

# Proportion Frequency Occurrence Count with Bat Algorithm (FOCBA) For Rule Optimization and Mining of Proportion Equivalence Fuzzy Constraint Class Association Rules (PEFCARs)

<sup>1</sup>R. Ramesh, Saravanan V.<sup>2</sup>

<sup>1</sup>\* Assistant Professor , Department of computer science , Sri Krishna arts and science college ,Coimbatore ,Tamilnadu , Pin-641 008 ,

<sup>2</sup>Professor & Dean, Department of Computer Applications, Sri Venkateswara College of Computer Applications and Management, Coimbatore, India

## Article Info

### Article history:

Received Dec 12<sup>th</sup>, 2017

Revised Apr 20<sup>th</sup>, 2018

Accepted Jun 26<sup>th</sup>, 2018

### Keyword:

Associative classification,  
Fuzzy Constraint Class  
Association Rules (FCARs),  
Enhanced Proportion  
Equivalence Fuzzy Class Rule  
tree (EPEFCR-tree),  
Euclidean Distance between  
two Obidsets (EDO),  
Proportion Frequency  
Occurrence Count with Bat  
Algorithm (PFOCBA),  
Rule pruning, and optimization.

## ABSTRACT

Fuzzy Class Association Rules (FCARs) play an important role in decision support systems and have thus been extensively studied. Mining the important rules in FCARs becomes very difficult task, so Enhanced Equivalence Fuzzy Class Rule tree (EEFCR-tree) algorithm is proposed in this work. However, a major weakness of FCARs Miner is that when the number of constrained rules in a given class dominates the total constrained rules; its performance becomes slower than the normal method. To solve this problem this paper proposes a Proportion of Constraint Class Estimation (PPCE) algorithm for mining Enhanced Proportion Equivalence Fuzzy Constraint Class Association Rules (EPEFCARs) in order to save memory usage, run time and accuracy. Then, Proportion Frequency Occurrence count with Bat Algorithm (PFOCBA) is proposed for pruning rules which much satisfying the class constraints. Finally, an efficient algorithm is proposed for mining PEFCARs rules. Experimental results show that the proposed EPEFCR-tree algorithm is more efficient than Enhanced Equivalence Fuzzy Class Rule tree (EEFCR-tree), Novel Equivalence Fuzzy Class Rule tree (NECR-tree) Miner results are measured in terms of run time, accuracy and memory usage. Experiments show that the proposed method is faster than existing methods.

## Corresponding Author:

R.Ramesh,  
Department of computer science ,  
Sri Krishna arts and science college ,Coimbatore.  
mail-id : ramesh4gift@gmail.com

## 1. Introduction

Association Rule Mining (ARM) is widely analysed because of its application in several areas for instance market basket analysis, protein sequencing, medicine, census data processing, and fraud detection. Various subjects have attracted researchers, comprising mining association rules [1-2] and Classification Based on Association Rules (CBARs) [3-4]. In the ARM general problem is frequent pattern mining. The techniques like Apriori technique [5], the Frequent Pattern (FP) growth technique is used for mining frequent itemsets in incremental databases [6-7].

In the ARM, rule-based classification is also considering an important step. Consequently, certain techniques have been proposed for mining classification rules dependent upon ARM. Examples are classification dependent upon predictive association rules [8], multi-label associative classification [9], multi-class classification based on association rules [10], associative classifier based on maximum entropy [11], and the usage of the equivalence class rule tree [12]. Veloso et al [13] proposed lazy associative classification that

varied from CARs in that it utilized rules mined from the dataset of an unidentified object for foreseeing the class in preference to utilizing the ones mined from the entire dataset. All the above mentioned approaches focused on the design of the techniques for mining CARs, on the other hand it didn't converse much pertaining to their mining time.

Nevertheless, the entire set of CARs is extremely huge as it comprises numerous redundant or unimportant rules. These inoperable rules waste memory space and reduce the performance of a classifier, nonetheless they contain a negative effect on decision-making. With the aim of resolving this issue, effort is dedicated to pruning redundant rules or ranking rules. Nguyen and Nguyen [14] proposed a new effective pruning method to construct a quicker classifier dependent upon CARs. Initially, create a form named Lattice of Equivalence Class Rules (LECR) and present a technique for fast mining CARs. Secondly, present a technique to prune rules, which are redundant in LECR. Experimentation results prove that research method is very effective compared to the one dependent upon the Equivalence Class Rules-tree (ECR-tree). The rule sets produced by two methods, ECR-tree and LECR, are similar, consequently the accurateness doesn't changed.

Azmi and Berrado [15] proposed a novel method for CARs pruning dependent upon Lasso regularization. In this method we present to exploit variable selection capability of Lasso regularization to prune less interesting rules. The experiment proves that the presented methodology provides superior results when compared to CBA in regard to the number in addition to the quality of the attained rules after pruning.

Gonzales et al [16] presented a novel post-processing technique for pruning CARs by a grouping of data and an evolutionary technique called Genetic Relation Technique (GRA). It is performed in two steps. In the initial step, the rules are pruned based upon their matching degree with data, and in the next step, GRA chooses the lot of interesting rules by utilizing the distance among them.

Even though rule pruning as well as rule ranking methods could support to remove redundant rules and get significant rules, enhancing classifier performance, there is small achieve mention the subject of finding optimal or beneficial rules from an end user's opinion. This research concentrates on mining Proportion Equivalence Fuzzy Constraint Class Association Rules (PEFCARs) with respect to class constraints. The contributions of this research are specified in this way:

(1) In order to proficiently mining PEFCARs with proportion class constraints, a new tree structure named the Enhanced Proportion Equivalence Fuzzy Class Rule tree (EPEFCR-tree) is presented. Every node in EPEFCR -tree comprises attribute values and their associated information.

(2) For mining PEFCARs efficiently and lesser memory consumption, Proportion of Constraint Class Estimation (PPCE) technique is presented.

(3) Proportion Frequency Occurrence count with Bat algorithm (PFOCBA) is proposed for rapidly pruning nodes, which are not capable of producing rules fulfilling the class constraints are designed.

(4) Proposed EPEFCR-tree technique is very capable when compared to Enhanced Equivalence Fuzzy Class Rule tree (EEFCR-tree), Novel Equivalence Fuzzy Class Rule tree (NECR-tree) Miner in regard to run time, accurateness and memory usage.

## 2. Literature review

Mohamed et al [17] proposed a novel method of CARs pruning dependent upon Lasso regularization. It exploits variable selection capability of Lasso regularization to prune less interesting rules, which is it does variable selection by lessening the coefficients equivalent to inappropriate variables to be equivalent to zero. Regularization is utilized to perform supervised learning model from overfitting the training sample by regulating the model complexity.

Subbulakshmi et al [18] utilizes Incremental Constraint Class Association Rule (ICCAR) Technique by building Incremental Constraint Class Rule (ICCR) Tree to produce the Constraint Class Association Rules. According to incremental data, the ICCR Tree is brought up to date for the novel set of records deprived of re-scanning the dataset, it takes less memory while matched up with the previous associative classifiers.

Nguyen et al [19] modified equivalence class rules tree (MECR-tree) is formed from the real dataset. While records are put in, nodes on the tree are kept up to date by altering their information comprising Obidset, count, and pos. Next, the notion of pre-large itemsets is used to evade re-scanning the real dataset. Lastly, a theorem is presented to rapidly prune nodes, which couldn't generate rules in the tree update process.

