Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 2 Issue 12, December - 2016

Applying Fuzzy Fp-Treeto Mine Fuzzy Association Rules Using Hedge Algebras Based Approach

Nguyen Tien Duy
Thai Nguyen University of Technology,
Thai Nguyen University, Thainguyen, Vietnam

Lam Hung Son
Thai Nguyen University of Technology,
Thai Nguyen University, Thainguyen, Vietnam

Nghiem Van Tinh
Thai Nguyen University of Technology, Thai Nguyen University
Thainguyen, Vietnam

***Abstract* — In recent years, research on data mining is widely concerned. A challenging problem is how to reduce the overall computation time of the mining algorithms association rules. In this paper, we propose fuzzy data compression method based on FP-Tree structure to find the frequent item sets. With this method, we use the hedge algebras for the fuzzy attributes to be defuzzicated. Experiment shows that this approach gives better results than a number of methods proposed previously.**

Keywords— Data mining, Association rules, FP-Tree, hedge algebras

I. INTRODUCTION

Recently, the strong development of technology has made the ability to collect and store the information systems increase quickly. Besides, the computerization of the manufacturing operations, business and many other areas of activity has created a large amount of data storage. Millions of database has been used in the production, sales, management, etc. Therefore, many of which are extremely large databases. This boom has led to an urgent requirement is to have the new techniques and tools to automatically convert huge amounts of data into useful knowledge. Then, the data mining techniques nowadays become a topical field of information technology in the world. Mining association rules has been implemented in research and bring good results [6] [7]. The authors have proposed many measures to reduce the time taken to explore the law, such as solution of mining association rules in parallel, using the compressed transaction solution on FP-Tree tree with binary database. However, in this area, there have been still many raised issues needed to be further investigated and resolved. Recently, the algorithms using data compression based on FP-tree in binary database provide a good solution and can reduce storage space requirements and the time for data processing. The authors of [6] [7] have proposed a solution for fuzzy transaction database compression based on FP-tree. With this approach, the authors have used the fuzzy theory to defuzzicate the transaction database. Processing the fuzzy data for data mining in

fuzzy association rules is mainly based on the theory of fuzzy set as shown in [6] [7]. However, through the way of using fuzzy, there are many factors affecting the accuracy such as determination of fuzzy sets and the opinions of experts. In order to improve the efficiency of mining association rules, in this paper we present method of defuzzicating the transaction database based on HA and compress fuzzy transaction database based on FP-Tree. Based approach will enable to reduce the number of nodes in the tree less than the method [6] [7] .

Rest of this paper is organized as follows. The related knowledge briefly is presented in Section II. In Section III, we presented mining fuzzy association rules. Then, the computational results and analyzed are shown in Section IV. The conclusions are discussed in Section V.

II. RELATED KNOWLEDGE

A. *Combination rule [1][1]*

Propose $\mathrm{I} = \mathrm{I}_1, \mathrm{I}_2, \dots, \mathrm{I}_\mathrm{m}$ is the set of m separate properties, each property is called an item. D is a transaction database, in which each record T is a transaction and containing the items $\mathrm{T} \subseteq \mathrm{I}$.

Definition 1: An association rule is a relation $X \Rightarrow Y$, where X, $Y \subset I$ is the ItemSets, and $X \cap Y = \emptyset$. Here, X is called the premise, Y is a result statement. Two important parameters of association rules are the support (s) and reliability (c).

Definition 2: The support of association rules $X \Rightarrow Y$ is the percentage of the record $X \cup Y$ including of the total of transactions in the database.

$$support(X \Rightarrow Y) = P(X \cup Y) = \frac{n(X \cup Y)}{N} \quad (2.1)$$

Definition 3: The confidence is the ratio of the number of transactions that contain $X \cup Y$ and the ones containing X.

$$confidence(X \Rightarrow Y) = P\left(\frac{X}{Y}\right) = \frac{n(X \cup Y)}{n(Y)} \quad (2.2)$$

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

Exploiting association rules through the database is to find all the laws which have the support and the confidence greater than the support level *Min_Sup* and the confidence level *Min_Sup* is determined by the user in advance.

