Journal of Universal Computer Science, vol. 19, no. 4 (2013), 521-538
submitted: 3/8/12, accepted: 20/2/13, appeared: 28/2/13 © *J.UCS*

An Integrated MFFP-tree Algorithm for Mining Global Fuzzy Rules from Distributed Databases

Chun-Wei Lin
(Innovative Information Industry Research Center (IIIRC)
Shenzhen Key Laboratory of Internet Information Collaboration
School of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus Shenzhen University Town, Xili, Shenzhen 518055 P.R. China
jerrylin@ieee.org)

Tzung-Pei Hong*
(Department of Computer Science and Information Engineering
National University of Kaohsiung, Kaohsiung, Taiwan, R.O.C.
Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C.
tphong@nuk.edu.tw)

Yi-Fan Chen
(Department of Applied Mathematics
National University of Kaohsiung, Kaohsiung, Taiwan, R.O.C.
A0974154@mail.nuk.edu.tw)

Tsung-Ching Lin, **Shing-Tai Pan**
(Department of Computer Science and Information Engineering
National University of Kaohsiung, Kaohsiung, Taiwan, R.O.C.
m0985507@mail.nuk.edu.tw, stpan@nuk.edu.tw)

Abstract: In the past, many algorithms have been proposed for mining association rules from binary databases. Transactions with quantitative values are, however, also commonly seen in real-world applications. Each transaction in a quantitative database consists of items with their purchased quantities. The multiple fuzzy frequent pattern tree (MFFP-tree) algorithm was thus designed to handle a quantitative database for efficiently mining complete fuzzy frequent itemsets. It however, only processes a database for mining the desired rules. In this paper, we propose an integrated MFFP (called iMFFP)-tree algorithm for merging several individual MFFP trees into an integrated one. The proposed iMFFP-tree algorithm firstly handles the fuzzy regions for providing linguistic knowledge for human beings. The integration mechanism of the proposed algorithm thus efficiently and completely moves a branch from one sub-tree to the integrated tree. The proposed approach can derive both global and local fuzzy rules from distributed databases, thus allowing managers to make more significant and flexible decisions. Experimental results also showed the performance of the proposed approach.

Keywords: iMFFP tree, integration, fuzzy data mining, quantitative database, distributed database
Categories: H.2.8, E.1, M.4, M.7

* Corresponding author

1 Introduction

Data mining techniques have recently been designed for deriving useful knowledge from databases [Agrawal et al., 93, Agrawal and Srikant, 94, Hu et al., 99, Lent et al., 97, Berzal et al., 01, Ferrer-Troyano et al., 05, Lan et al., 11]. Depending on the variety of knowledge required, data mining approaches can be divided into association rules [Agrawal et al., 93, Agrawal and Srikant, 94, Hong et al., 08, Lin et al., 09], classification [Hu et al., 99, Sucahyo and Gopalan, 05, Zaman and Hirose, 11, Woźniak and Krawczyk, 12], clustering [Lent et al., 97, Liu et al., 01, Hong and Wu, 11], and sequential patterns [Agrawal and Srikant, 95, Srikant and Agrawal, 96] among others. Among them, association rules mining is especially common in data mining research [Berzal et al., 01, Chen et al., 96, Park et al., 95]. Most of the algorithms in association rules mining use the level-wisely approach to generate-and-test candidate itemsets for obtaining the required information in a batch way. The iterative approach, however, requires a high computational cost for rescanning the whole database. Han et al. thus proposed the frequent-pattern tree (FP–tree) approach to speed up the mining process [Han et al., 04].Fuzzy set theory [Zadeh, 65, Jain and Stallings, 78] has been increasingly used in intelligent systems [Pulkkinen and Koivisto, 10, Senge and Hüllermeier, 11, Wang et al., 12] due to its simplicity and similarity to human reasoning. Linguistic representation is popular since it makes knowledge more understandable for human beings. In the past, Janikow used the fuzzy representation to combine symbolic decision trees on rule-based systems for fuzzy control [Janikow, 98]. Hong et al. proposed a fuzzy mining algorithm for mining fuzzy association rules from quantitative data [Hong et al., 04]. Lin et al. then proposed the fuzzy FP-tree [Lin et al., 10a] to improve the mining of fuzzy frequent itemsets from quantitative databases. That algorithm transformed quantitative values in transactions into linguistic terms based on Hong et al.'s approach. Only the linguistic term with the maximum cardinality was used, making the number of fuzzy regions processed equal to the number of original items. It could thus reduce the processing time. Hong et al. [Hong et al., 12] then proposed a multiple fuzzy frequent pattern tree (MFFP-tree) algorithm for deriving more complete information than the fuzzy FP-tree algorithm.

The above approaches processed the whole database to find the desired information. In some applications, however, a company may own multiple branches, and each branch has its own local database. The manager in a parent company needs to make a decision for the entire company from the collected databases in different branches. Besides, each branch may need to make its own decision as well. Thus, it is important to efficiently integrate many different databases to efficiently mine both the entire and individual knowledge.

In the past, Lan and Qiu proposed a parallel algorithm called PFPTC algorithm [Lan and Qiu, 05] for merging several FP trees into an integrated one from binary databases. The QFP-growth mining approach of the PFPTC algorithm was also designed for mining the desired knowledge without generating a huge number of intermediate results. In this paper, we extend the PFPTC algorithm to quantitative data and propose an MFFP-tree merging algorithm for integrating different MFFP trees into one (iMFFP) tree. Based on the proposed algorithm, both the global and local knowledge can be derived at the same basis of frequent items. The remainder of this paper is organized as follows. The related work is mentioned in Section 2. The

proposed algorithm for integrating multiple MFFP trees is described in Section 3. An example to illustrate the proposed algorithm is given in Section 4. Experimental results are shown in Section 5. Conclusions and discussions are described in Section 6.

2 Related Work

In this section, some related researches are briefly reviewed. They are fuzzy-set concepts, fuzzy data mining approaches, and the fuzzy FP-tree algorithm.