Nguyen et al [20] proposed a novel Class-Association Rule with Interestingness Measure (CARIM) for mining CARs dependent upon various interestingness measures. It utilizes a Modified MECR-tree structure for keeping the associated information of item sets in the nodes, therefore speeding up the process of generation of rules. It could be effortlessly prolonged to incorporate certain measures together for ranking of rules.

Shimada et al [21] proposed a Genetic Network Programming (GNP) technique for mining CAR mining. GNP is one among the evolutionary optimization methods that utilizes directed graph structures as genes. Moreover, the technique suits class association rule mining from dense databases; here numerous repeatedly occurring items are identified in every tuple. Users could define conditions of taking important CARs.

Sarno et al [22] proposed a Multi-Level Class Association Rule Learning (ML-CARL) to identify fraud in business process. It is aided by the Semantic Web Rule Language (SWRL) Rule that utilized to ensure the conformance among the typical business process model Standard Operating Procedure (SOP) and event logs. Additionally, Multi Attribute Decision Making (MADM) is consistent to compute rates of anomaly in keeping with the expert valuation as well as the occurrences of anomalies attribute.

Mabu et al [23] proposed a new fuzzy class-association rule mining technique dependent upon GNP was for identifying network intrusions. By merging fuzzy set theory with GNP, this technique could handle the mixed database, which encompasses discrete as well as continuous attributes in addition excerpt numerous significant class association rules, which contribute to improving detection capability. So, the above stated technique could be submissively used to misuse as well as anomaly detection in network-intrusion-detection problems.

Kharche et al [24] proposed GA based Fuzzy Class Association Rule Mining (GA-FCAR) for anomaly detection system. GA is utilized to excerpt the rules that are required for anomaly detection system. The usage of the fuzzy logic handles mixed categories of attribute and avoid sharp boundary problem. The GA-FCAR method brings higher detection rate and less false positive rate that are two significant conditions for security systems. According to anomaly detection, the process produces higher detection rate and sensible false positive rate deprived of prior information of attack signatures that is a momentous enhancement over other methods.

Jia et al [25] proposed a new fuzzy system dependent upon Class Association Rule (CAR) known as FS\_CARs. It utilizes CARs to generate fuzzy systems rules that resolves the inflexible issue of the curse of dimensionality in fuzzy system as well as increases the rule's interpretability. It uses trapezoid as membership function. Therefore for prediction, the fuzzy rule base is constructed based on CARs provides a comprehensive solution to the whole data in input space.

Abu et al [26] proposed a novel pruning approach, which remove redundant and irrelevant rules all through constructing the classifier are applied. These pruning actions eliminate any rule, which either contains no training case coverage or covers a training case deprived of the obligation of class likeness among the rule class and that of the training case. This facilitates huge coverage for every rule and decreases overfitting in

addition to build exact and moderated size classifiers. Alongside, a new class assignment technique dependent upon multiple rules is presented that applies group of rule to create the prediction decision.

### 3. Proposed methodology

This research proposes a new classification method that incorporates FCARs and Technique. A fuzzy discretization method is used to transform the training set, essentially quantitative attributes, to a format suitable for association rule mining. Consequently estimation technique is proposed for measure the ratio of the every class, Enhanced Proportion Equivalence Fuzzy Class Rule tree (EPEFCR-tree) method. A Proportion Frequency Occurrence count with Bat algorithm (PFOCBA) is adapted for automatic frequency occurrence count and PEFCARs are mined consequently. The compatibility among the mined rules and frequent patterns is taken to build a group of vectors that are utilized to produce a classifier. The results outcomes prove that proposed EPEFCR-tree technique provides a highly-qualified source of discrimination knowledge, which could take ably affect the prediction power of the final classifier.

#### 1.1. Preliminary concepts

Take DS be a dataset with  $d$  attributes  $\{Att_1, \dots, Att_d\}$  and  $n$  signify records (objects), here every record contains an Object Identifier (OID)[27]. Assume  $C = \{c_1, c_2, \dots, c_k\}$  is a list of class labels ( $k$  is the number of classes). Presume Constraint Class ( $Con_C$ ) is a subset of  $C$  comprising specific class labels taken by end users. A particular value of an attribute  $Att_i$  and the  $m^{th}$  record is signified by  $att_{mi}$  ( $m \in [1, n], i \in [1, d]$ ) and a particular value of class  $C$  is represented by  $c_x$  ( $x \in [1, k]$ ).

**Definition 1:** An item is defined as an attribute and a particular fuzzy value for that attribute, represented by  $(Att_i, fatt_{mi})$  ( $m \in [1, n], i \in [1, d]$ ).

**Definition 2:** An itemset is a set of items, signified by  $\{(A_i, fatt_{mi}), \dots, (A_j, fatt_{mj})\}$  ( $m \in [1, n], i, j \in [1, d]$  and  $i \neq j$ )

**Definition 3:** Fuzzy Class Association Rule (FCAR) contains the structure  $\{(Att_i, fatt_{mi}), \dots, (Att_j, fatt_{mj})\} \rightarrow c_x$ , here  $\{(Att_i, fatt_{mi}), \dots, (Att_j, fatt_{mj})\}$  is an itemset and  $c_x \in C$  is a class label.

**Definition 4:** In the FCARs, specified a group of attributes  $\{Att_1, \dots, Att_d\}$  and a fuzzy partition  $P_f$  described for every attribute  $Att_{df}$ , the single item is described as the duo of  $I_f = (Att_d, FAtt_f)$ , here  $FAtt_f$  is one among the fuzzy values described in the partition  $P_f$  of variable  $Att_f, f = 1, \dots, F$ . A common fuzzy CAR for classification is denoted as:

$$FCAR_r: FAnt_r \rightarrow C_r$$

here  $C_r$  is the class label chosen for the rule amongst the set  $C = \{C_1, \dots, C_k\}$  of probable classes and  $FAnt_r$  is a conjunction of items.

$$FAnt_r: Att_1 \text{ is } FAtt_{1,r} \dots \text{ And } Att_F \text{ is } FAtt_{F,r}$$

here  $FAtt_{F,r}$  is the fuzzy value utilized for variable  $A_f$  in rule  $FCAR_r$ .

**Definition 5:** The real occurrence ActOcc(R) of rule R in DS is known as the number of records of DS, which match R's antecedent.

**Definition 6:** According to the CAR analysis, support and confidence are the supreme measures to find out the strength of a CAR. Support and confidence could be denoted for a fuzzy rule  $FCAR_r$  in this manner:

$$FSup(FAnt_r \rightarrow C_r) = \frac{\sum_{Att_n \in C_r} M_r(Att_n)}{N} \quad (1)$$

$$FConf(FAnt_r \rightarrow C_r) = \frac{\sum_{Att_n \in C_r} M_r(Att_n)}{\sum_{Att_n \in TS} M_{ant_r}(Att_n)} \quad (2)$$

Here TS is known as the training set, N is called the number of objects in TS,  $M_r(Att_n)$  is called the matching degree of rule  $FCAR_r$  and  $M_{ant_r}(Att_n)$  is known as the matching degree of the entire rule that contain the antecedent equivalent to  $FAnt_r$ .

## 1.2. Mining constrained class fcar

An improved tree structure named the Enhanced Equivalence Fuzzy Class Rule tree (EEFCR-tree) is presented for capably mining of FCARs with class constraints. Fuzzy items are produced by discretizing the input variables and describing strong fuzzy partitions on the intervals resultant from these discretizations. In EEFCR-tree, every node encompasses one itemset in the company of the subsequent information:

(a) ( $Obidset_1, Obidset_2, \dots, Obidset_k$ ): every  $Obidset_i$  is a group of Object IDentifiers (OID) which comprise itemset as well as class  $c_i$ . Remind that k is the amount of classes in the dataset.