Based on the fuzzy association rules, each item can be divided into the fuzzy domain (such as "*youth*", "*middle-aged*", ...), in fact we split an initial item into small items and the value of each row on that item will be in [0,1], not just 0 or 1.

Then the support of a fuzzy domain s_k which belongs to the item x_i is defined as:

$$FS(A_{s_k}^{x_i}) = \frac{1}{N}\sum_{j=1}^{N}\mu_{s_k}^{x_i}(d_j^{x_i}) \quad (II.1)$$

And the support of fuzzy domains $s_1, s_2, .., s_k$ of the corresponding items $x_1, x_2, .., x_k$ will be:

$$FS(A_{s_1}^{x_1}, A_{s_2}^{x_2}, \dots, A_{k}^{x_k}) = \frac{1}{N}\sum_{j=1}^{N} min\left(\mu_{s_1}^{x_1}(d_j^{x_1}), \mu_{s_2}^{x_2}(d_j^{x_2}), \dots, \mu_{s_k}^{x_k}(d_j^{x_k})\right) \quad (II.2)$$

With x_i is i[th] item, s_j is fuzzy domain of i[th] item, N is the total of transactions in the database, $\mu_{s_k}^{x_i}(d_j^{x_i})$ is the dependent level of values in i[th] column, j[th] row on fuzzy sets s_k.

B. *The FP-Growth Algorithm [2]*

FP-Growth was proposed by Jiawei Han's team, University of Illinois United States in 2000 [2]. FP-Growth algorithm transferred all the data into internal memory in a tree structure, the process to find the frequent set is the process of the tree browser. FP-Growth hit a new mark in the development of data mining, solves two nodes of the algorithm Apriori and Partition. The regular data items are detected with twice browse the database and no exceptional process for the candidate set.

FP-Tree algorithm is effective in calculating for three reasons. First, the process of solving the problem is just based on the regular data item, irregular data items are removed, so the data to review will be much smaller. Second, this algorithm just browses the database twice. Third, the FP-Tree uses the method of “divide and rule” to significantly reduce the size of the tree, a long branch was created with appended data item into a short branch, not to start over.

In the process of mining the database, users can change the level of support, but with FP-Tree, while changing the level of support, it must be done from the beginning. Another limitation of the FP-Tree is not suitable for increasing cases of data. Once the database changes, data mining work must also begin again.

C. *Some basic definitions of hedge algebra*

Consider an example of a set of linguistic value is the linguistic domain of the speed linguistic variable (SPEED) including the followings: *X = dom(SPEED) = {big, small, Very big, Very small, More big, More small, Approximately big, Approximately small, Little big, Little small, Possible big, Possible small, Less big, Less small, Very More big, Very More small, Very Possible big, Very Possible small, …}*. Meanwhile, linguistic domain $X = dom(SPEED)$ can be expressed as an algebraic structure $AX = (X, G, H, \leq)$, where: X is the background set of AX;

G is the set of original elements (set of generating elements: *big, small*), H is the set of monadic operators, called the hedges (*Very, More*, ...), demonstrates the order correlation on the linguistic values which is "induced" from the natural semantics of "the elements". X was born from G by the hedges of H. Thus, the representation of each element of x = $h_n h_{n-1} \dots h_1 c$, $c \in$ G. The set of the elements that is generated from the elements x and is represented as H (x).

Definition 2.1: [4][5]Hedge algebra is a quintuple, $AX = (X, G, C, H, \leq)$, Where: $G = \{c-, c+\}$, $C = \{0, W, 1\}$, $H = H- \cup H+$ và $\leq$ presents the order correlation on X.

Element **0** indicates the smallest element, the element **1** stands for only the largest elements and elements **W** is neutral one.

Definition 2.2: [4][5] Suppose the hedge algebra $AX = (X, G, C, H, \leq)$, $f: X \to [0,1]$ is the semantic quantitative function of AX if $\forall h, k \in H^+$ or $\forall h, k \in H^-$ and $\forall$x, y $\in$X:

$$\left|\frac{f(hx)-f(x)}{f(kx)-f(x)}\right| = \left|\frac{f(hy)-f(y)}{f(ky)-f(y)}\right|$$

Given a semantic quantification function f of X. At any x$\in$X, the fuzzy of x is then measured by the diameter $f(H(x)) \subseteq [0,1]$.