2.1 Fuzzy-set Concepts

In 1965, Zadeh proposed fuzzy sets and introduced membership functions as a method of linguistic description [Zadeh, 65]. The fuzzy sets with their linguistic modes of reasoning are more natural to human beings, rather than the binary logic of 0 and 1 in computer science. The fuzzy set theory extends this concept by defining partial memberships, which can take values ranging from 0 to 1. A membership function is formally defined as:

$$u_A : X \to [0, 1],$$

where X refers to the universal set for a specific problem. Assume that A and B are two fuzzy sets with membership functions $u_A(x)$ and $u_B(x)$, respectively. The following common fuzzy operators can be defined [Kandel, 92].

(1) The intersection operator:

$$u_{A \cap B}(x) = u_A(x) \, \tau \, u_B(x),$$

where τ is a t-norm operator. That is, τ is a function of $[0,1] * [0,1] \to [0,1]$ and must satisfy the following conditions for each $a, b, c \in [0,1]$:

i. $a \, \tau \, 1 = a$;
ii. $a \, \tau \, b = b \, \tau \, a$;
iii. $a \, \tau \, b \geq c \, \tau \, d$ if $a \geq c, b \geq d$;
iv. $a \, \tau \, b \, \tau \, c = a \, \tau \, (b \, \tau \, c) = (a \, \tau \, b) \, \tau \, c$.

Two instances of a t-norm operator for $a \, \tau \, b$ are $min(a, b)$ and $a \times b$.

(2) The union operator:

$$u_{A \cup B}(x) = u_A(x) \, \rho \, u_B(x),$$

where ρ is an s-norm operator. That is, ρ is a function of $[0,1] * [0,1] \to [0,1]$ and must satisfy the following conditions for each $a, b, c \in [0,1]$:

i. $a \, \rho \, 0 = a$;
ii. $a \, \rho \, b = b \, \rho \, a$;
iii. $a \, \rho \, b \geq c \, \rho \, d$ if $a \geq c, b \geq d$;
iv. $a \, \rho \, b \, \rho \, c = a \, \rho \, (b \, \rho \, c) = (a \, \rho \, b) \, \rho \, c$.

Two instances of an s-norm operator for a ρ b are $max(a, b)$ and $a + b - a \times b$.

(3) The α-cut operator:

$$A_\alpha(x) = \{x \in X \mid u_A(x) \geq \alpha\},$$

where A_α is an α-cut of a fuzzy set A. A_α thus contains all the elements in the universal set X that have membership grades in A greater than or equal to the specified value of α. These fuzzy operators are used in the proposed iMFFP-tree algorithm to derive fuzzy association rules.

2.2 Fuzzy Data Mining Approaches

Fuzzy set theory [Zadeh, 65] has been increasingly used in intelligent systems due to its simplicity and similarity to human reasoning. It is useful to extract knowledge from real-world data and to represent it in a comprehensible form. Linguistic representation is popular since it makes knowledge more understandable for human beings and easily implemented by fuzzy sets by concerning the fuzzy-set theory with quantifying and reasoning using natural language. Many fuzzy learning algorithms for inducing rules from given sets of data have been proposed and used to good effect in specific domains.

In the past, Hong et al. proposed a fuzzy mining algorithm for mining fuzzy association rules from quantitative data [Hong et al., 04]. Traditional association rules can be represented as: ***IF*** *Bread is bought,* ***THEN*** *Milk is bought together, with a confidence value 80%.* The fuzzy association rules, on the other hand, can be represented as: ***IF*** *a high amount of Bread is bought,* ***THEN*** *a middle amount of Milk is bought together, with a confidence value 80%.* Chan et al. also proposed the F-APACS algorithm to transform quantitative attribute values into linguistic terms [Chan and Au, 1997]. The algorithm uses membership functions to transform each quantitative value into a fuzzy set in linguistic terms. The cardinality of each linguistic term is then calculated for all transactions for mining interesting associations among attributes. Kuok et al. proposed a fuzzy mining approach for handling numerical data and deriving fuzzy association rules in databases [Kuok et al., 98].

Mahmoudi et al. integrated the fuzzy-set concept, the ant colony system, and multiple-level taxonomy to derive multiple-level fuzzy association rules from quantitative transactions [Mahmoudi et al., 11]. Three phases respectively for extracting membership functions by the ACS algorithm, for computing minimum supports of items in a database, and for defining multiple minimum supports of items were involved in the proposed algorithm. Li and Hu weighted items in a database for developing a framework to find weighted fuzzy association rules [Li and Hu, 11]. The item weight, item-set transaction weight, fuzzy weighted support, and fuzzy weighted confidence were stated in that proposed framework for discovering more frequent itemsets and rules than those from traditional approaches. Some other mining methods for finding fuzzy association rules have been proposed as well [Delgado et al., 03, Wang et al., 05, Lin et al., 10b].

2.3 Fuzzy FP-tree Algorithm

Frequent pattern mining is one of the most important research issues in data mining. The initial algorithm for mining association rules was given by Agrawal et al. [Agrawal et al., 93] in the form of the Apriori algorithm, which is based on level-

wisely generation-and-test approach for candidates. Since the size of a database may be very large, it is thus very costly to repeatedly scan the database to calculate supports of candidate itemsets. The limitation of the Apriori algorithm was overcome by an innovative approach, the frequent pattern (FP) tree structure and the FP-growth algorithm, proposed by Han et al. [Han et al., 04]. The approach can efficiently mine frequent itemsets without the generation of candidate itemsets, and it scans the original transaction database only twice. The mining algorithm consists of two phases. The first constructs an FP-tree structure and the second recursively mines the frequent itemsets from the structure. After the FP-growth algorithm is executed, the frequent itemsets satisfying a given minimum support threshold are derived from the FP-tree structure.