(b) Pos: maintains the position of the class with the maximum cardinality of  $Obidset_i$ , that is to say

$$pos = \operatorname{argmax}_{i \in \{1, k\}} \{|Obidset_i|\}$$

(c) Total: keeps the summation of cardinalities of all  $Obidset_i$ , i.e.,  $total = \sum_{i=1}^k |Obidset_i|$

In EEFCR-tree, itemset is transformed into the form att, values, Fvalues for simply programming, here:

(1) att: It is a list of attributes.

(2) Values: It is known as a list of values, each of which is enclosed in one attribute in att.

(3) Fvalues: It is called a list of fuzzy values, each of which is contained in one attribute in att.

**Example 1:** In the subsequent, would define exhaustively the operations carried out in the four scans with the aim of an example of application in which adopt the training set presented in Table 1 with seven objects, four attributes ( $X_1, X_2, X_3$ , and  $X_4$ ), and three classes (1, 2, and 3).

Table 1: Dataset with four attributes ( $X_1, X_2, X_3$ , and  $X_4$ )

OID	$X_1$	$X_2$	$X_3$	$X_4$	Class
1	20	20	0	10	$C_1$
2	25	-60	10	80	$C_3$
3	-25	40	100	40	$C_1$
4	75	60	35	110	$C_2$
5	20	80	100	75	$C_2$
6	30	90	75	10	$C_3$
7	120	50	75	-25	$C_1$

In this table 1, values are signified by five fuzzy regions: Low Low (LL), Low (L), Middle (M), High (H), Very High (VH). Therefore, three fuzzy membership values are created for every item amount in keeping with the predefined membership functions. In this research, fuzzy membership function  $\mu(x)$  [28-29] is used at every level. Nevertheless value of a, b and m are diverse at every level, a=lower bound of the attribute value, m=middle value of the attribute and b=upper limit of the attribute value [30].

$$\mu(x) = \begin{cases} 0, & x < 0 \text{ or } x > b \\ \frac{(x - a)}{(b - a)}, & x \geq a, x < m \\ \frac{(b - x)}{(b - m)}, & x \geq m, x < b \end{cases} \tag{3}$$

At every level, there are totally 5 regions with three low, middle, high. For level 1 values of a,b,m at every region is conversed in table 2.

Table 2. Counts of the fuzzy regions

Fuzzy value	Fuzzy count	Fuzzy support
A <sub>1,1</sub> (LL)	0.98	0.14
A <sub>1,2</sub> (L)	2.1	0.30
A <sub>1,3</sub> (M)	2.38	0.34
A <sub>1,4</sub> (H)	0.49	0.07
A <sub>1,5</sub> (VH)	0.98	0.14
A <sub>2,1</sub> (LL)	0.98	0.14
A <sub>2,2</sub> (L)	0.49	0.07
A <sub>2,3</sub> (M)	2.66	0.38
A <sub>2,4</sub> (H)	1.75	0.25
A <sub>2,5</sub> (VH)	0.98	0.14
A <sub>3,1</sub> (LL)	1.61	0.23
A <sub>3,2</sub> (L)	0.98	0.14
A <sub>3,3</sub> (M)	0.42	0.06
A <sub>3,4</sub> (H)	2.03	0.29
A <sub>3,5</sub> (VH)	2.03	0.29
A <sub>4,1</sub> (LL)	0.98	0.14
A <sub>4,2</sub> (L)	1.82	0.26
A <sub>4,3</sub> (M)	2.1	0.30
A <sub>4,4</sub> (H)	1.12	0.16
A <sub>4,5</sub> (VH)	2.1	0.30

Table 3. The fuzzy values associated with the highest membership degree and the corresponding fuzzy objects for each pattern in the example dataset

OID	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	Class
1	A <sub>1,2</sub>	A <sub>2,3</sub>	A <sub>3,1</sub>	A <sub>4,2</sub>	C <sub>1</sub>
2	A <sub>1,3</sub>	A <sub>2,1</sub>	A <sub>3,1</sub>	A <sub>4,4</sub>	C <sub>3</sub>
3	A <sub>1,1</sub>	A <sub>2,3</sub>	A <sub>3,5</sub>	A <sub>4,3</sub>	C <sub>1</sub>
4	A <sub>1,4</sub>	A <sub>2,4</sub>	A <sub>3,2</sub>	A <sub>4,5</sub>	C <sub>2</sub>
5	A <sub>1,2</sub>	A <sub>2,4</sub>	A <sub>3,5</sub>	A <sub>4,3</sub>	C <sub>2</sub>
6	A <sub>1,3</sub>	A <sub>2,5</sub>	A <sub>3,4</sub>	A <sub>4,2</sub>	C <sub>3</sub>
7	A <sub>1,5</sub>	A <sub>2,3</sub>	A <sub>3,4</sub>	A <sub>4,1</sub>	C <sub>1</sub>

Fuzzy support (Fsup ) of every fuzzy value A<sub>f,j</sub>. The fuzzy support is calculated as:

$$Fsup(A_{f,j}) = \frac{\sum_{n=1}^N A_{f,j}(X_{f,n}) \cdot Obidset_{1,pos}}{N} \tag{4}$$

The Fuzzy confidence threshold is measured by taking the imbalance ratio among every class and the majority class in this way

$$\min FConf_{Conc}(A_{f,j}) = Fconf \cdot \frac{N_{Conc} \cdot Obidset_{1,pos}}{N_{MajorityClass}} \quad (5)$$

**Example 2:** Itemset  $X = (X_1, A_{1,2}; X_2, A_{2,3})$  is signified as  $X = 3 \times A_{1,2}A_{2,3}$ . A bit representation is utilized for itemset attributes. Attributes  $X_1X_2$  could be denoted by 11 in this bit representation, consequently the value of these attributes is 3. Bitwise operations are utilized to merge itemsets rapidly.

**Example 3:** In Table 1, itemset  $X = (X_1, A_{1,2}; X_2, A_{2,3})$  is enclosed in objects 1 of which is in class 1. So, node  $\{(X_1, A_{1,2}), (X_2, A_{2,3})\}, \{1, \emptyset, \emptyset\}$  will be added to the EFECR-tree in case of minFSup were 2. This node contains  $Obidset_1 = 1$  (i.e., objects 1 itemset X and class 1),  $Obidset_2 = \emptyset$  for short (i.e., no objects comprise itemset X as well as class 2),  $Obidset_3 = \emptyset$  (that is to say no objects comprise itemset X as well as class 3),  $pos = 2$  (a line under position 2 of list  $Obidset_i$  and total = 2. The pos value is 2 as the count of  $Obidset$  for class 1 is the upper limit. At that time, an effective technique is proposed for mining FCARs with class constraints dependent upon the specified theorems is presented in this manner:

**Theorem 1:** Specified two nodes  $X_1$  and  $X_2$ , if  $X_1.att = X_2.att$  and  $X_1.Fuzzyvalues \neq X_2.Fuzzy$  values, at that point  $X_1.Obidset_i \cap X_2.Obidset_i = \emptyset$ .

**Theorem 2:** Specified two nodes  $X_1$  and  $X_2$ , if  $X_1$  is a parent node of  $X_2$  and  $|X_2.Obidset_{X_1,pos}| = |X_1.Obidset_{X_1,pos}|$ , at that point  $X_2.pos = X_1.pos$ .