Definition 2.3: Fuzziness measure [5].

fm: X $\to$ [0, 1] is called the fuzziness measure when: $fm(c-) = \theta > 0$ and $fm(c+) = 1 - \theta > 0$, where $c-, c+ \in G$.

Suppose the set of hedge algebras $H = H + \cup H-$, $H- = \{h-1, h-2, \dots, h-q\}$ with $h-1 < h-2 < \dots < h-q$, $H+ = \{h1, h2, \dots, hp\}$ where $h1 < h2 < \dots < hp$. Then:

With any x, y $\in$X, $h\in$H, $\frac{fm(hx)}{fm(x)} = \frac{fm(hy)}{fm(y)}$

This equality does not depend on the elements x, y, and therefore we can denote it as $\mu(h)$ and call it as fuzziness measure of the hedge *h*. The characteristic of $fm(x)$ and $\mu(h)$ is as follows:

$$fm(hx) = \mu(h)fm(x), \forall x \in X \quad (II.5)$$

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

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

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

Sign function: $Sign: X \to \{-1, 0, 1\}$ is recursively defined as follows [5]:

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

$sign(hc) = +sign(c)$ if *h* is positive to *c* and $sign(hc) = -sign(c)$ if *h* is positive to *c*.

$sign(hc) = sign(h) \times sign(c)$

$sign(khx) = +sign(hx)$ if *k* positive to *h* (*sign*(*k*, *h*) = +1) and $sign(khx) = -sign(hx)$ if *k* positive to *h* $(sign(k,h) = +1)$

$\forall x \in H(G)$ can be presented as $x = hm \ldots h1c$, with $c \in G$ and $h1, \ldots, hm \in H$. Then:

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

$(sign(hx) = +1) \Rightarrow (hx \geq x)$ and $(sign(hx) = -1) \Rightarrow (hx \leq x)$

Suppose to preset the fuzziness measure of hedges μ(*h*) and the values of fuzziness measure for the generating elements $fm(c^-)$, $fm(c^+)$ and θ is the neutral element.

The semantic quantification function ν of T is recursively defined as follows [5]:

$v(w) = fm(c^-)$, $v(c^-) = \theta - \alpha fm(c^-) = \beta fm(c^-)$, $v(c^+) = \theta + \alpha fm(c^+) = 1 - \beta fm(c^+)$

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

(II.12)

$$\omega(h_j x) = \frac{1}{2}[1 + Sign(h_j x) sign(h_p h_j x)(\beta - \alpha)] \in \{\alpha, \beta\}$$

$, j \in [-q \wedge p], j \neq 0.$

D. Method of defuzzication

Based on the hedge algebra's approach, we will carry out computing the membership function value of each value of the database in the following way. First, we consider each attribute domain of each fuzziness as a hedge algebra. Instead of building the membership functions for the fuzzy domain was identified, we used quantitative semantic value to measure the value of the membership level at any of these items which is reviewed the new fuzzy domain determined.

Step 1: Standardize the values of the fuzzy properties in the range [0,1].

Step 2: Consider the fuzzy domain s_j of the attribute x_i as the elements of hedge algebra AX_i. Meanwhile, any of value $d_j^{x_i}$ of x_i is located between any two semantically quantitative values of the two certain semantic elements AX_i and the distance on the interval [0,1] (is the determined domain standardized of attributes x_i) between $d_j^{x_i}$ and the semantically quantitative value of the two elements that is the closest to $d_j^{x_i}$ to the both sides can be used to determine the proximity of $d_j^{x_i}$) into those two fuzzy domains (two elements of hedge algebra). The proximity between $d_j^{x_i}$ with other elements of hedge algebra is determined by 0. To determine the final membership level, we must standardize (transfer of value in the interval [0, 1] and then take 1 minus that standardized distance). For each value $d_j^{x_i}$, a couple of membership degree will be formed

