Journal of Computer Science and Cybernetics, V.30, N.4 (2014), 397–408
DOI: 10.15625/1813-9663/30/4/4020

IMPROVE EFFICIENCY OF FUZZY ASSOCIATION RULE USING HEDGE ALGEBRA APPROACH

TRAN THAI SON[1], NGUYEN TUAN ANH[2]

[1]*Institute of Information Technology, Vietnam Academy of Science and Technology; trn˙thaison@yahoo.com*
[2]*University of Information and Communication Technology, Thai Nguyen University; anhnt@ictu.edu.vn*

Abstract. A major problem when conducting mining fuzzy association rules from the database (DB) is the large computation time and memory needed. In addition, the selection of fuzzy sets for each attribute of the database is very important because it will affect the quality of the mining rule. This paper proposes a method for mining fuzzy association rules using compressed database. We also use the approach of Hedge Algebra (HA) to build the membership function for attributes instead of using the normal way of fuzzy set theory. This approach allows us to explore fuzzy association rules through a relatively simple algorithm which is faster in terms of time, but it still brings association rules which are as good as the classical algorithms for mining association rules.

Keywords. Data mining, association rules, compressed transactions, knowledge discovery, hedge algebras

1. INTRODUCTION

In recent years, the fast development of technologies has made the collecting and storing abilities of information systems quickly increase. Moreover, the computerization of the production, sales and many other activities has created a huge amount of data needed for storage. There have been so many very large databases among millions of records used in the aforementioned activities. This boom has led to an urgent demand that is necessary to apply new techniques and tools in order to extract huge amounts of data to useful knowledge. Therefore, data mining techniques have attracted a great deal of attention in the field of information technology.

Mining association rules have been under active research and have brought many good results [1–4]. The authors have come up with many solutions to reduce the time taken to exploit the rules, such as mining association rules in parallel, using compression solutions dealing with binary database. However, in this field, there are still many issues that need further investigation and resolution. Recently, the compression algorithm using binary data in the database to provide a good solution can reduce storage space requirements and data processing time. Jia-Yu Dai suggested an algorithm named M2TQT [5]. The basic idea of this algorithm is: adjacent transactions will be merged to form a new transaction. As a result, a new database which has the smaller size is created and can reduce the data processing time as well as the storage space. In [5], the experiment results showed that the M2TQT performed better than existing methods. However, this algorithm can just be applied to binary database.

Fuzzy data processing to explore the data in the fuzzy association rules is mainly based on the fuzzy set theory as shown in [1,2,6]. In the past, the algorithms using fuzzy set theory when building

the membership functions of attribute face many difficulties. However, people nowadays show more interest in this construction. If you build a strong FB (Fuzzy Baseset of membership functions), the next data mining hopes to bring the best results (shown in [7]). The construction of this function requires a satisfaction of several criteria:

1) The number of MFs per variable is moderate.

2) MFs are distinguishable, i.e. two MFs do not present the same or almost the same linguistic meaning.

3) Each MF is normal. An MF is normal if it has membership value 1 at least at one point of domain values

4) Domain values are strongly covered. At least one MF receives a membership value β (where $\beta > 0$) at any point of domain values.

For the fuzzy set theory, it is not entirely easy [8]. For HA, due to the linguistic variable values form a partition on the value domain, we can easily create membership functions on the basis of the following: likelihood of one element in a fuzzy set can be determined based on the distance from that element to the quantitative semantic value of the fuzzy set (where the fuzzy set is an element of HA, for example "young", "very old"..); the smaller the distance is, the greater the degree has. Methods in [9, 10] applying HA in solving the problem of mining the association rules have been proposed in order to overcome disadvantages of the fuzzy set theory. Specifically, to construct the membership function when using the fuzzy logic, the researchers determine the degree of membership of the value in the database instead of subjectively selecting a membership function (the form of an isosceles triangle is usually taken). However, HA approach selects the values of the database through distance values to quantified semantic value. Quantified semantic values are determined from the beginning when the parameters of HA are determined. The authors in [9] consider the range of values $\mathrm{Dom}(A)$ of fuzzy properties as a HA. Each $x \in \mathrm{Dom}(A)$ corresponds to an element y in HA (using the inverse function in HA). This method is simple, but such mapping may cause the information loss. The method in can solve this problem by determining the distance of x to quantitative semantic values of the two closest elements of x to both sides, and other elements are considered to zero. Therefore, each value of x gives us a pair of values to save instead of just one value.