**Example 4:** Let two nodes  $1 \times A_{1,1}(3, \emptyset, \emptyset)$  and  $1 \times A_{1,2}(1,5, \emptyset)$ , whose attribute is  $att = 1$ . It could be viewed that  $Obidset_i(A_{1,1}A_{1,2}) = Obidset_i(A_{1,1}) \cap Obidset_i(A_{1,2}) = \emptyset$ . Likewise,  $2 \times A_{1,1}A_{2,3}(3, \emptyset, \emptyset) \cap A_{1,2}A_{2,3}(1,5, \emptyset)$  as they contain the similar attributes  $X_1X_2$  nevertheless diverse values ( $A_{1,1}A_{2,3}$  and  $A_{1,2}A_{2,3}$ ).

**Example 5:** Node  $Z = 8 \times A_{1,1}A_{1,2}A_{2,3}A_{3,1}A_{3,5} A_{4,3}A_{4,2}(3, \emptyset, \emptyset)$  contains two parent nodes  $X = 4 \times A_{1,1}A_{2,3}A_{3,5}A_{4,3}(3, \emptyset, \emptyset)$  and  $Y = 4 \times A_{1,2}A_{2,3}A_{3,1}A_{4,2}(1, \emptyset, \emptyset)$ .  $Y.pos = X.pos = 4$  as  $|Y.Obidset_4| = |1| = |3| = |X.Obidset_4|$ .

**Corollary 1.** The condition for adding one node to the EFECR-tree is  $|Obidsetpos_j| = \min FSUP$ .

**Theorem 3:** Specified node X, this node and its child nodes could not produce rules fulfilling  $Conc$  if  $j |X.Obidset_i| < \min FSUP (\forall i \in Conc)$ .

**Theorem 4:** Specified node X, in case this node could not create rules sustaining  $Conc$ , and its child nodes  $Y_j$  contain  $X.pos = Y_j.pos$  and  $|X.Obidset_{X,pos}| = |Y_j.Obidset_{Y_j,pos}|$  at that time its grandchild nodes moreover could not produce rules fulfilling  $Conc$ .

### 1.3. EEFCR-tree algorithm

Dependent upon these four theorems, design an effective EEFCR-tree technique for mining constrained FCARs. According to Theorem 1, it is not essential to merge two nodes with the similar attributes, and according to Theorem 2, don't want to calculate the information for certain child nodes. According to Theorems 3 and 4, don't want to produce certain nodes. Initially, the root node of the EEFCR-tree ( $L_r$ ) encompasses child nodes; every node encompasses a single frequent 1-itemset. Afterwards, the procedure Constraint-FCAR-Miner is called with the parameters  $L_r$ ,  $\min FSUP$ ,  $\min FConf$ , and  $Conc$  to mine the whole

constrained FCARs from dataset DS. The more information of the EFCR is conversed in the current research [31]. The membership value associated is conversed in table 4.

Table 4. Final example

Serial no	Membershi p value	Object set(Class 1,Class 2,Class 3)	Class
1	A <sub>1,1</sub>	(3, ∅, ∅)	C <sub>1</sub>
2	A <sub>1,2</sub>	(1,5, ∅)	C <sub>1</sub> , C <sub>2</sub>
3	A <sub>1,3</sub>	(∅, ∅, 26)	C <sub>3</sub>
4	A <sub>1,4</sub>	(∅, 4, ∅)	C <sub>2</sub>
5	A <sub>1,5</sub>	(7, ∅, ∅)	C <sub>1</sub>
6	A <sub>2,1</sub>	(∅, ∅, 2)	C <sub>3</sub>
7	A <sub>2,3</sub>	(13,7, ∅)	C <sub>1</sub> , C <sub>2</sub>
8	A <sub>2,4</sub>	(∅, 45, ∅)	C <sub>2</sub>
9	A <sub>2,5</sub>	(∅, ∅, 6)	C <sub>3</sub>
10	A <sub>3,1</sub>	(1, ∅, 2)	C <sub>1</sub> , C <sub>3</sub>
11	A <sub>3,2</sub>	(∅, 4, ∅)	C <sub>2</sub>
12	A <sub>3,4</sub>	(7, ∅, 6)	C <sub>1</sub> , C <sub>3</sub>
13	A <sub>3,5</sub>	(3, ∅, ∅)	C <sub>1</sub>
14	A <sub>4,1</sub>	(7, ∅, ∅)	C <sub>1</sub>
15	A <sub>4,2</sub>	(1, ∅, 6)	C <sub>1</sub> , C <sub>3</sub>
16	A <sub>4,3</sub>	(3,5, ∅)	C <sub>1</sub> , C <sub>2</sub>
17	A <sub>4,4</sub>	(∅, ∅, 2)	C <sub>3</sub>
18	A <sub>4,5</sub>	(∅, 1, ∅)	C <sub>2</sub>

In this segment, the example dataset in Table 4 is utilized to demonstrate the general miner process with minFSup = 1.6,

minFConf = 0.4. Figure 1 illustrates the process of NECR tree in the example 1.

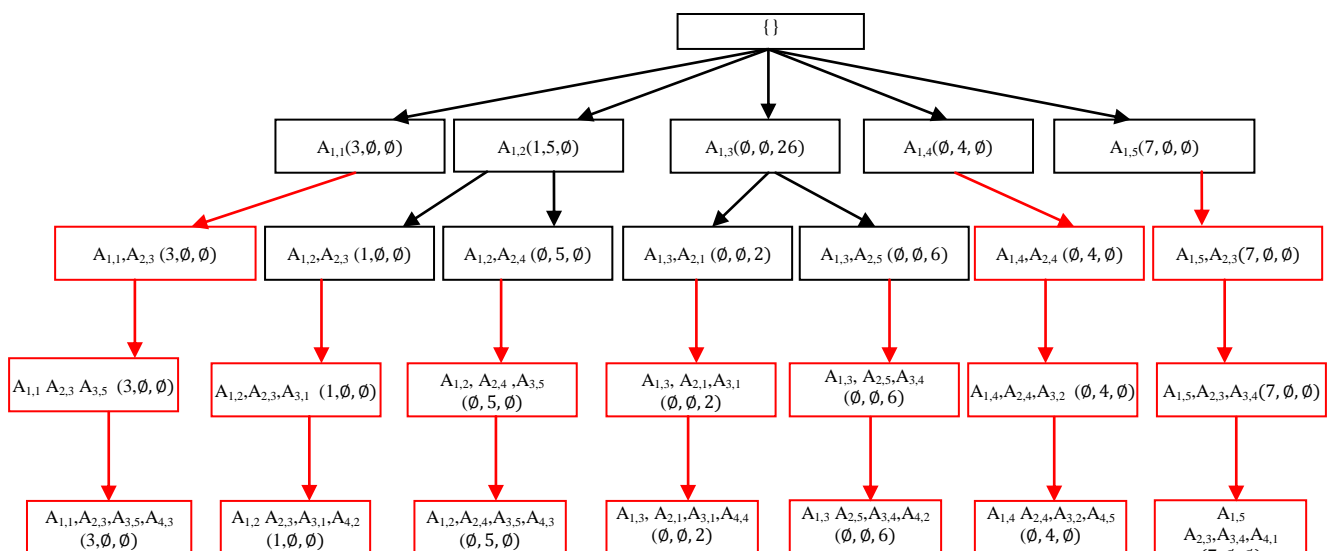


Figure 1. According to the theorems NECR tree

Figure 1 is utilized to demonstrate the tree miner process in keeping with theorem 4 and theorem 3. As stated by theorem 4 in case the parent node A<sub>1,1</sub>(3,∅,∅), and the child nodes A<sub>1,1</sub> A<sub>2,3</sub> (3,∅,∅) couldn't produce