Therefore, to calculate the membership degree of attributes x_i in fuzzy domain s_j : $\mu_{s_j}(d_j^{x_i}) = 1 - |v(s_j) - d_j^{x_i}|$, with $v(s_j)$ is the semantically quantitative value of the element s_j. (II.9)

III. Mining Fuzzy Association Rules

(II.10)

In this paper, we propose using hedge algebra and FP-Tree to find popular files. Finding common set is involved in the following steps: Phase 1: Defuzzicate the transaction database based on hedge algebra. Using fuzzy transaction database to build FP-tree fuzzy tree (called as HAFP), Phase 2: Finding common set based on HAFP trees.

A. Algorithm for the construction of fuzzy tree HAFP [6].

Symbol of parameters of the algorithm is as follows:

(II.11)

N Total transactions in the transaction database

m Total properties

A_j the j[th] property, 1 ≤ j ≤ m (digital attribute or category one)

$|A_j|$ The number of hedge tag of the property A_j

R_{jk} The j[th] hedge tag of attributes A_j

$D^{(i)}$ The i[th] transaction data, 1 ≤ i ≤ N

$v_j^{(i)}$ The value of the A_j in $D^{(i)}$

$f_{jk}^{(i)}$ The value of the membership degree $v_j^{(i)}$ of hedge labels R_{jk}, $0 \leq f_{jk}(i) \leq 1$

$Sup(R_{jk})$	The degree of support R_{jk}
Sup	The supportive value of each common set or item.
Conf	The reliability of each common set or item.
Min_Sup	The given minimum support value
L_r	The set of common items that correspond with r hedge labels (item set) 1 ≤ r ≤ m

Input: Transaction database D contains N transactions, the hedge algebras give the fuzzy atributes, *Min_Sup*

Output: Fuzzy tree HAFP

Step 1. Defuzzicate the fuzzy properties in the database. In this step, we use fuzzy method that is presented in item 2.4. The results of this step gained fuzzy transaction database.

Step 2. Browse the database is completed in Step 1, to compute the support degree *expCount(I)* and the frequency of appearance *f(I)*. Frequency of appearance *f(I)* is the number of transactions contained I that is not equal 0.

Step 3. Based on the support degree, *expCount(I)* is determined in the Step 2. If the value of support degree is more than or equal to Min_Sup, I is taken to the common set L1.

$$L1 = \{I: expCount(I) \mid expCount(I) \geq N * Min_{Sup}\}$$

Step 4. Sort the items in descending order in L1 of the frequency of appearance *f(I)* in I.

Step 5. Based on L1, build Header_Table contains: Item, expCount, and the frequency of appearance of item. Sort the items by the same way as in Step 4.

Step 6. Start the tree *HAFP* which has the root *Null*.

Step 7. Browse the database obtained in Step 1, omit the items that do not exist in L1. Arrange the items in the transactions in Step 4.

Step 8. Insert the transactions that have been arranged into the following tree HAFP as in these steps:

Step 8.1. In the transaction of appearance there is an element which has existed in the tree HAFP. Computing the values: multiple the item I's value to the value of the first item of the node's "parent", then plus the value of the corresponding element of the array (called *expArr*).

Step 8.2. In contrast, adding a new node corresponding to the tree. Calculating the values: Multiple the item's value to the value of the first item of the node's "parent", then plus the values of the corresponding element of the array (called expArr). Insert the similar links to the FP-tree algorithm [3].

B. *Algorithm to find the common set HAFP-Growth [6]*

After constructing fuzzy HAFP trees, we use the algorithm HAFP-Growth to find the common set.

Input: The fuzzy tree HAFP, Header_Table, the support Min_Sup

Output: The common set

Solve each item in Header_Table in order from the bottom to the top.

Step 1. Find all the nodes with items which are being processed I in the tree HAFP.

Step 2. Extract full of item set with the expected count from the array *expAry(K)* in each node K in Step 1.

Step 3. Compute the expected count of the same ItemSets.