To improve the efficiency of mining association rules, in this article we propose a new method of mining the fuzzy association rules based on the HA and using compressed transactions. With this approach, adjacent transactions are merged into a new transaction which can reduce the vertical size of input database. Experiments proved that this proposed method offers better results compared to other available methods.

The paper is organized as follows: The basic concepts of association rules and HA are reviewed in section 2; Mining fuzzy association rules based on HA; compressed database and the mining of fuzzy association rules according to compressed database are described in section 3; Result analysis in section 4 shows the performance of the proposed algorithm and fuzzy Apriori algorithm based on FAM95 database.

2. PRELIMINARIES

2.1. Association rules

Let $I = I_1, I_2, , I_m$ be a set of items. Let D, the task-relevant data, be a set of database transactions where each transaction T is a set of items, such is $T \subseteq I$. Each transaction is associated with an identifier, called *TID* [11].

Definition 2.1 **([4])** An association rule has the form of $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$.

Two important measures of association rule are support(s) and confidence(c) defined in [4].

Definition 2.2 **([4])** The support of association rule $X \Rightarrow Y$ is the probability that $X \cup Y$ exists in a transaction in the database D.

$$support(X \Rightarrow Y) = P(X \cup Y) = \frac{(n(X \cup Y))}{N} \tag{1}$$

Definition 2.3 **([4])** The confidence of the association rule $X \Rightarrow Y$ is the probability that $X \cup Y$ exists given that a transaction contains X, i.e.

$$confidence(X \Rightarrow Y) = P\left(\frac{X}{Y}\right) = \frac{(n(X \cup Y))}{n(Y)} \tag{2}$$

Where: $n(X)$ is the number of transactions, including X, N is the total of transaction database.

Mining the association rules of the database is finding all of the rules that have the degree of support and confidence greater than degree of support *Min_sup* and confidence *Min_conf* determined by the available user.

In fuzzy association rules, the degree of support of a fuzzy range s_k belonging to x_i is defined as follows:

$$FS(A_(s_k)^(x_i)) = \frac{1}{N}\sum_{j=1}^{N} \mu_{s_k}^{x_i}\left(d_j^{x_i}\right) \tag{3}$$

And the reliability of a fuzzy range s_1, s_2,..,s_k of items x_1, x_2, ...,x_k, respectively is:

$$FS\left(A_{s_1}^{x_1}, A_{s_2}^{x_2}, \ldots, A_k^{x_k}\right) = \frac{1}{N}\sum_{j=1}^{N} \min\left(\mu_{s_1}^{x_1}\left(d_j^{x_1}\right), \mu_{s_2}^{x_2}\left(d_j^{x_2}\right), \ldots, \mu_{s_k}^{x_k}\left(d_j^{x_k}\right)\right) \tag{4}$$

Where x_i is i^{th} item, s_j is fuzzy range belonging to item i^{th}, N is the total of transactions in the database, $\mu_{s_k}^{x_i}\left(d_j^{x_i}\right)$ is the membership degree of the value at the i^{th} column, row j into the fuzzy set s_k.

2.2. Hedge algebras

Let X be a linguistic variable and Xbe a set of its terms, called a term-domain of X. E.g. if X is the rotation speed of an electrical motor and linguistic hedges used to describe its speed are *Very*, *More*, *Possibly*, *Little*, denoted correspondingly for short by V, M, P and L, then $X = $ –*fast, V fast, M fast, L P fast, L fast, P fast, L slow, slow, P slow, V slow, ...*″ $\cup \mathbf{0}, W, 1$ is a term-domain of X. It

can be considered as an abstract algebra $AX = (X, C, H, \leq)$, where H is a set of linguistic hedges, which can be regarded as one-argument operations, $\leq$ is called a semantics-based ordering relation on X and $\boldsymbol{W}, 0, 1$ is a set of constants in X with *fast* and *slow* being primary terms of X and $\boldsymbol{W}, 0, 1$ being additional elements in X interpreted as the neutral, the least and the greatest ones, respectively. Denote by hx the result of applying an $h \in H$ to $x \in X$ and by $H(x)$ the set of all $u \in X$ generated algebraically from x by using hedges in H, i.e. $H(x) = u \in X : u = h_n \dots h_1 x, h_1, \dots, h_n \in H$. As pointed out in [12–15], the elements in terms-domain can be ordered, based on their meaning, which is expressed by means of a semantics-based relation by the following way (see [1, 9, 10]):

It is natural that there is a demand to transform fuzzy sets defined on a real interval $[a, b]$, which represents the meaning of terms in a term-domain X, into $[a, b]$ or, for normalization, into [0, 1]. This defines a mapping of the term-domain X into [0, 1], called in the algebraic approach a semantically quantifying mapping (SQM). Now, we take these mappings in mind to define a notion of *fuzziness measure*. Let us consider a mapping f from X into [0, 1], which *preserves* the ordering relation on X. Then, the "*size*" of the set $H(x)$, for $x \in X$, can be measured by the diameter of $f(H(x)) \subseteq [0, 1]$. That is that this *diameter* will be considered as a fuzzy measure of the term x.