constrained rules, and they contain the similar position pos and the similar Obidset at position pos, at that moment consider  $A_{1,1} A_{2,3} (3, \emptyset, \emptyset)$  and their siblings them from the tree since their grandchild nodes likewise couldn't produce constrained rules. Correspondingly verify them this theorem for the whole tree, those nodes are labeled as red color in the tree. Figure 2 displays the outcomes of this process. In the figure 3 ,  $A_{1,1}(3, \emptyset, \emptyset)$ ,  $A_{1,2}$ ,  $A_{2,3} (1, \emptyset, \emptyset)$  contain the similar class with diverse class, consequently those ought to long-standing as novel node in the tree by gauging the Euclidean distance among two OID is demonstrated in figure 3. Euclidean distance [5] between two points  $Obidset_1$  and  $Obidset_2$  . According to Cartesian coordinates, in case  $Objset(p) = (p_1, p_2, \dots, p_n)$  and  $Objset(q) = (q_1, q_2, \dots, q_n)$  are two points in Euclidean n-space, at that point the distance (d) from p to q.

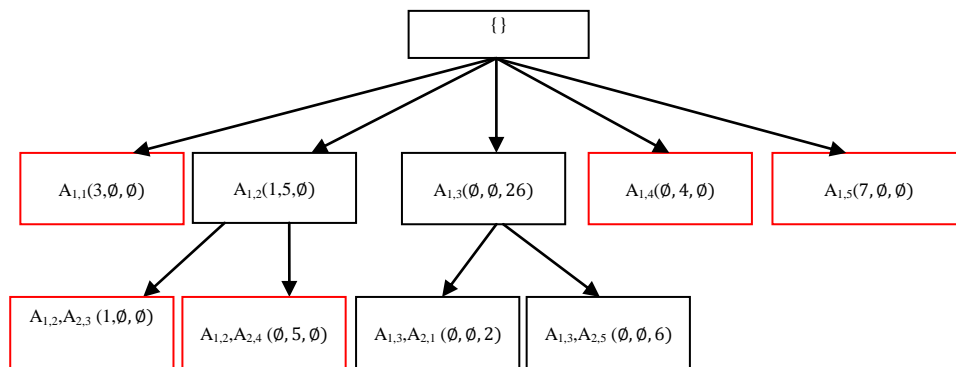


Figure 2. EFCR-tree outcomes

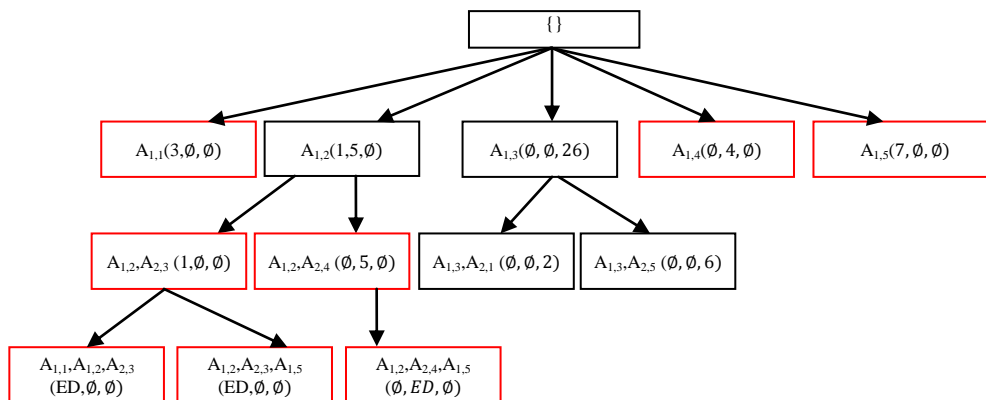


Figure 3. EFCR tree with EOD outcomes

#### 1.4. Proportion of constraint class estimation (PPCE)

This section converse the problem of approximating the ratios of constraint classes the unlabeled testing data that might vary from those in the training data set. There are M classes, and a training sample for every constraint class:

$$X_1^i, \dots, X_{n_i}^i \stackrel{iid}{\sim} Pr_i \tag{6}$$

here  $P_i$  is the  $i^{th}$  constraint class-conditional distribution, and  $X_j^i$  signifies the  $j^{th}$  training sample from constraint class i. Furthermore, there is an unlabeled testing data.

$$X_1^0, \dots, X_{n_0}^0 \sim Pr_0 := \sum_{i=1}^M \pi_i Pr_i \tag{7}$$

derived from a combination of the diverse constraint classes. Where  $\pi_i \geq 0$  and  $\sum_i \pi_i = 1$ . The significant feature of this problem is that the proportions  $\pi_i$  are unidentified and dissimilar from the proportions signified in the training data, with the intention that  $n_i / \sum_l n_l$  is not a sensible estimate. The objective is to guesstimate the  $\pi_i$  exactly, when creating least suppositions on the  $Pr_i$ . One among the main motivation to PPCE is development of a right EPEFCARs classifier for the test data. Presume that there is a joint distribution on labels as well as instances with  $Pr_0$  the marginal distribution on instances,  $Pr_i$  the class-conditional distributions, and  $\pi_i$  the prior distribution on labels. The risk of a classifier  $f : X \rightarrow \{1, \dots, M\}, X \subseteq \mathbb{R}^d$  signifying the data space, might be stated

$$R(f) := Pr_i \pi_i E_i(f) \tag{8}$$

here

$$E_i(f) := Pr_i(\{x: f(x) \neq i\}) \tag{9}$$

The constrained class-conditional errors  $E_i$  is approximated as the training data present examples from every constrained class. Take  $Pr_M, \dots, Pr_1$  probability measures on  $(X, S)$ . Class proportion guesstimate decreases to mixture proportion estimation that is revised. Take  $(X, S)$  is a measurable space, and take F, G, and H are distributions on X so

$$F = (1 - v)G + vH \tag{10}$$

here  $0 \leq v \leq 1$ . Mixture proportion estimation is the subsequent problem: specified ‘iid’ training data of sizes m and n from F and H correspondingly, and no information regarding G, estimate v. certain work completed associated to the general class problem [32] and afterwards used to the problem of classification with label noise [33]. Deprived of further suppositions, v is not a recognisable parameter. Definitely, in case  $F = (1 - v)G + vH$  holds, at that point any other decomposition of the form

$$F = (1 - v + \delta)G' + (v - \delta)H \tag{11}$$

with  $G' = (1 - v + \delta)^{-1} ((1 - v)G + \delta H)$ , and  $\delta \in [0, v)$ , is as well legal. Without any associate of G, couldn’t choose which depiction is the right one. Consequently, the notion is to enforce a condition on G so v turns out to be recognizable.

**Theorem 5:** Assume G,H are a probability distribution, G is irreducible relating to H in case there is no decomposition of the form  $G = \gamma H + (1 - \gamma)F'$ , here  $F'$  is known as the correct probability distribution of the constraint class and  $0 \leq \gamma \leq 1$

**Theorem 6:** Take F, H is probability distributions. In case  $F \neq H$ , there is a distinctive  $v^* \in [0, 1)$  and G so the decomposition

$$F = (1 - v^*)G + v^*H \tag{12}$$

holds, and so that G is irreducible regarding H. In case furthermore describe  $v^* = 1$  while  $F = H$ , after that in all cases,

$$v^* := \max \{ \alpha \in [0,1]: \exists \text{ a distribution } G' \text{ s.t. } F = (1 - \alpha)G' + \alpha H \} \quad (13)$$

According to this outcome, the subsequent is definite.