Step 4. If ItemSets in Step 3 has the the value more than or equal to Min_Sup, these ItemSets will be taken to the common set.

Step 5. Repeat the Step 1 to Step 4 with the other items in Header_table until the last item is processed.

Ending the Step 5, we obtain the entire the common set from the HAFP tree.

IV. TEST EXAMPLES

A. *Build the tree hafp*

Transaction database in Table 1 is used in this example. This database contains 6 transactions and 6 fuzzy properties. The minimum level of support is 30%. We use a common hedge algebra X for the fuzzy attributes A, B, C, D, E, F. This hedge algebra includes generating element and two hedges shown follows:

$X = (X, G, H, \leq)$, where $C^- = \{Low\}$, $C^+ = \{Height\}$, $H^+ = \{Very\}$, $H^- = \{Least\}$. With $fm(Low) = fm(Height) = 0.5$, $\mu(Very) = \mu(Least) = 0.5$, $Dom(A) = [0, 13]$. Then $fm(Least\ Low) = 0.25$, $fm(Very\ Low) = 0.25$, $fm(Least\ Height) = 0.25$, $fm(Very\ Height) = 0.25$. We have the values: $v(Very\ Low) = 0.125$, $v(Least\ Low) = 0.375$, $v(Least\ Height) = 0.625$, $v(Very\ Height) = 0.875$.

TABLE I: TRANSACTION DATABASE

TID	A	B	C	D	E	F
1		7		12		10
2		10		10		12
3	2		9		1	
4	1		3	9	12	
5	4	9			9	12
6	1		3		1	9

Step 1: Fuzzy the transaction database.

Apply the fuzzy method presented in Section 2.4, the database in the Table 1 is defuzzicated and give the results shown in Table 2.

The attribute A has $Dom(A) = [0, 13]$, the value of properties A in the range [0, 1] as follows: {0, 0, 0.15, 0.08, 0.31, 0.08}. With $v(Very\ Low) = 0.125$, $v(Least\ Low) = 0.375$, $v(Least\ Old) = 0.625$, $v(Very\ Old) = 0.875$. For example, with A = 0.15: Because the value $v(Very\ Low) < 0.15 < v(Least\ Low)$, we just calculate the distance between the 0.15 and the two respective fuzzy domains Very Young and Least Young, the fuzzy domains Least Height, Very Height has the value 0. The difference between 0.15 and the fuzzy domain Very Young: $1 - Abs(0.15 - 0.125) = 0.98$. The difference between 0.15 and the fuzzy domain Least Young: $1 - Abs(0.15 - 0.375) = 0.78$. Similarly, we have the transaction database that is defuzzicated as in the Table 2.

Symbol: A1: A. Very Low; A2: A. Least Low; A3: A. Least Height; A4: A. Very Height; B1: B. Very Low; B2: B. Least Low; B3: B. Least Height; B4: B. Very Height; C1: C. Very Low; C2: C. Least Low; C3: C. Least Height; C4: C. Very Height; D1: D. Very Low; D2: D. Least Low; D3: D. Least Height; D4: D. Very Height; E1: E. Very Low; E2: E. Least Low; E3: E. Least Height; E4: E. Very Height; F1: F. Very Low; F2: F. Least Low; F3: F. Least Height; F4: F. Very Height.

TABLE II: RESULTS THE TRANSACTION DATABASE IN TABLE AFTER FUZZYING BASED ON HEDGE ALGEBRA

TID	ItemSets
1	B2=0.84; B3=0.92; D4=0.96; F3=0.86; F4=0.9;
2	B3=0.86; B4=0.9; D3=0.86; D4=0.9; F4=0.96;
3	A1=0.98; A2=0.78; C3=0.94; C4=0.69; E1=0.96;
4	A1=0.96; D3=0.94; D4=0.82; E4=0.96;
5	A1=0.82; A2=0.94; B1=0.44; B3=0.94; B4=0.82; E3=0.94; E4=0.82; F4=0.96;
6	A1=0.96; C1=0.9; C2=0.86; E1=0.96; F3=0.94; F4=0.82;