Papadimitriou and Mavroudi then proposed an approach based on FP trees for finding fuzzy association rules [Papadimitriou and Mavroudi, 05]. A fuzzy region in a transaction is removed if its fuzzy value is smaller than the support threshold. In their process, only the local frequent fuzzy 1-itemsets kept in each transaction are used for mining. The expression of fuzzy patterns is straight without using any fuzzy operations to form the desired rules. This procedure thus makes the mined fuzzy rules a little difficult to understand. Mishra et al. proposed a frequent-pattern mining approach to handle a fuzzified gene expression dataset [Mishra et al., 11]. They showed that the fuzzy vertical dataset format could derive more fuzzy frequent itemsets than the original one.

Lin et al. respectively proposed the fuzzy FP-tree and the CFFP-tree [Lin et al., 10a, Lin et al., 10b] to efficiently mine fuzzy frequent itemsets from quantitative databases. These algorithms transform quantitative values in transactions into linguistic terms based on Hong et al.'s approach [Hong et al., 04]. Only the linguistic term with the maximum cardinality is used in later mining processes, making the number of fuzzy regions processed equal to the number of original items. It can thus reduce the processing time. The frequent fuzzy itemsets, represented by linguistic terms, are then derived from the fuzzy FP tree. Hong et al. [Hong et al., 12] then proposed a multiple fuzzy frequent pattern tree (MFFP-tree) algorithm for deriving more complete information than the fuzzy FP-tree algorithms. Using multiple regions of an item can derive more fuzzy association rules than using a single region. The former can thus be considered to attain more complete information. Different linguistic terms for an item can exist in fuzzy association rules. For example, in the two rules "***IF*** *a high amount of Bread is bought,* ***THEN*** *a middle amount of Milk is bought together, with a confidence value 80%" and "****IF*** *a low amount of Bread is bought,* ***THEN*** *a high amount of Cake is bought together, with a confidence value 70%",* the two terms *"high"* and *"low"* for Bread can exist at the same time. This kind of rules is thus adopted in this paper.

3 The Proposed iMFFP-tree Algorithm

Data mining is usually used to find the relationships between the purchased items for indicating the purchasing habits of customers. That is, it is a powerful tool for making the efficient and correct decisions for managers in companies. In the past research, many algorithms were proposed for processing the whole database to find the desired information. In real-world applications, however, a parent company may own

multiple branches, and each branch has its own local database. A manager in a parent company may need to make a decision for the entire company from the collected databases in different branches. Thus, it is important to efficiently integrate many different databases for forming a useful decision.

In this section, a MFFP-tree merging algorithm for integrating different databases into one is proposed, thus forming an integrated MFFP (iMFFP) tree. The iMFFP tree inherits the property of MFFP tree for handling quantitative databases in fuzzy data mining. Each branch of a company thus has its specified MFFP tree for making its own decision. The parent company then integrates those individual MFFP trees for making the global decision for the company.

3.1 Notation

N	the number of quantitative databases;
DB_k	the quantitative k(-th) database, $1 \le k \le N$;
n	the number of transactions in D;
T	the i-th transaction in D, $1 \le i \le n$;
m	the number of items in D;
I_j	the j-th item, $1 \le j \le m$;
h_j	the number of fuzzy regions for I_j
R_{jl}	the l-th fuzzy region of I_j, $1 \le l \le h_j$;
t	the level of the processed fuzzy region R_{jl}, which is then increased from bottom to top;
v_{ij}	the quantitative value of I_j in T;
f_{ijl}	the membership value of v_{ij} in region R_{jl};
$count_{jl}$	the count of the fuzzy region R_{jl} in D;
s	the predefined minimum support threshold.

3.2 The Proposed iMFFP-tree Algorithm

In the proposed algorithm, each sub-MFFP tree is merged into the integrated MFFP tree in sequence. The branches in each sub-MFFP tree are extracted and then inserted into the integrated MFFP tree. The details of the algorithm are described below.

***The integrated MFFP-tree algorithm*:**

INPUT: A quantitative database D with n transactions from N multiple quantitative sub-databases, a set of membership functions, and a predefined minimum support threshold s.

OUTPUT: k multiple MFFP trees and an integrated MFFP (iMFFP) tree.

STEP 1: Transform the quantitative value v_{ij} of each item I_j in the i-th transaction of the database D into a fuzzy set f_{ij} represented as ($f_{ij1}/R_{j1} + f_{ij2}/R_{j2} + \ldots + f_{ijh}/R_{jh}$) using the given membership functions, where h is the number of fuzzy regions for I_j, R_{jl} is the l-th fuzzy region of I_j, $1 \le l \le h$, and f_{ijl} is the fuzzy membership value of v_{ij} in region R_{jl}. Note that f_{ijl}/R_{jl} means that the membership value of region R_{jl} for v_{ij} is f_{ijl}.

STEP 2: Calculate the scalar cardinality $count_{jl}$ of each fuzzy region R_{jl} in the transactions of D as:

$$count_{ji} = \sum_{i=1}^{n} f_{jil}.$$

STEP 3: Check whether the value $count_{jl}$ of the fuzzy region R_{jl} is larger than or equal to the predefined minimum count $n \times s$. If the count of a fuzzy region R_{jl} is equal to or greater than the minimum count, it can be treated as a fuzzy frequent itemset and put it in the set of L_1. That is:

$$L_1 = \{R_{jl} \mid count_{jl} > n \times s,\ 1 \le j \le m,\ m \text{ is the number of items}\}.$$

STEP 4: Build the sub-MFFP tree of each sub-database DB, which only keeps the fuzzy regions existing in L_1.

STEP 5: Initially set the variable k as 1 and the iMFFP tree as the the sub-$MFFP_1$ tree derived from DB_1.