**Definition 7.** For any two probability distributions F, H, describe

$$v^*(F, H) := \max \{ \alpha \in [0,1]: \exists \text{ a distribution } G' \text{ s.t. } F = (1 - \alpha)G' + \alpha H \} \quad (14)$$

Therefore, G is irreducible regarding H if and only if  $v^*(G, H) = 0$ . Additionally, it is not difficult to prove that for any two distributions F and H,  $v^*(F, H) = F(x)/H(x)$  [33]. These identities make it probable to verify irreducibility in diverse situations. Let's say,  $v^*(G, H) = 0$  when the support of G doesn't comprise the support of H. Although the supports are equivalent, irreducibility could hold like in the case here g and h are two Gaussian densities with distinct means; here the variance of h is no lesser than the variance of g.

(A) For all that  $i = 1, \dots, M$ , each element of  $\text{conv}\{Pr_l, l \neq i\}$  is irreducible respecting  $Pr_i$ .  $\text{conv}\{Q_1, \dots, Q_K\}$  signifies the group of convex sets of  $Q_1, \dots, Q_K$ , specifically, the set of mixture distributions dependent upon  $Q_1, \dots, Q_K$ . In order to illuminate (A), present a second condition, here  $\text{supp}(Q)$  signifies the support of distribution Q.

(B) For all that  $i = 1, \dots, M$ ,  $\text{supp}(Pr_i) \not\subseteq \cup_{l \neq i} \text{supp}(Pr_l)$ . (B) Obviously implies (A) from the description of irreducible.

### 1.5. EPEFCR-tree algorithm

Dependent upon the theorem 5 and theorem 6, design an effective EPEFCR –tree technique for mining restrained PEFCR-tree. According to Theorem 1, needn't to join two nodes with the similar attributes, and according to Theorem 2, don't want to calculate the information for certain child nodes. According to Theorems 3 and 4, don't want to produce certain nodes. According to Theorem 5, needn't to join two attributes with the diverse constraint class, and according to Theorem 6, want to calculate the probability value for certain child nodes. Initially, the root node of the EPEFCR -tree ( $L_r$ ) comprises child nodes; every node encompasses a single frequent 1-itemset. It is eminent that the nodes, which could not produce constrained rules as identified from Theorem 3 are detached from  $L_r$ . At that point, the procedure Constraint-EPEFCR -Miner is called with the parameters  $L_r$ , minFSup, minFConf,  $Con_C$  and probability  $Pr_i$  to mine each and every constrained EPEFCR from dataset D.

The Constraint- EPEFCR -Miner procedure takes every node  $l_i$  with other nodes  $l_j$  in  $L_r$ , with  $j > i$  (lines 2 and 6), to produce a candidate child node O. With every pair  $(l_i, l_j)$  the technique verifies if  $l_i.att \neq l_j.att$  (line 7, utilizing theorem 1. In case they are dissimilar, it calculates the elements att; values; Obidset; and total for the new node O (lines 8–10 and 17). Line 11 verifies if the amount of OIDs at position  $l_i.pos$  of node  $l_i$  is equivalent to the amount of OIDs at the similar position  $l_i.pos$  of node O (by Theorem 2). In case this is right, the technique copies the pos information from node  $l_i$  to node O (line 12). Correspondingly, in case of a false outcome on line 11, the technique performs the similar check among  $l_j$  and O (line 13) and it copies the pos information from node  $l_j$  to node O in case the check holds right (line 14). Or else, the technique calculates O.pos by computing the max value of  $j$  O.Obidset; (line 16). Subsequently calculating all information for node O, the technique utilizes Corollary 1 and theorem 3 to verify if this node could produce a rule, which fulfills minFSup (line 18) and  $Con_C$  (line 19), correspondingly. After that, it adds node O to  $P_i$  ( $P_i$  is initialized empty on line 5) in case both conditions is true (line 20). On line 20, node O is evaluated with Theorem 4 (line 20), simultaneously the ratio of the constraint class is gauged in line 20 and variable Child<sub>i</sub> is improved (line 21) in case O is a case of Theorem 4 and if theorem 5,6. On lines 22–23, in case each and every child nodes

produced from parent node  $l_i$  fulfill Theorem 4, Theorem 5,6, at that point they are detached from  $P_i$  as they and their descendant nodes could not produce rules fulfilling  $Con_C$ . Lastly, Constraint-PEFCARs -Miner is recursively called with a novel set  $P_i$  as its input parameter (line 24). The procedure Constraint-PEFCARs -Miner( $P_i, minFSup, minFConf, Pr_i$ ) produces a rule from node  $l$ . It primarily calculates the confidence of the rule (line 25); in case the confidence of this rule fulfils  $minConf$  (by Lemma 1), and the rule consequent matches up the constraints (line 27), after that this rule is added to the set of PEFCARs (line 28).

### Algorithm1: EPEFCR Miner procedure

**Input :** Dataset  $Ds$ ,  $minFSup$ ,  $minFConf$ , constraint class and  $Pr_i$

**Output:** All EPEFCR fulfilling the  $minFSup$ ,  $minFConf$ , constraint class and  $Pr_i$

Procedure:

Constraint EPEFCR \_Miner ( $L_r, minFSup, minFConf, Con_C, Pr_i$ )

1. EPEFCR =  $\emptyset$
2. For all  $l_i \in L_r$  children do
3.  $child_i = 0$
4. Enumerate EPEFCR ( $l_i, minconf, Con_C, Pr_i$ )
5.  $P_i = \emptyset$
6. for all  $l_j \in L_r$  children with  $j > i$  do
7. if  $l_i.att \neq l_j.att$  then //using theorem 1
8.  $O.att = l_i.att \cup l_j.att$  //using bitwise operation
9.  $O.values = l_i.values \cup l_j.values$
10.  $O.fuzzyvalues = l_i.fuzzyvalues \cup l_j.fuzzyvalues$
11.  $O.obiset_i = EDO(l_i.obiset_i, l_j.obiset_i)$
12. Compute  $Pr_i$
12. IF  $|O.obiset_{l_i.pos}.Pr_i| = |l_i.obiset_{l_i.pos}.Pr_i|$  //  $Pr_i$  by using the theorem 5
13.  $O.pos = l_i.Pos$
14. else if  $|O.obiset_{l_j.pos}.Pr_i| = |l_j.obiset_{l_j.pos}.Pr_i|$  //  $Pr_i$  by using the theorem 5
15.  $O.pos.Pr_i = l_j.Pos.Pr_i$
16. else  $O.pos = \arg \max_{i \in \{1, k\}} \{|O.obiset_i.Pr_i|\}$
17.  $O.total = \sum_{i=1}^k |O.obiset_i.Pr_i|$
18. if  $|O.obiset_{O.pos}.Pr_i| \geq minFSup$
19. if  $|O.obiset_i.Pr_i| \geq minFSup$  with  $\forall t \in Con_C$  then //using theorem 3
20.  $P_i = P_i \cup O$  //using theorem 4
21. if  $O.pos = l_i.pos \notin Con_C$  and  $|O.obiset_{O.pos}.Pr_i| = |l_i.obiset_{l_i.pos}.Pr_i|$  then  $child_{i++}$
22. if  $child_i = |P_i|$  then //using theorem 4
24.  $P_i = \emptyset$ ;
25. Constraint-PEFCARs -Miner( $P_i, minFSup, minFConf, Con_C, Pr_i$ );
- ENUMERATE-PEFCARs ( $l, minFConf, Con_C, Pr_i$ )
26.  $Fconf = \frac{|l.obiset_{l.pos}.Pr_i|}{l.total}$
27. if  $Fconf \geq minFConf$  and  $l.pos \in Con_C$  then // using Lemma 1
28. PEFCARs = PEFCARs  $\cup$   
 $\{l.itemset \rightarrow c_{pos}(|l.obiset_{l.pos}.Pr_i|, FConf)\}$
29. Selected rules from PEFCARs