Step 2: Browse the database which has been completed in Step 1 to calculate the level of support expCount(I) and the frequency of appearance f(I). Frequency of appearance f (I) is the number of transactions contained I with the value that is not equal to 0. We have:

TABLE III: TRANSACTION DATABASE IN THE TABLE I AFTER DEFUZZICATING BASED ON HEDGE ALGEBRA

Item	expCount	Count	Item	expCount	Count	Item	expCount	Count	Item	expCount	Count
A1	3.72	4	B3	2.72	3	D1	0	0	E3	0.94	1
A2	1.72	2	B4	1.72	2	D2	0	0	E4	1.78	2
A3	0	0	C1	0.9	1	D3	1.8	2	F1	0	0
A4	0	0	C2	0.86	1	D4	2.68	3	F2	0	0
B1	0.44	1	C3	0.94	1	E1	1.92	2	F3	1.8	2
B2	0.84	1	C4	0.69	1	E2	0	0	F4	3.64	4

Step 3: With the minimum degree of support 30%, based on the degree of support expCount(I) is determined in 0. We can determine the common set L1= {A1:3.72; B3: 2.72; D4: 2.68; E1: 1.92; F4: 3.64}.

Step 4: Sort L1 in descending the frequency of appearance of f(I), we can obtain L1 = {A1: 3.72; F4: 3.64; B3: 2.72; D4: 2.68; E1: 1.92; }.

Step 5: Based on L1 and Table 1, we construct the table Header_table as in **Error! Reference source not found.**.

TABLE IV: HEADER_TABLE

Item	expCount	Count
A1	3.72	4
F4	3.64	4
B3	2.72	3
D4	2.68	3
E1	1.92	2

Step 6: Initialize the tree HAFP which has the root *Null*.

TABLE V: THE FUZZY DATABASE AFTER HAVING BEEN UPDATED

TID	ItemSets
1	F4=0.9; B3=0.92; D4=0.96;
2	F4=0.96; B3=0.86; D4=0.9;
3	A1=0.98; E1=0.96;
4	A1=0.96; D4=0.82;
5	A1=0.82; F4=0.96; B3=0.94;
6	A1=0.96; F4=0.82; E1=0.96;

Step 7: Browse the database obtained in 0, remove the items which do not exist in L1. Sort the items in the transactions in order as in 0. We obtained a transaction database after updating as shown in Table 5.

Step 8: After all of the transactions are processed, we obtain the tree HAFP as shown in Fig 1.