STEP 6: Find all the leaf nodes in the sub-$MFFP_{k+1}$ tree derived from DB_{k+1} and trace the leaf nodes bottom-up from branches. Let the processed node be donated as R_{jl} and its child node as R_{jl}'. If R_{jl} has no or only one child node, i.e. R_{jl}' is ***null***, set the count of node R_{jl} as its original count in the node. Else if the currently processed branch is not the last one of R_{jl}, the count of $R_{jl} = R_{jl}'$; otherwise, the count of R_{jl} = the original count of R_{jl} − the summation of counts of R_{jl} in the other branches.

STEP 7: Merge the extracted branches in STEP 6 in a top-down way to the iMFFP tree. The following two cases may exist.

Substep 7-1: If a fuzzy region R_{jl} in an extracted branch is at the corresponding branch of the iMFFP tree, add the fuzzy value f_{ijl} of R_{jl} in the extracted branch to the corresponding node of the branch in the iMFFP tree.

Substep 7-2: Otherwise, add a node of R_{jl} at the end of the corresponding branch, set the count of the node as the fuzzy value f_{ijl} of R_{jl}, and connect the node to the last R_{jl} node in the other branch.

STEP 8: Set $k = k+1$.

STEP 9: Repeat STEPs 6 to 8 until all DB_k are integrated into the iMFFP tree.

STEP 10: Build the Header_Table by keeping fuzzy frequent itemsets in STEP 3. Insert a link from the entry of R_{jl} in the Header_Table to the first branch of node R_{jl} in the iMFFP tree.

4 An Example

In this section, an example is given to illustrate the proposed iMFFP-tree algorithm. Assume that there are two quantitative databases DB_1 and DB_2 which are shown in Table 1, and the minimum support threshold s is set to 30%. Both of them consist of 4 transactions and 5 items, denoted $\{A\}$ to $\{E\}$.

Table 1: Two quantitative databases

TID	*Item*	*DB*
1	(*A*:5) (*C*:10) (*D*:2) (*E*:9)	DB_1
2	(*A*:8) (*B*:2) (*C*:3)	DB_1
3	(*B*:3) (*C*:9)	DB_1
4	(*A*:7) (*C*:9) (*D*:3)	DB_1
5	(*A*:5) (*B*:2) (*C*:5)	DB_2
6	(*A*:3) (*C*:10) (*D*:2) (*E*:2)	DB_2
7	(*A*:5) (*B*:2) (*C*:8) (*E*:6)	DB_2
8	(*C*:9) (*D*:3)	DB_2

Assume that the fuzzy membership functions are the same for all the items shown in Figure 1. In this example, amounts are represented by three fuzzy regions: {*Low*}, {*Middle*}, and {*High*}. Thus, three fuzzy membership values are produced for each item in a transaction according to the predefined membership functions in Figure 1. Note that the proposed approach also works when the membership functions of the amounts for the items are not the same.

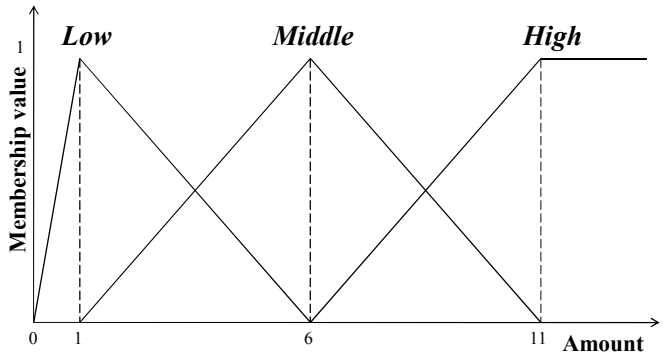

Figure 1: Membership functions used in the example

The procedure of the MFFP-tree merging algorithm for this example is described below. Note that the sub-MFFP tree of DB_2 is then merged into sub-MFFP tree of DB_2 to form the iMFFP tree.

The quantitative values of the items in the transactions are first represented as fuzzy sets using the membership functions shown in Figure 1. Take item {*A*} in transaction 1 as an example. The amount "5" of {*A*} is converted into the fuzzy set $(\frac{0.2}{A.Low}, \frac{0.8}{A.Middle})$ by the membership functions in Figure 1. This step is repeated for the other items in Table 1, and the results are shown in Table 2.

Table 2: Fuzzy sets transformed from Table 1

TID	***Item***	***DB***
1	$(\frac{0.2}{A.Low},\frac{0.8}{A.Middle}),(\frac{0.2}{C.Middle},\frac{0.8}{C.High}),(\frac{0.8}{D.Low},\frac{0.2}{D.Middle}),(\frac{0.4}{E.Middle},\frac{0.6}{E.High})$	DB_1
2	$(\frac{0.6}{A.Middle},\frac{0.4}{A.High}),(\frac{0.8}{B.Low},\frac{0.2}{B.Middle}),(\frac{0.6}{C.Low},\frac{0.4}{C.Middle})$	DB_1
3	$(\frac{0.6}{B.Low},\frac{0.4}{B.Middle}),(\frac{0.4}{C.Middle},\frac{0.6}{C.High})$	DB_1
4	$(\frac{0.8}{A.Middle},\frac{0.2}{A.High}),(\frac{0.4}{C.Middle},\frac{0.6}{C.High}),(\frac{0.6}{D.Low},\frac{0.4}{D.Middle})$	DB_1
5	$(\frac{0.2}{A.Low},\frac{0.8}{A.Middle}),(\frac{0.8}{B.Low},\frac{0.2}{B.Middle}),(\frac{0.2}{C.Low},\frac{0.8}{C.Middle})$	DB_2
6	$(\frac{0.6}{A.Low},\frac{0.4}{A.Middle}),(\frac{0.2}{C.Middle},\frac{0.8}{C.High}),(\frac{0.8}{D.Low},\frac{0.2}{D.Middle}),(\frac{0.8}{E.Low},\frac{0.2}{E.Middle})$	DB_2
7	$(\frac{0.2}{A.Low},\frac{0.8}{A.Middle}),(\frac{0.8}{B.Low},\frac{0.2}{B.Middle}),(\frac{0.6}{C.Middle},\frac{0.4}{C.High}),(\frac{1}{E.Middle})$	DB_2
8	$(\frac{0.4}{C.Middle},\frac{0.6}{C.High}),(\frac{0.6}{D.Low},\frac{0.4}{D.Middle})$	DB_2