### 1.6. Proportion frequency occurrence count with bat algorithm (PFOCBA)

Lastly, PFOCBA is presented for rule pruning step, which removes redundant or noisy information enclosed in the rule set and chooses a subset of higher quality PEFCARs. The average frequently occurred matching degree of rule  $r$  with data in class  $k$ , i.e.,  $fm_k(r)$  is computed in this manner and in case it is less than a threshold value, then rule  $r$  must be optimized

$$fm_k(r) = \frac{1}{|D_k|} \sum_{d \in D_k} Match_k(d, r) * Pr_i \quad (15)$$

$$Match_k(d, r) = \frac{N_k(d, r)}{N_k(r)} * S_k(r) * Pr_i \quad (16)$$

here,  $N_k(r)$  is called the amount of attributes in the antecedent part of rule  $r$  in class  $k$ ,  $N_k(d, r)$  is known as the amount of matched attributes with data  $d$  in the antecedent part of rule  $r$  in class  $k$ ,  $S_k(r)$  is called the strength of rule  $r$  in class  $k$  and  $D_k$  is known as set of suffixes of data in class  $k$ . Furthermore, with the intention of enhancing rules in the PEFCARs, regard the accurateness of a prediction. Assume that each and every instance in a database is autonomous of each other. Statistical theory aids the subsequent assertion [34]:

$$acc_t(R) = acc_{tr} \pm fm_k(r) \sqrt{\frac{acc_{tr}(1 - acc_{tr})}{N}} \quad (17)$$

here  $acc_t(R)$  is the true (predictive) accurateness,  $acc_{tr}$  is the accurateness over training samples,  $N$  is known as the number of training data.

Subsequent to the matching degree is identified after that identifying the lot of frequently occurring class values are identified, nevertheless in case the amount of rules is huge after that identifying lot of frequently occurred rules in the ruleset turns out to be hard. With the aim of resolving this issue and decreasing the searching time of frequently occurred rules in the  $R$ , Bat algorithm (BA) [9-10]. Bat begins from frequently occurred rules in dataset. Amount of rules formed in the ruleset is treated as the bats. At primarily it begins from the bat stochastically chooses an optimal rule within its frequency count and collaborates with this rule by taking probability value into account.

The position of the  $i^{\text{th}}$  rule of PEFCARs ruleset be represented by PEFCARs =  $(pefcar_1, \dots, pefcar_j)$ . The rule of PEFCARs rulesets are checked by  $(acc_t(R), fm_k(r), Fconf, Pr_i)$ , correspondingly. The follow-up bat phase analysis processes are in this manner. At first, every bat is encoded with a velocity  $v_i^t$  and a rule position as  $ep_{Fcar_i}^t$ , improved proportion of fuzzy class association rules at iteration  $t$ , in the rule pruning space. The rules position could be taken into account as a solution vector to a problem of interest. Amongst the 'n' number of rules (bats) in the ruleset, the present best rule position  $ep_{Fcar_*}$  identified up to now could be archived throughout the iterative search process. Initially, in case the neighbouring bat  $n_r$  had not presented a rule optimization, at that time afterwards the first rule contains lower error value, at that point those rules are enhanced by utilizing rule pruning stage. This specifies that the insignificant rules there in the rule pruning. Likewise other rules are chosen dependent upon the behaviour of the bats and accordingly, the mathematical equations for bringing up to date the rule position  $x_i^t$  and velocities  $v_i^t$  could be expressed as

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (18)$$

$$v_i^t = v_i^{t-1} + (ep_{Fcar_i}^{t-1} - ep_{Fcar_*})f_i \quad (19)$$

$$ep_{Fcar_i}^t = ep_{Fcar_i}^{t-1} + v_i^t \quad (20)$$

here  $\beta \in [0, 1]$  is known as a random vector derived from a uniform distribution. Moreover, the loudness  $A_i$  and pulse emission rates  $r_i$  could be changed throughout the iterations with fixed frequency  $f_{min} = 20 \text{ kHz}$ ,  $f_{max} = 500 \text{ kHz}$ . For easiness, make use of the subsequent equations for changing the loudness  $A_i$  and pulse emission rates  $r \in [0, 1]$ :

$$A_i^{t+1} = \alpha A_i^t \quad (21)$$

and

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (22)$$

here  $0 < \alpha < 1$  and  $\gamma > 0$  are constants. The optimal selection of every bat is chosen dependent upon the fitness function (accurateness of the classifier). The BA utilized three generalized rules for rule optimization in ruleset is defined in this manner [35-36]:

1. Each and every bat utilize an echolocation to intellect the greater accurateness that discovers the optimal rules in the EPEFCARs, they likewise estimate the dissimilarity among the source and destination node with background barriers in a fairly magic manner.
2. While looking for their optimal rule in the EPEFCARs, the bats fly arbitrarily with velocity  $v_i$  at rule position  $x_i$  with fixed frequency  $f_{min}$ , changing wavelength  $\lambda \in (0.7 \text{ mm}, 17 \text{ mm})$  and loudness  $A_0$ . They could mechanically regulate the frequency of their emitted pulses and regulate the rate of pulse emission  $r \in [0, 1]$ , based upon the contiguity of their target.
3. Even though the loudness could change from a positive  $A_0$  to a minimum constant value  $A_{min}$ .

#### Algorithm. 2. Bat algorithm(BA)

1. Objective function  $f(x)$ ,  $x=(x_1, \dots, x_d)^T$
2. Initialize the number of rules in the bat population  $x_i$  and  $v_i$  for  $i = 1 \dots n$
3. Define pulse frequency  $f_i$  at  $x_i$
4. Initialize pulse rates  $r_i$  and the loudness  $A_i$
5. While ( $t < T_{max}$ ) // number of iterations
  - 5.1. Produce new solutions by adjusting frequency and updating velocities with locations /solutions
  - 5.2. Generate a new solution by flying randomly
  - 5.3. if( $\text{rand}(0,1) < A_i$  and  $f(x_i) < f(x)$ )
    - 5.3.1. Accept the new solutions
    - 5.3.2. Increase  $r_i$  and reduce  $A_i$

- 5.4. end if
- 5.5. Rank the fitness value of all rules affected by the classification accuracy of the EPEFCARs ruleset and find the optimal rules for all bats
6. end while
7. Post-process outcomes and visualization

In the presented PFOCBA technique,  $V$  is a group of all attribute and value pairs, and sorted by their first  $fm_k(r)$ , references. A node of the candidate tree contains  $\{A, FV, Q, TC\}$ .  $A$  is a group of attribute and value pairs in the path from the root to the node, and is the antecedent of a probable rule. As  $A$  is distinct in a candidate tree, we utilize it as identity of the node. The probable target set  $TC$  is a set of classes, which might be consequences of  $A$ .  $Q$  a subset of probable attribute and fuzzy value pair sets, for every class (For instance  $z_j$ ) in  $Z$ , there is a set of probable attribute and value pairs that might be conjunct with  $A$  to form more accurate rules,  $Q_i \in Q$ :