Taking this model of fuzziness measure in mind, we may adopt the following definition:

Let $AX = (X, C, H, \leq)$ be a linear HA. An $fm : X \to [0, 1]$ is said to be a fuzzy measure of terms in X if:

fm1) $fm(c^-) + fm(c^+) = 1$ and $\sum_{h \in H} fm(hu) = fm(u)$, for all $u \in X$.

fm2) $fm(x) = 0$, for all x such that $H(x) = \{x\}$. Especially, $fm(\mathbf{0}) = fm(\boldsymbol{W}) = fm(\mathbf{1}) = 0$;

fm3) $\forall x, y \in X, \forall h \in H, \frac{fm(hx)}{fm(x)} = \frac{fm(hy)}{fm(y)}$, that is, it does not depend on specific elements and, therefore, is called the fuzziness measure of h, denoted by $\mu(h)$.

The condition in fm1) and fm2) is intuitively evident. fm3) seems also natural: the relative effect of h is the same, i.e. this proportion does not depend on the terms that h applies to.

The characteristics $fm(x)$ v $\mu(h)$ as following:

$$fm(hx) = \mu(h) fm(x), \forall x \in X, \tag{5}$$

$$\sum_{i=-q, i \neq 0}^{p} fm(h_i c) = fm(c), \text{ with } c \in \{c^-, c^+\}, \tag{6}$$

$$\sum_{i=-q, i \neq 0}^{p} fm(h_i x) = fm(x), \tag{7}$$

$$\sum_{i=-1}^{(} -q)\mu(h_i) = \alpha \text{ and } \sum_{i=1}^{p} \mu(h_i) = \beta, \text{ with } \alpha, \beta > 0 \text{ and } \alpha + \beta = 1. \tag{8}$$

Signal function: $\mathrm{Sign} : X \to \{-1, 0, 1\}$ is recursively defined as following [16]:

With $k, h \in H, c \in \{c^-, c^+\}, sign\,(c^+) = +1$ and $sign\,(c^-) = 1, \{h \in H^+ | sign\,(h) = +1\}$ and $\{h \in H^- | sign(h) = 1\}$.

$sign(hc) = +sign(c)$ if h is positive for c and

$sign(hc) = -sign(c)$ if h is negative for c. $sign(hc) = sign(h) \times sign(c)$

$sign(khx) = +sign(hx)$ if k is positive for $h (sign(k, h) = +1)$ and

$sign(khx) = -sign(hx)$ if k is negative for $h(sign(k,h) = +1)$
$\forall x \in H(G)$ can be written as $x = hm \ldots h1c$ with $c \in G$ and $h1, \ldots, hm \in H$. Then:

$$sign(x) = sign(hm, hm-1) \times \ldots \times sign(h2, h1) \times sign(h1) \times sign(c), \tag{9}$$

$$(sign(hx) = +1) \Rightarrow (hx \geq x) \text{ and } (sign(hx) = 1) \Rightarrow (hx \leq x). \tag{10}$$

Suppose that preset fuzzy measure of the hedges $\mu(h)$ and values of fuzzy measure of the generating elements $fm(c^-), fm(c^+)$ and θ is the neutral element.

The function of quantification semantics v of T is set up recursively as follows [16]:

$$v(W) = fm(c^-),\ v(c^-) = \theta - \alpha fm(c^-) = \beta fm(c^-),$$
$$v(c^+) = \theta + \alpha fm(c^+) = 1 - \beta fm(c^+) \tag{11}$$

$$v(h_j x) = v(x) + sign(h_j x)\{\sum_{i=sign(j)}^{j} fm(h_j) - \omega(h_j x) fm(h_j x)\} \tag{12}$$

$$\omega(h_j x) = \frac{1}{2}\left[1 + sign(h_j x) sign(h_p h_j x)(\beta - \alpha)\right] \in \{\alpha, \beta\},\ j \in \{[-q^p], j \neq 0\}$$

3. MINING FUZZY ASSOCIATION RULES BASED ON HEDGE ALGEBRA

In this section, we propose a new method of fuzzy database compression based on the HA approach. Transaction database is compressed based on the distance of transactions. Moreover, we build the quantification table in order to reduce the numbers of candidate itemsets. Finally, we propose a new algorithm of mining association rule based on compressed database.