The scalar cardinality of each fuzzy region in the transactions of the two databases is calculated as the count value and be checked against the specified minimum count, which is (8×0.3) (= 2.4) to find fuzzy frequent 1-itemsets. Take the fuzzy region {*B.Low*} as an example to explain the procedure. {*B.Low*} appears in transactions 2, 3, 5 and 7. Its scalar cardinality is calculated as (0.8 + 0.8 + 0.8 + 0.6) (= 3.0). Since the count for {*B.Low*} is larger than the minimum count, {*B.Low*} is then kept in the set of L_1. The results are shown in Table 3.

Table 3: Counts of fuzzy regions (fuzzy frequent items)

Fuzzy region	***Count***
A.Middle	4.2
B.Low	3.0
C.Middle	3.4
C.High	3.8
D.Low	2.8

The sub-MFFP trees of two quantitative databases are then respectively built. The results of the two trees are shown in Figures 2 and 3. The leaf nodes of the MFFP-tree in DB_2 are then traced one by one. In this example, the three leaf nodes of the MFFP tree in DB_2 are marked in red color in Figure 4.

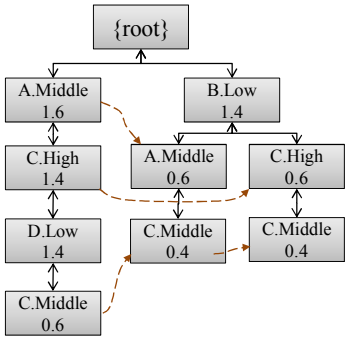

Figure 2: The sub-MFFP tree of DB_1

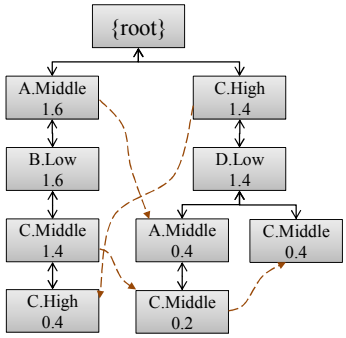

Figure 3: The sub-MFFP tree of DB_2

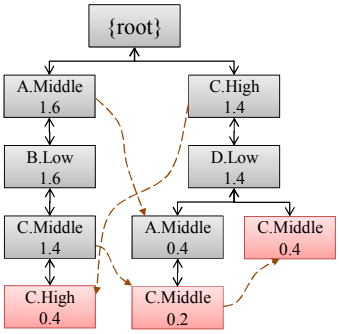

Figure 4: The leaf nodes of the currently processed MFFP-tree

The three branches can then be extracted from these leaf nodes. Take the right branch in Figure 4 as an example to illustrate the process. The leaf node (*C.Middle*:0.2) is set as the currently processed node and firstly extracted to form the extracted branch for integration. The parent of (*C.Middle*:0.2) is (*A.Middle*:0.4), which has only one child of (*C.Middle*:0.2). Thus, (*A.Middle*:0.4) is directly extracted to form the extracted branch with (*C.Middle*:0.2); (*A.Middle*:0.4) is then set as the currently processed node. The parent of (*A.Middle*:0.4) is (*D.Low*:1.4). Since (*D.Low*:1.4) has two children; the count of (*D.Low*) is then set at 0.4, which is the same as the (*A.Middle*) in the extracted branch. (*D.Low*:1.4) is then set as the currently processed node. The parent node of (*D.Low*:1.4) is (*C.High*:1.4). Since (*C.High*:1.4) has only one child of (*D.Low*:1.4), (*C.High*) is directly extracted to form the extracted branch with (*D.Low*:0.4, *A.Middle*:0.4, *C.Middle*:0.2). The count, however, of (*D.Low*) was set at 0.4 in the extracted branch. Thus, the count of (*C.High*) is also set at 0.4, and the currently processed node is changed to (*C.High*:1.4). The parent of (*C.High*:1.4) is ***root***. The process to extract the nodes of leaf node (*C.Middle*:0.2) is terminated. The other leaf nodes are processed in the same way. Figure 5 shows the result of the three branches.

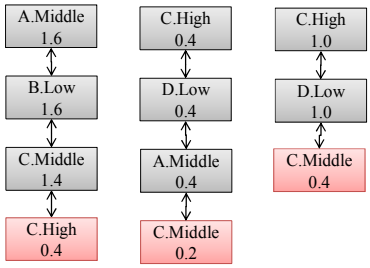

Figure 5: Three branches of the currently sub-MFFP tree

The three branches of the sub-MFFP tree of DB_2 are then inserted into the iMFFP tree, which is initially set as the sub-MFFP tree from DB_1. Take the first branch as an example. Since the item {*A.Middle*} is at the corresponding branch of the initial iMFFP tree, the fuzzy value of the item {*A.Middle*} in this branch is added to the node of *A.Middle*. The remaining items of the branch are then inserted into the iMFFP tree as a new branch. The result is shown in Figure 6.

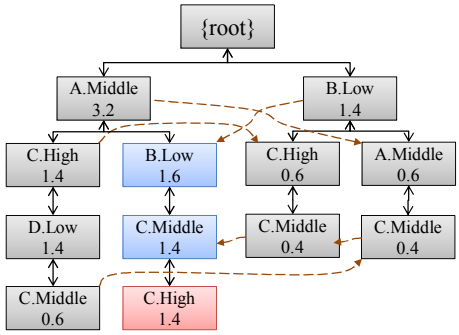

Figure 6: The iMFFP tree after merging the first branch

The above steps are repeated for the other two branches. Figures 7 and 8 show the remaining merging procedure.

