19.04
5	0.6	256	29.28	19.07
3	0.8	344	33.22	19.38
3	0.7	412	33.31	19.57
3	0.6	487	33.45	19.48

Table 3: Comparison results for the mining time.

The results in Table 3 clearly indicate that the method for tree building the Max_Item_It_Tree is faster than the method in [Amir, 05] with respect to all the values (min M_sup and min M_conf) described in Table 3. Table 3 also shows that both algorithms obtained the same maximal association rules. For the alogirthm described in [Amir, 05], the mining time increases when decreasing the min M_sup. However, the mining time of our proposed algorithm is little changed. Besides, the change of min M_conf does not affect in the mining time.

5 Conclusions and Future Work

This paper presented an approach to apply soft set theory for maximal association rule mining from transaction databases. The approach traverses the database once for building the Max_Item_IT_Tree and generating maximal association rules. We defined the notions of support and confidence of maximal association rules based on soft set theory. In addition, we developed methods through the collection of a labeled standard database for text mining Reuters-21578. The obtained rules are exactly the same method for maximal association rule mining proposed in [Amir, 05]. However, the mining time of our approach is faster than the previous method.

In the future, some other methods for mining association rules using soft set will be discussed. In addition, we will study how to apply this method for weighted maximal association rule mining in transaction databases. Finally, class association rule mining has also been proposed in recent years [Nguyen, 14] [Nguyen, 15a][Nguyen, 15b], we will apply soft set theory into mining class association rules.

Another interesting point is how to exploit linguistic constraints for mining association rule in text application. The asscoation rule mining technique descibed in

[Duong, 15] can be applicable for this purpose. We let this research as one of open problem in our future work.

Acknowledgments

This research was funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2015.10.

References

[Agrawal, 94] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. VLDB'94, 1994, 487–499.

[Amir, 05] Amir, A., Aumann, Y., Feldman, R., Fresco, M.: Maximal association rules: A tool for mining associations in text. Journal of Intelligent Information Systems 25(3), 2005, 333–345.

[Bi, 03] Bi, Y., Anderson, T., McClean, S.:. A rough set model with ontologies for discovering maximal association rules in document collections. Knowledge-Based Systems 16, 2003, 243–251.

[Chen, 05] Chen, D., Tsang, E.C.C., Yeung, D.S., Wang, X.: The parameterization reduction of soft sets and its applications. Computers and Mathematics with Applications 49, 2005, 757–763.

[Duong, 15] Duong, H.V., Truong, T.C.: An efficient method for mining association rules based on minimum single constraints. Vietnam Journal Computer Science 2, 2015, 67-73.

[Feldman, 97] Feldman, R., Aumann, Y., Amir, A., Zilberstein, A., Klosgen, W.: Maximal association rules: a new tool for mining for keywords cooccurrences in document collections. The Proceedings of the KDD-1997, 1997, 167–170.

[Goralazayny, 87] Gorzalzany, M.B.: A method of inference in approximate reasoning based on interval-valued fuzzy sets. Fuzzy Sets and Systems 21, 1987, 1-17.

[Guan, 03] Guan, J.W., Bell, D.A., Liu, D.Y.: The rough set approach to association rule mining. The Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), 2003, 529–532.

[Guan, 05] Guan, J.W., Bell, D.A., Liu, D.Y.: Mining association rules with rough sets. Studies in Computational Intelligence, Springer Heidelberg, 2005, 163–184.

[Han, 00] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In Proceedings SIGMODKDD'00, 2000, 1–12.

[Herawan, 11] Herawan, T., Deris, M.M.: A soft set approach for association rule mining. Knowledge-Based Systems 24, 2011, 186-195.

[Herawan, 09] Herawan, T., Yanto, I. T. R., Deris, M.M.: Soft set approach for maximal association rules mining. Springer-Verlag Berlin Heidelberg, 2009, 163-170.

[Lucchese, 06] Lucchese, B., Orlando, S., Perego, R.: Fast and memory efficient mining offrequent closed itemsets. IEEE Transaction on Knowledge and Data Engineering, 18(1), 2006, 21–36.

[Molodtsoy, 99] Molodtsov D.: Soft set theory-First results. Computers and Mathematics with Applications 37, 1999, 19-31.

[Maji, 03] Maji, P.K., Biswas, R., Roy, A.R.: Soft set theory. Computers and Mathematics with Applications 45, 2003, 555-562.

[Maji, 02] Maji, P.K., Roy, A.R.: An Application of Soft Sets in a Decision Making Problem. Computers and Mathematics with Applications 44, 2002, 1077-1083.

[Nguyen, 14] Nguyen, D., Vo, B., Le, B.: Efficient strategies for parallel mining class association rules. Expert Systems with Applications, 41(10),2014, 4716-4729.

[Nguyen, 15a] Nguyen, D., Nguyen, L.T.T., Vo., B., Hong, T.P.: A novel method for constrained class association rule mining. Information Sciences, 320, 2015, 107-125.

[Nguyen, 15b] Nguyen, L.T.T., Nguyen, N.T.: An improved algorithm for mining class association rules using the difference of Obidsets. Expert Systems with Applications, 42(9), 2015, 4361-4369.

[Pawlak, 82] Pawlak, Z.: Rough sets. International Journal of Information and Computer Sciences 11, 1982, 341-356.

[Pawlak, 91] Pawlak, Z.: Rough sets: A theoretical aspect of reasoning about data. Kluwer Academic Publisher, Dordrecht, 1991.

[Pasquier, 99] Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules using Closed Itemset Lattices. Information Systems, 24(1), 1999, 25–46.

[Rajpoot, 12] Rajpoot, V., Shrivastava, S.K., Mathur, A.: An Efficient Constraint Based Soft Set Approach for Association Rule Mining. International Journal of Engineering Research and Applications (IJERA), 2012, 2210-2215.

[Vo, 11] Vo, B., Le, B.: Interestingness measures for mining association rules: Combination between lattice and hash tables. Expert Systems with Applications, 38(9), 2011, 11630–11640.

[Vo, 13] Vo, B., Hong, T.P., Le, B.: A Lattice-based Approach for Mining Most Generalization Association Rules. Knowledge-Based Systems, 45, 2013, 20–30.

[Vo, 14] Vo, B., Hong, T.P., Le, B.: An effective approach for maintenance of pre-large-based frequent-itemset lattice in incremental mining. Applied Intelligence, 41(3), 2014, 759-775.

[Zaki, 04] Zaki, M.J.: Mining non-redundant association rules. Data Mining and Knowledge Discovery, 9(3), 2004, 223–248.

[Zadeh, 65] Zadeh, L.A.: Fuzzy sets. Information and Control 8, 1965, 338-353.