3.1. Hedge algebra approach to the problem of association rules [9, 10]

On HA approach, the membership function values of each database value are calculated as shown below:

First, the attribute value of each fuzzy domain is regarded as a HA. Instead of building a membership function of the fuzzy set, a quantitative semantic value is used to determine the degree of membership value in any row in fuzzy sets defined above.

Step 1: Standardize values ??of the fuzzy attribute between [0, 1].

Step 2: Consider the fuzzy range s_j of the attribute x_i as an element of HA AX_i

Then, any value $d_j^{x_i}$ of x_i lies between any two quantification semantic values of 2 elements of AX_i and the distance between $d_j^{x_i}$ and quantification semantic value of the closest element to $d_j^{x_i}$ of the two sides may be to determine the closeness level of $d_j^{x_i}$ in the fuzzy range (two elements of that HA). Closeness level between $d_j^{x_i}$ and other elements of HA are determined as 0. In order to determine the last level of membership, we have to standardize (transfer of the value between $[0, 1]$, then we have 1 minus that standardized distance). We will have a pair of membership levels for each value $d_j^{x_i}$. In summary, we can determine the membership degree of the attribute x_i into the fuzzy range s_j as: $\mu_{s_j}(d_j^{x_i}) = 1 - |v(s_j) - d_j^{x_i}|$, with $v(s_j)$ is quantitative semantics value of the element S_j.

3.2. Relationship of Transaction Distance [5]

Based on the distance of transactions, we can merge the transactions which have the adjacent distance in order to form a transaction group; as a result, we have a new database with a smaller size.

The definition of transaction relationship and transaction distance relationship as below:

(1) Transactional relationship: The two transactions $T1$, $T2$ are considered to be related to each other if $T1$ is the subset of $T2$ or $T1$ is the superset of $T2$.

(2) Transactional distance relationship: Distance relationship between two transactions is the number of different items.

Example: Preset 3 transactions $T1 = \{B = 0.9; C = 0.86; D = 0.43\}$, $T2 = \{A = 0.65; C = 0.55; D = 0.75\}$, $T3 = \{A = 0.65; B = 0.23; C = 0.82; D = 0.94\}$, then, the distance between $T1$ and $T2$ is $D(T1-T2) = 2$, distance between $T2$ and $T3$ is $D(T2-T3) = 1$.

3.3. Quantification table

TID	**Transactions**
100	$\{A = 0.3; B = 0.2; C = 0.6; D = 0.2; E = 0.5;\}$
200	$\{C = 0.4; D = 0.7; E = 0.2;\}$
300	$\{A = 0.5; C = 0.3; D = 0.4;\}$

Table 1: Example of database transaction

To reduce the numbers of candidate itemsets, there should be more information to eliminate the itemset which is not frequent set. Quantification table is built to save this information when each transaction is under handling. The items appear in the transaction need to be sorted by lexicographical. First, we start at the left item and it is called the prefix of the item. After that, the length of the input transaction (n) is computed and the number of items taken note in the transaction depends on the length of the transaction: $TL_n, TL(n-1), \ldots, TL_1$. Quantification table includes of items, in which each TL_i contains one item prefix and its support value. Table 2 is the qualification table built for database in Table 1.

For example, transaction $TID = 100$ has the value $\{A = 0.3; B = 0.2; C = 0.6; D = 0.2; E = 0.5\}$. Transaction 100 has the length $n = 5$, with prefix A, value from TL_5 to TL_1, it is increased by 0.3 (at the beginning, it is 0). Therefore $A = 0.3$ appears in each TL_i, with $I = 5 \ldots 1$. With the prefix B, the value from TL_4 to TL_1, it is increased by 0.2 (at the beginning, it is 0), so $B = 0.2$ appears in each TL_i, with $I = 4 \ldots 1$. C, D and E are treated similarly. Then, transaction $TID = 200$ having the value of $\{C = 0.4; D = 0.7; E = 0.2\}$ is treated, qualification table has the value $C = 1.0$ in TL_3, TL_2,and TL_1; $D = 0.9$ in TL_2, TL_1; $E = 0.7$ in TL_1. With the last transaction $\{A = 0.5; C = 0.3; D = 0.4\}$, will increase the value from $A = 0.3$ to $A = 0.8$ in TL_3, TL_2, and TL_1; C=1 to C=1.3 in TL_2 and TL_1; $D = 0.9$ to $D = 1.3$ in TL_1.