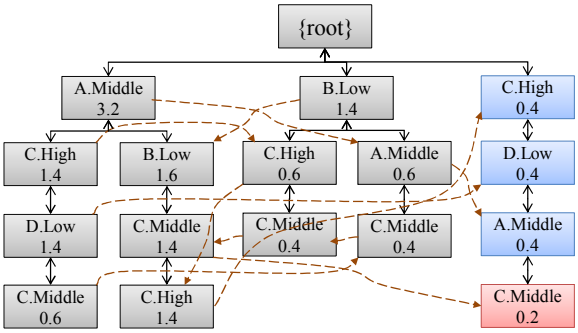

Figure 7: The iMFFP tree after merging the second branch

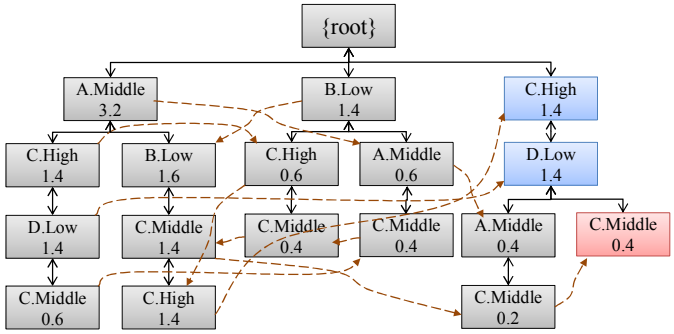

Figure 8: The iMFFP tree after merging the third branch

Since there are no other sub-MFFP-trees to be merged, the Header_Table is created and links are connected from the entry of each fuzzy region in the Header_Table to the corresponding node of its first branch. The final integrated MFFP-tree and its Header_Table are shown in Figure 9.

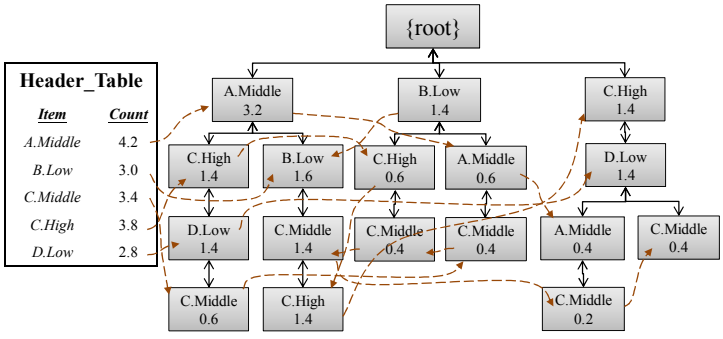

Figure 9: The final integrated MFFP-tree

After that, the MFFP-growth mining algorithm [Hong et al., 12] is then performed to extract the desired fuzzy frequent itemsets.

5 Experimental Results

The experiments were performed on a real dataset called CONNECT [Bayardo]. This dataset was prepared by Roberto Bayardo from the UCI datasets which belongs to a condensed dataset. It has 129 items in 67,557 transactions, and the average length of each transaction is 43. This database is a binary database, thus the quantity value is randomly given to each item for each transaction in the database. In the experiments, the database is divided into 2 and 5 different databases for constructing 2 and 5 sub-MFFP trees. The execution time of the proposed iMFFP-tree algorithm, the PFPTC algorithm [Lan and Qiu, 05], and the one-step MFFP-tree algorithm [Hong et al., 12] was compared for different minimum support thresholds. The MFFP-tree algorithm directly processed the database in a batch way. The PFPTC algorithm extracted the branches one by one for integration. The proposed iMFFP-tree algorithm is to extract maximal counts of the nodes, forming the extracted branches for integration. The results are shown in Figure 10. It is obvious to see from Figure 10 that the proposed algorithm had a better performance than the PFPTC algorithm for tree integration while the minimum support threshold was less than 28%. The batch MFFP-tree algorithm had better performance than the other two since it did not require the execution time to merge the branches from the tree structures. When the minimum support threshold was larger than 28%, the number of frequent items became few, thus making the execution time of the above approach nearly the same. This is further discussed below.

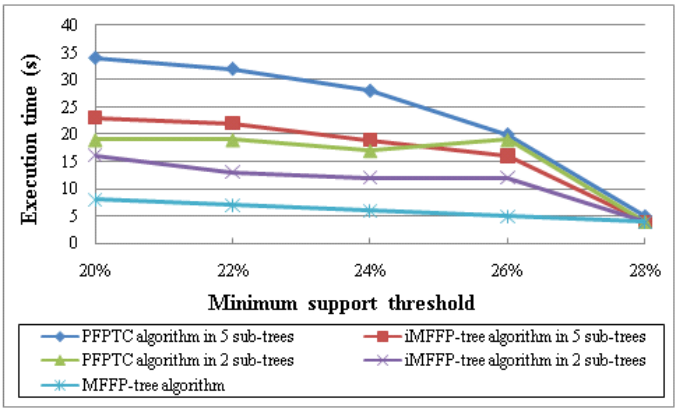

Figure 10: The comparison of execution time for different approaches

In the tree structure, the fuzzy frequent 1-itemsets were kept in the Header_Table as an index to derive longer fuzzy frequent itemsets. The numbers of fuzzy frequent 1-itemsets were then compared to show that multiple regions per item could generate

more fuzzy frequent 1-itemsets than a single region (using the maximal cardinality). The results are shown in Figure 11.