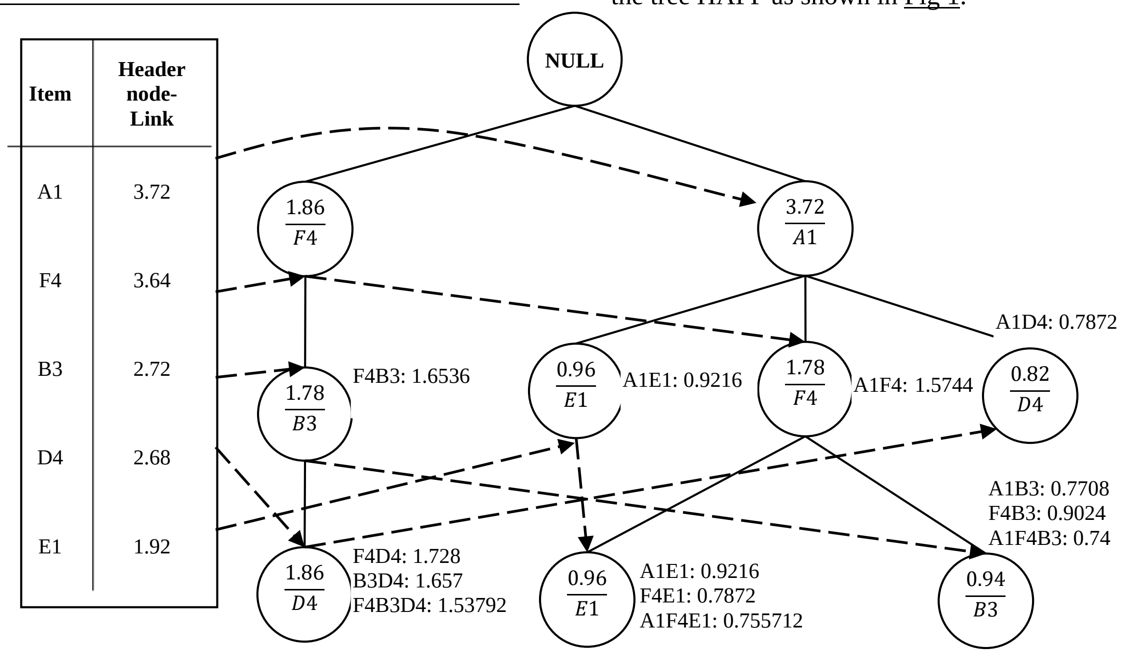

Fig. 1: The tree HAFP

B. Finding the common set

To find common set, we carry out the following steps:

Step 1: Browse items orderly one by one in Header_Table from the bottom to the top. The order to perform is respectively: E1, D4, B3, F4, A1.

Step 2: The point E1 is carried out firstly, in the tree HAFP are 2 points E1.

Step 3: The candidate sets of two points E1 is: A1E1: 0.9216, A1E1: 0.9216, F4E1: 0.7872, A1F4E1: 0.755712.

TABLE VI: THE COMMON SET

1- ItemSets	
A1	A1
F4	F4
B3	B3
D4	D4
E1	E1
2- ItemSets	
A1E1	A1E1

Step 4: Plusing the values of the same candidate sets, we obtain the sets of candidates as

follows: A1E1: 1.8432, F4E1: 0.7872, A1F4E1: 0.755712.

Step 5: With the minimum support of 30% (0.3 * 6 = 1.8). The candidate sets in Step 4 have the value that are greater than or equal to 1.8, then they are put into common set. In this example A1E1: 1.8432 was put into common set.

Step 6**:** Repeat with the other items in Header_Table, we obtain common as in the table

V. CONCLUSION

In this paper, we present methods of mining fuzzy association rules based hedge algebra's approach, using data compression based on FP-tree for a database. With this approach, we used hedge algebra to fuzzy and represent the data after defuzzicating based on FP-tree. Compared with other methods, this method enables to minimize the number of points in the tree to help speed up finding common set.

In the article we used the hedge algebra for the items with the same choosen parameters. To improve the efficiency of fuzzy mining association rules and to find the more meaningful laws, we need to optimize these fuzzy parameters which are suitable for each attribute, assign weights to the attributes.

ACKNOWLEDGMENT

The work reported in this paper has been supported by Thai Nguyen University of Technology - Thai Nguyen University.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami (1994), "Fast algorithms for mining association rules", *The International Conference on Very Large Database*, pp. 487 - 499.

[2] Han J., Pei H., Yin Y (2000), "Mining Frequent Patterns without Candidate Generation", *In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press*, New York, NY, USA, pp.1-12.

[3] Cornelia Gyorödi, Robert Gyorödi, T. Cofeey & S. Holban, (2003) "Mining association rules using Dynamic FP-trees", *in Proceedings of The Irish Signal and Systems Conference*, University of Limerick, Limerick, Ireland, ISBN 0-9542973-1-8, pp. 76-82.

[4] N. Cat Ho and W. Wechler (1992), "Extended hedge algebras and their application to Fuzzy logic", *Fuzzy Sets and Systems*, 52, pp. 259 - 281.

[5] Cat Ho, N. and H. Van Nam (2002), "An algebraic approach to linguistic hedges in Zadeh's fuzzy logic", *Fuzzy Set and System*, 129, pp. 229-254.

[6] Chun-Wei Lin, Tzung-Pei Hong, and Wen-Hsiang Lu (2009), "Fuzzy Data Mining Based on the Compressed Fuzzy FP-trees", *Fuzzy-IEEE*, Korea.

[7] Chun-Wei Lin, Tzung-Pei Hong (2012), "A new mining approach for uncertain databases using CUFP trees", *Expert Systems with Applications*, 39, pp 4084–4093.