TL_5	TL_4	TL_3	TL_2	TL_1
$A = 0.3$	$A = 0.3$	$A = 0.8$	$A = 0.8$	$A = 0.8$
	$B = 0.2$	$B = 0.2$	$B = 0.2$	$B = 0.2$
		$C = 1.0$	$C = 1.3$	$C = 1.3$
			$D = 0.9$	$D = 1.3$
				$E = 0.7$

Table 2: Quantification table for the database of Table 3.3.

3.4. Transaction database compression

Let d represent the relative distance relationship which is initialized to 1. Based on the distances between transactions, we merge all transactions with distances less than or equal to d in order to form a new transaction group.

Algorithm 1: Algorithm of compressed transaction

Input: Fuzzy transaction database

Output: Compressed database

The notations of parameters in the algorithm as follows:

3.5. Transaction database compression

Let d represent the relative distance relationship which is initialized to 1. Based on the distances between transactions, we merge all transactions with distances less than or equal to d in order to form a new transaction group.

Algorithm 1: Algorithm of compressed transaction

Input: Fuzzy transaction database

Output: Compressed database

The notations of parameters in the algorithm as follows:

$ML = \{ML_k\}$: ML_k The transaction group having the length k (the length of a transaction is the number of items in this transaction)

$L = \{L_k\}$: L_k Transaction with the length k

$T_i : i^{th}$ Transaction in fuzzy database

$|T_i|$: The length of transaction T_i

Step 1: Read one transaction T_i at a time from fuzzy database

Step 2: Computing the length of the transaction T_i

Step 3: Based on an input transaction, the qualification table is built.

Step 4: Computing the distance between transactions T_i and the transaction group in blocks ML_{n-1}, ML_n, ML_{n-1}. If there is an existence of a transaction group in the blocks ML_{n-1}, ML_n, ML_{n-1}, the distance to the transaction T_i will be less than or equal to d. Then the transaction T_i is merged into the relevant transaction group. The old transaction group will be removed.

For example, let $d = 1$ and two transactions $\{B = 0.23; C = 0.55; D = 0.75\}$ and $\{C = 0.82; D = 0.94\}$. Because the distance between these two transactions is 1, these two transactions merge into a new transaction group $\{B = 0.23; C = 1.37; D = 1.69\}$. This transaction group has the length of 3. Therefore, this transaction group is given to block ML_3. The sign "=" is used to present the total of membership degree of the items in the transaction group. With the transaction $\{B = 0.4; C = 0.5\}$, distance between $\{B = 0.23; C = 1.37; D = 1.69\}$ and $\{B = 0.4; C = 0.5\}$ is 1. Therefore, the transaction $\{B = 0.4; C = 0.5\}$ merges into the transaction $\{B = 0.23; C = 1.37; G = 1.69\}$ to form a new transaction group. The final transaction group becomes $\{B = 0.63; C = 1.87; G = 1.69\}$. The transaction group $\{B = 0.23; C = 1.37; G = 1.69\}$ is removed from the block ML_3 and the transaction group $\{B = 0.63; C = 1.87; G = 1.69\}$ is moved to the block ML_3

Step 5: If the transaction T_i is not merged with the transaction group in the blocks ML_{n-1}, ML_n, ML_{n+1}. Computing the distance between transactions T_i and transactions in the blocks L_{n-1}, L_n, L_{n+1}. If there is an existence of the transaction T_j so that $D_{T_i - T_j} \leq d$, merging the transaction T_i to the transaction T_j in order to form a new transaction group and add more this transaction group into respective blocks (depending on the length of the transaction group created), and remove the

transaction T_j in the blocks: L_{n-1}, L_n, L_{n+1}. If there is not an existence of any transaction satisfying the distance d, the transaction T_i will be classified to the block L_n.

Step 6: Repeat 5 above steps until the final transaction is read.