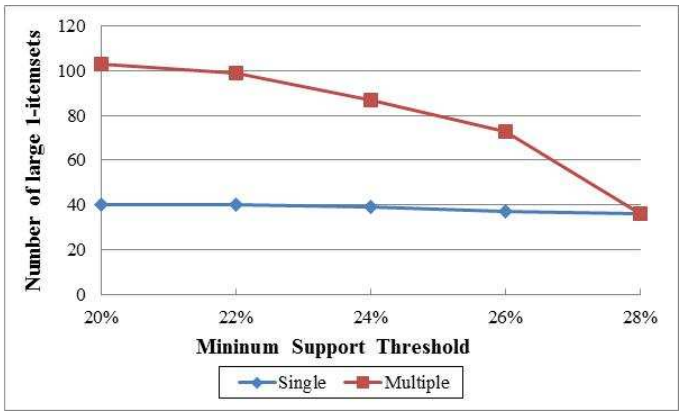

Figure 11: The comparison of fuzzy frequent 1-itemsets

From Figure 11, it is obvious to see that when the minimum support threshold was set higher than the 28%, few fuzzy frequent 1-itemsets were derived from the tree structure. Thus, the execution time of the proposed iMFFP-tree algorithm took nearly the same time as the PFPTC algorithm from Figure 10. The proposed iMFFP-tree algorithm could efficiently merge the sub-trees when the minimum support threshold was set lower. Generally, more rules could provide more information to decision makers, but it may also cause the information overloading problem.

Since the fuzzy frequent itemsets could be generated from the tree structure by the MFFP-growth mining algorithm [Hong et al., 12], the numbers of tree nodes were also compared to show that the adopted multiple regions could generate more rules than the single region (the maximal cardinality one) from the combination of tree nodes. The results are shown in Figure 12.

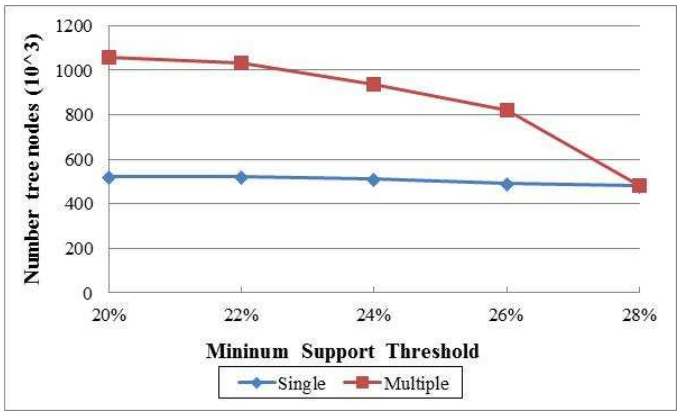

Figure 12: The comparison of tree nodes

6 Conclusions and Discussions

In traditional data mining approaches, most of them usually handle binary databases to find desired information. The discovered knowledge, however, is represented in a statistical or a numerical way. In real-world applications, it is better to state those rules in natural language. Fuzzy-set theory can more reflect human thinking to increase the flexibility for making significant decision. Fuzzy data mining approaches can thus derive better representation of knowledge than traditional rules in binary databases.

In the past, a multiple fuzzy frequent pattern tree (MFFP-tree) algorithm was designed to discover desired rules from quantitative databases. It selects multiple representative linguistic terms for each item in the mining process, thus generating more complete rules than the maximum cardinality one. The designed MFFP-tree algorithm, however, can handle one database in a batch way. In real-world applications, the information from several branches can be integrated into effective knowledge for a parent industry to make correct decision. In this paper, an integrated MFFP-tree (iMFFP-tree) algorithm for merging several sub-MFFP trees into one has been proposed. The branches in a sub-MFFP tree are efficiently extracted and integrated into the iMFFP-tree in sequence. Experimental results also show that the proposed iMFFP-tree algorithm has a better performance than the PFPTC algorithm for generating the multiple fuzzy frequent itemsets. In summary, the contributions of the proposed algorithm are mentioned below.

1. The proposed iMFFP-algorithm handles the multiple fuzzy regions for generating more complete rules than the maximum cardinality one.
2. The proposed iMFFP-tree algorithm can efficiently extract the branches from the designed tree structure and merge them into an integrated tree.
3. The representation of linguistic rules is more familiar to users in real-world applications. In the proposed approach, the intersection operation is used for generating the complete rules for human beings.
4. The proposed iMFFP-algorithm can efficiently derive both global and local association rules.

In the future, we will attempt to design more efficient data structures for helping various kinds of fuzzy data mining. We will also make experiments on larger datasets to validate its scalability.

References

[Agrawal et al., 93] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases, The International Conference on Management of Data (1993), 207-216.

[Agrawal and Srikant, 94] Agrawal, R. and Srikant, R.: Fast algorithms for mining association rules in large databases, The International Conference on Very Large Data Bases (1994), 487-499.

[Agrawal and Srikant, 95] Agrawal, R. and Srikant, R.: Mining sequential patterns, The International Conference on Data Engineering (1995), 3-14.

[Bayardo] Bayardo, R.: UCI repository of machine learning databases, Available: http://fimi.ua.ac.be/data/connect.dat.

[Berzal et al., 01] Berzal, F., Cubero, J. C., Marín, N., Serrano, J. M.: Tbar: An efficient method for association rule mining in relational databases, Data and Knowledge Engineering, (2001), 37(1), 47-64.

[Chan and Au, 97] Chan, K. C. C. and Au, W. H.: Mining fuzzy association rules, The International Conference on Information and Knowledge Management (1997), 209-215.

[Chen et al., 96], Chen, M. S., Han, J., and Yu, Philip S.: Data mining: An overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering (1996), 8(6), 866-883.

[Delgado et al., 03] Delgado, Miguel F., Marín, N., Sánchez, D., Vila, M.-A. A.: Fuzzy association rules: General model and applications, IEEE Transactions on Fuzzy Systems (2003), 11(2), 214-225.

[Ferrer-Troyano et al., 05] Ferrer-Troyano, F. J., Aguilar-Ruiz, J. S., Riquelme, J. C.: Incremental rule learning and border examples selection from numerical data streams, Journal of Universal Computer Science (2005), 11(8), 426-1439

[Han et al., 04] Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Mining and Knowledge Discovery (2004), 8(1), 53-87.