Step 7: Read one transaction T_i at a time from $L = \{L_k\}$

Step 8: Computing the length of the transaction $T_i : n$

Step 9: Computing the distance of the transaction T_i and transaction groups in the blocks ML_{n-1}, ML_n, ML_{n+1}. If there exists a group of transactions with distance less than or equal to the d, the transaction Ti would merge into the group to create a new transaction group. Based on the length of the new transaction group, we add this transaction group into the respective blocks: ML_{n-1}, ML_n, ML_{n+1}, remove the old transaction group in the blocks: ML_{n-1}, ML_n, ML_{n+1}, and remove the transaction T_i in the block L_n.

Step 10: Repeat the step 7, step 8 and step 9 until the final transaction in $L = \{L_k\}$ is read.

Finally, the obtained compressed database includes $L = \{L_k\}$, $ML = \{ML_k\}$ and quantification table.

3.6. Fuzzy association rules [9]

Algorithm 2: Fuzzy association rule based on compressed database

The notations of parameters of the algorithm as follows:

N	The total number of transactions in the database
m	The number of attribute
A_j	j^{th} attribute, 1≤j≤m
$\|A_j\|$	The number of HA labels of attribute A_j
R_{jk}	HA labels of attribute A_j, $1 \le k \le \|A_j\|$
$D^{(i)}$	i^{th} transaction database, $1 \le I \le N$
$v_j^{(i)}$	The value of A_j in $D^{(i)}$
$f_{jk}^{(i)}$	The value of membership degree of $v_j^{(i)}$ with HA label R_{jk}, $0 \le f_{jk} \le 1$
$Sup(R_{jk})$	The degree of support of R_{jk}
Sup	The value of support of each frequent ItemSet
Conf	Degree of correlation of each frequent ItemSet
Min_sup	The available minimum support value
Min_conf	Available reliability value
C_r	The set of candidate ItemSets with attribute r (ItemSets), $1 \le r \le m$
L_r	The set of frequent ItemSets is hedge label r (ItemSets), $1 \le r \le m$

The algorithm of mining database based on HA for quantitative value is carried out as follows:

Input: Transaction database D, hedge algebras for the fuzzy attribute, *Min_sup* and *Min_conf*

Output: Association rules

Step 1: Convert the quantitative value $v_j(i)$ of each transaction $D^{(i)}$, i from 1 to N. For each attribute A_j, if A_j is located beyond to one of two both ends (the two maximum and minimum hedge labels), there will be only one hedge label which agrees with that end; if not, A_j will be represented by two continuous hedge labels which have the smallest values in the field value of A_j, each label

with one of the values which is represented the membership degree $f_{jk}(i)$ $(j=1,2)$ of A_j with that HA. This membership degree is considered to be the distance between A_j and the value represented for the appropriate hedge label.

Step 2: Carry out the algorithm of compressed transactions (Algorithm 1) while the fuzzy database obtained in the step 1. As a result of this step, we have the compressed database and quantification table.

Similar to the Apriori algorithm, we apply the algorithm to the compressed database to create a frequent ItemSets.

Step 3: Based on the value in TL_1 of the quantification table, value in TL_1 is the support of R_{jk}. If $Sup(R_{jk}) \geq Min\_sup$, then R_{kj} is put into L_1.

Step 4: If $L_1 \neq \emptyset$, go to the next step; if $L_1 = \emptyset$, the algorithm is ended.

Step 5: The algorithm that builds the frequent itemset of level r from the frequent itemset of level $r-1$ by choosing 2 frequent itemsets of level $r-1$ when these 2 itemsets are different from each other in only one set. After joining these two itemsets, we have the candidate itemset C_r. Before using the compressed database to compute the support degree of itemsets in C_r, we can eliminate some candidates without revising compressed database, based on the value of TL_r in the quantification table.

Step 6: Approve compressed database basing on the formula (4) in order to compute the support degree of each itemset in C_r. If there is any itemset which has the support degree appropriate with minimum support, it is taken to L_r

Step 7: Follow the next steps and repeat *frequent* itemsets with greater levels, which are produced with form (or +1), the *frequent* itemset S with the item $(s_1, s_2, \ldots, s_t, \ldots, s_{r+1})$ in C_{r+1}, $1 \leq t \leq r+1$:

(a) According to the form (4), compute the support degree sup(S) of S in the transaction;

(b) If $Sup(S) \geq Min\_sup$, then S is taken to L_{r+1}.

Step 8: If L_{r+1} is null, then the next step is carried out; in contrast, propose $r = r+1$, step 6 and step 7 are repeated.

Step 9: Give the association rules from the collected *frequent* itemset as follows:

For each following feasible association rule: $s_1 \cap \ldots \cap s_x \cap s_y \cap \ldots \cap s_q \rightarrow s_k$ ($k=1$ to q, $x=k-1$, $y=k+1$). The confidence of the rule is computed by following formula:

$$Conf\left(s_1 \cap \ldots \cap s_x \cap s_y \cap \ldots \cap s_q \rightarrow s_k\right) = \frac{Sup(S/s_k)}{Sup(S)} \tag{13}$$

4. RESULT ANALYSIS

The proposed algorithm and the algorithm in [9] are tested by the C# programming language on a computer with detailed descriptions: Intel(R) Core(TM) i5 CPU 1.7GHz, RAM 6GB.

The source of the data is taken from FAM95 database, conducted by the Bureau of the Census for the Bureau of Labor Statistics in 1995. Within all attributes of the database, five are taken for testing purpose which includes Age, Hours, IncFam, IncHead, and Sex. Where, Age is the age of Head in years, Hours is the working hours per week, IncFam is family income, IncHead is Head's personal income, and Sex is the gender of Head. The Age, Hours, IncFam, and IncHead attributes are fuzzy attributes. The Sex attribute assigns the value of 0 for female or 1 for male. The number of records is 63565.

Duration for compressing the above database is 135 seconds. After compression, the number of transactions obtained is 2402. With 60% confidence, testing results on the two algorithms: Hedge

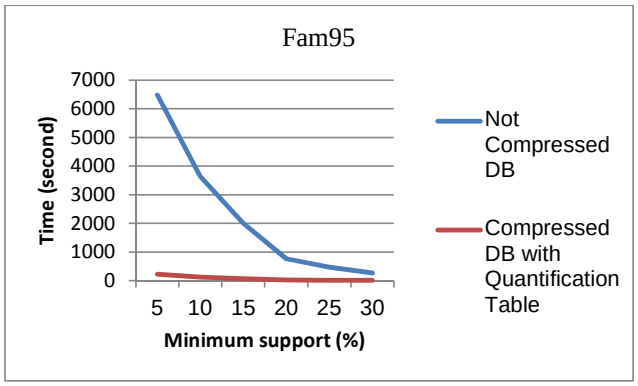

Figure 1: The experiment result of FAM95

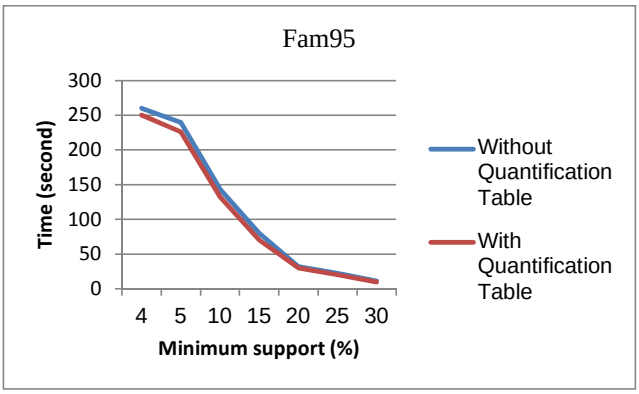

Figure 2: With and without using a quantification table

algebra based- fuzzy association rule method in [9] and Hedge algebra based- fuzzy compressed database method are shown in the graphs below. The computation results prove that our method offers a better result than the one in [9]. Moreover, the value of obtaining frequent itemsets is the same as itemsets without database compression in [9].

The dataset FAM95 is used to run our algorithm and the algorithm in [9]. Let the average size of the potentially large itemset be 5 for the minimum supports 5%, 10%, 15%, 20%, 25%, and 30%, and compare our algorithm with the algorithm in [9]. As a result, our algorithm's performance is much better. As shown in Figure 1, when the minimum support is 5%, the execution time of the algorithm without compressing transaction is about 28 times on our approach.

As being seen in Figure 2, the performance of using a quantification table is better than without using it.

5. CONCLUSION

In this paper, we presented the method of mining the hedge algebra-fuzzy association rules and applying the data compression method for one database. With this approach, adjacent transactions will be merged into a new transaction. Thus, vertical size of input database is smaller. The algorithm

we gave has the characteristics: check the original database once in order to form a compressed database. As the compressed database has small size, we can take the whole database into RAM of the computer to handle, so it enables to increase the efficiency when mining data. Experiment shows that this proposed method offers better results compared to others available.