[Hong et al., 04] Hong, T. P., Kuo, C. S., Wang, S. L.: A fuzzy aprioritid mining algorithm with reduced computational time (2001), IEEE International Conference on Fuzzy Systems, 360-363.

[Hong et al., 08] Hong, T. P., Lin, C. W., Wu, Y. L.: Incrementally fast updated frequent pattern trees, Expert Systems with Applications (2008), 34(4), 2424-2435.

[Hong and Wu, 11] Hong, T. P. and Wu, C. H.: An improved weighted clustering algorithm for determination of application nodes in heterogeneous sensor networks, Journal of Information Hiding and Multimedia Signal Processing (2011), 2(2), 173-184.

[Hong et al., 12] Hong, T. P., Lin, C. W., Lin, T. C.: The MFFP-tree fuzzy mining algorithm to discover complete linguistic frequent itemsets, Computational Intelligence (2012), DOI: 10.1111/j.1467-8640.2012.00467.x.

[Hu et al., 99], Hu, K., Lu, Y., Zhou L., Shi, C.: Integrating classification and association rule mining: A concept lattice framework, The International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing (1999), 443-447.

[Janikow, 98] Janikow, C. Z.: Fuzzy decision trees, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (1998), 28(1), 1-14.

[Jain and Stallings, 78] Jain, R., Stallings, W.: Comments on "Fuzzy set theory versus Bayesian statistics", IEEE Transactions on System, Man, and Cybernetics (1978), 8(4), 332-333.

[Kandel, 92] Kandel, A., Fuzzy expert systems (1992).

[Kuok et al., 98] Kuok, C. M., Fu, A., Wong, M. H.: Mining fuzzy association rules in databases, SIGMOD Record (1998), 41-46.

[Lan and Qiu, 05] Lan, Y. J. and Qiu, Y.: Parallel frequent itemsets mining algorithm without intermediate result, The International Conference on Machine Learning and Cybernetics (2005), 2102-2107.

[Lan et al., 11] Lan, G. C., Hong, T. P., Vincent Tseng S.: Discovery of high utility itemsets from on-shelf time periods of products, Expert Systems with Application (2011), 38(5), 5851-5857.

[Lent et al., 97] Lent, B., Swami, A., Widom, J.: Clustering association rules, The International Conference on Data Engineering (1997), 220-231.

[Li and Hu, 11] Li, G. Y. and Hu, Q. B.: Framework for weighted association rule mining from boolean and fuzzy data, The International Conference on Internet Technology and Applications (2011), 1-4.

[Lin et al., 09] Lin, C. W., Hong, T. P., Lu, W. H.: The Pre-FUFP algorithm for incremental mining, Expert Systems with Applications (2009), 36(5), 9498-9505.

[Lin et al., 10a] Lin, C. W., Hong, T. P., Lu, W. H.: Linguistic data mining with fuzzy FP-trees, Expert Systems with Applications (2010), 37(6), 4560-4567.

[Lin et al., 10b] Lin, C. W., Hong, T. P., Lu, W. H.: An efficient tree-based fuzzy data mining approach, International Journal of Fuzzy Systems (2010), 12(2), 150-157.

[Liu et al., 01] Liu, F., Lu, Z., Lu, S.: Mining association rules using clustering, Intelligent Data Analysis (2001), 5(4), 309-326.

[Mahmoudi et al., 11] Mahmoudi, E. V., Sabetnia, E., Torshiz, M. N., Jalali, M., Tabrizi G. T.: Multi-level fuzzy association rules mining via determining minimum supports and membership functions, The International Conference on Intelligent Systems, Modeling and Simulation (2011), 55-61.

[Mishra et al., 11] Mishra, S., Mishra D., Satapathy, S. K.: Fuzzy pattern tree approach for mining frequent patterns from gene expression data, The International Conference on Electronics Computer Technology (2011), 359-363.

[Papadimitriou and Mavroudi, 05] Papadimitriou, S. and Mavroudi, S.: The fuzzy frequent pattern tree, The WSEAS International Conference on Computers (2005), 1-7.

[Park et al., 95] Park, J. S., Chen, M. S., Yu, Philip S.: Using a hash-based method with transaction trimming for mining association rules, IEEE Transactions on Knowledge and Data Engineering (1995), 9(5), 813-825.

[Pulkkinen and Koivisto, 10] Pulkkinen P. and Koivisto H. J.: A dynamically constrained multiobjective genetic fuzzy system for regression problems, IEEE Transactions on Fuzzy Systems (2010), 18(1), 161-177.

[Srikant and Agrawal, 96] Srikant, R. and Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements, The International Conference on Extending Database Technology: Advances in Database Technology (1996), 3-17.

[Sucahyo and Gopalan, 05] Sucahyo, Y. G. and Gopalan, R. P.: Building a more accurate classifier based on strong frequent patterns, Lecture Notes in Computer Science (2005), vol. 3339, 1036-1042.

[Senge and Hüllermeier, 11] Senge, R. and Hüllermeier, E.: Top-down induction of fuzzy pattern trees, IEEE Transactions on Fuzzy Systems (2011), 19(2), 241-252.

[Wang et al., 05] Wang, S., Chung, Korris F. L., Shen, H.: Fuzzy taxonomy, quantitative database and mining generalized association rules, Intelligent Data Analysis (2005), 9(2), 207-217.

[Woźniak and Krawczyk, 12] Woźniak, M. and Krawczyk, B.: Combined classifier based on feature space partitioning, International Journal of Applied Mathematics and Computer Science (2012), 22(4), 69-80.

[Wang et al., 12] Wang, X. Z., Dong, L. C., Yan, J. H.: Maximum ambiguity-based sample selection in fuzzy decision tree induction, IEEE Transactions on Knowledge and Data Engineering (2012), 24(8), 1491-1505.

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

[Zaman and Hirose, 11] Zaman, M. F. and Hirose, H.: Classification performance of bagging and boosting type ensemble methods with small training sets, New Generation Computing (2011), 29(3), 277-292.