In additions, hedge algebras for items with the same algorithms are used in this paper. In order to improve the efficiency of mining association rules and to find out the significant rules more than the ones before, we need to maximum the fuzzy algorithms providing that they are appropriate to each attribute and assign weights to the attributes.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2013.34.

REFERENCES

[1] DL Olson, Yanhong Li, "Mining Fuzzy Weighted Association Rules", in *Proceedings of the 40th Hawaii International Conference on System Sciences*, 2007.

[2] Hannes Verlinde, Martine De Cock, and Raymond Boute, "Fuzzy Versus Quantitative Association Rules: A Fair Data - Driven Comparison", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 3, June 2006, pp. 679 - 684.

[3] C.H. Cai, Ada W.C. Fu, C.H. Cheng, W.W. Kwong, "Mining Association Rules with Weighted Items", in *Proceedings of IEEE International Database Engineering and Applications Symposium*, United Kingdom, 1998, pp. 68–77.

[4] R. Agrawal, T. Imielinski, A. Swami, "Fast Algorithms for Mining Association Rules", *the International Conference on Very Large Database*, 1994, pp. 487 - 499.

[5] J. Dai, D. Yang, J. Wu, and M. Hung, "An Efficient Data Mining Approach on Compressed Transactions", *World Academy of Science, Engineering and Technology*, vol. 3, Feb. 2008, pp. 76-83.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo., "Fast Discovery of Association Rules", *in Advances in Knowledge Discovery and Data Mining*, 1996, pp. 307–328.

[7] Jess Alcal-Fdez, Rafael Alcal, Mara Jos Gacto, Francisco Herrera, "Learning the Membership Function Contexts for Mining Fuzzy Association Rules by Using Genetic Algorithm Fuzzy Sets and Systems", vol. 160, 2009, pp. 905–921.

[8] Pietari Pulkkinen, Hannu Koivisto, "A Dynamically Constrained Multiobjective Genetic Fuzzy System for Regression Problems", *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, Feb. 2010.

[9] Nguyen Cong Hao, Nguyen Cong Doan, "Semantic Hedge Algebra based Fuzzy Association Rules (Luat Ket hop Mo dua tren Ngu nghia Dai so Gia tu)", *Journal of Science - Hue University, (Tap chi Khoa hoc âĂŞ Dai hoc Hue)*, vol. 5, 2012, pp. 39 - 52.

[10] Tran Thai Son, Do Nam Tien, Pham Dinh Phong, "Associate Rule Using Hedge Algebra approach *(Luat Ket hop theo Cach tiep can cua Dai so Gia tu)*", Journal of Computer Science and Cybernetics (*Tap chi Tin hoc va Dieu khien hoc*), vol. 27, no. 4, 2011.

[11] Jiawei Han, *Data Mining: Concepts and Techniques*. University of Illinois at Urbana-Champaign, Micheline Kamber.

[12] N. Cat Ho, W. Wechsler, "Extended hedge algebras and their application to Fuzzy logic," *Fuzzy Sets and Systems*, vol. 52, 1992, pp. 259 - 281.

[13] Nguyen Cat Ho, Tran Thai Son, "Intervals between Linguistic Variable Values in Hedge Algebra and the Problem of Fuzzy Classifications *(Ve Khoang cach giua Cac gia tri cua Bien Ngon ngu trong Dai so Gia tu va bai toan sap xep mo)*", *Journal of Computer Science and Cybernetics (Tap chi Tin hoc va Dieu khien hoc)*, vol. 1, 1995, pp. 10 - 20.

[14] Tran Thai Son, "Approximate Argument Using Linguistic Variable Values *(Lap luan Xap xi voi Gia tri cua Bien Ngon ngu)*", *Journal of Computer Science and Cybernetics (Tap chi Tin hoc va Dieu khien hoc)*, vol. 15, Oct. 1996.

[15] Nguyen Cat Ho, Tran Thai Son, Tran Dinh Khang, Le Xuan Viet, "Fuzziness Measure, Quantified Semantic Mapping And Interpolative Method of Approximate Reasoning in Medical Expert Systems", *Journal of Computer Science and Cybernetics (Tap chi Tin hoc va Dieu khien hoc)*, vol. 18, 2002, pp. 237 - 252.

[16] N. Cat Ho, H. Van Nam, "An Algebraic Approach to Linguistic Hedges in Zadeh's Fuzzy Logic", *Fuzzy Set and System*, vol. 129, 2002, pp. 229-254.

Received on May 18 - 2014
Revised on October 17 - 2014