### **PFOCBA Algorithm 3: Optimal Proportion Fuzzy Class Association Rules(PFCARs) miner**

**Input:** database  $D$  with class attribute  $C$ , the  $minFSup$ ,  $minFConf$ ,  $Probability$ .

**Output:** Optimal PFCARs ruleset  $FR$ .

```

Set optimal PFCARs  $FR = Y$ 
Count  $F_{Sup}$  of frequent patterns
Initialize candidate EFCR-tree  $T$ 
ENUMERATE-PFCARs ( $l$ ,  $minFConf$ ,  $Conc$ )
Select strong PFCARs rules from  $T$  with optimized  $FR$  and include them in  $FR$ 
Generate new candidates as leaves of  $T$ 
While (new candidate set is non-empty)
  Count  $F_{Sup}$  of the new candidates
  Prune the new candidate set
  Select strong PFCARs rules from  $T$  and include them in  $FR$ 
  Generate new candidates as leaves of  $T$ 
Return rule set  $FR$ 

```

Finally, an efficient technique for mining constrained PFCARs is presented and optimized using PFOCBA Technique .

## **4. Experimentation outcomes**

Experimentations were performed to prove the effectiveness of the proposed technique. The techniques were developed in C++ by utilizing MATLAB simulation environment on a system with an Intel Core i7-2600 3.40-GHz CPU and 4.00 GB of RAM running Windows 7 Enterprise (64-bit). The experimentations were assessed with datasets acquired from the UCI Machine Learning Repository (<http://mllearn.ics.uci.edu>). Table 5 proves the key features of the experimentation datasets called the amount of attributes , the amount of class labels, the amount of unique values, and the amount of objects in every dataset. The Breast cancer as well as Vehicle datasets contains countless attributes, distinctive values, and objects (records), where the other datasets contain a small number of attributes and objects.

Table 5. Characteristics of experimental datasets

Dataset	# of attributes	# of classes	# of distinctive values	# of objects
Lymphography	18	4	63	148
Breast	12	2	737	699
Vehicle	19	4	1434	846

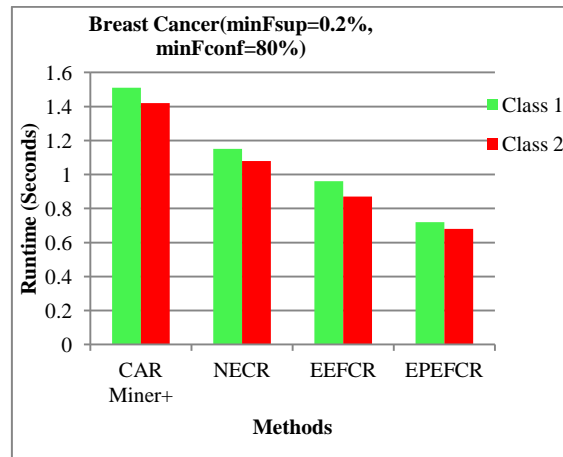


Figure 5. Runtimes comparison of tree techniques for Breast dataset

The results from Figures 5–7 prove that proposed EEFCR is higher to CAR-Miner+ and NECR in lot of experiments. For instance, take the Breast dataset with minFSup = 0.2% (Figure 6(a)). The mining times of CAR-Miner+ are 1.51 s for class 1 and 1.42s for class 2, NECR Miner wants merely 1.15s and 1.08s, correspondingly. EEFCR needs nearly 0.96s and 0.87s, correspondingly; but the proposed EPEFCR technique grosses simply 0.72 seconds and 0.68 seconds. It proves that the proposed EPEFCR technique results are 52.31% and 52.11% greater while matched up with CAR-Miner+ correspondingly.

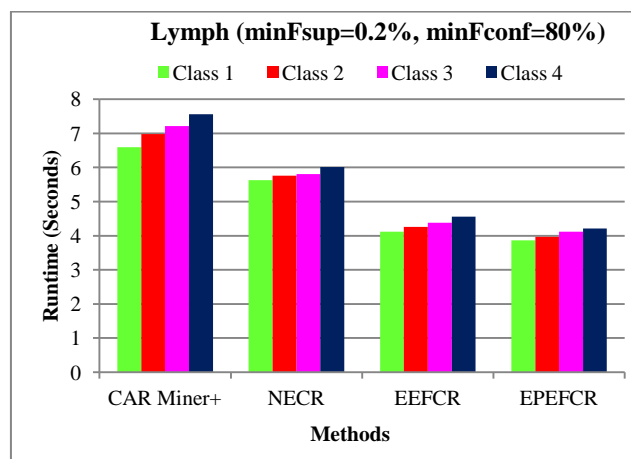


Figure 6. Runtimes comparison of tree techniques for Lymphdataset

Likewise, take the Lymph dataset with minFSup = 0.2% (Figure 6). The mining times of CAR-Miner+ are 6.59s, 6.98s, 7.21s, and 7.56s for classes 1, 2, 3, and 4, correspondingly. It takes 4.12s, 4.26s, 4.38s, and 4.56s for EEFCR correspondingly. The mining times of proposed EPEFCR technique are 3.86s, 3.97s, 4.12s and 4.21s for classes correspondingly.



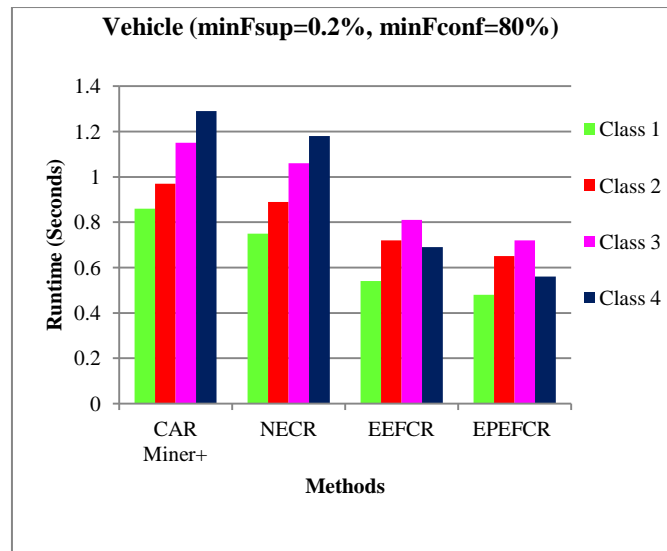
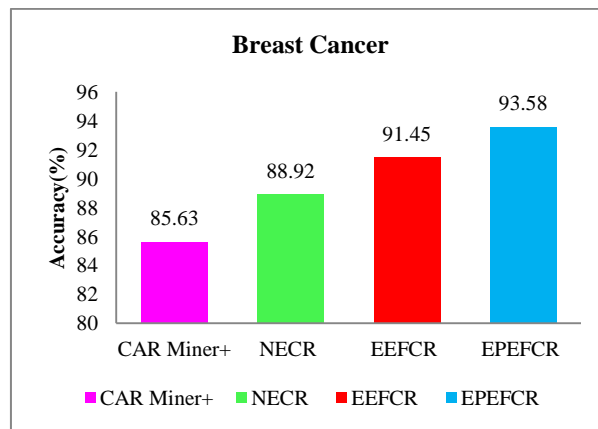
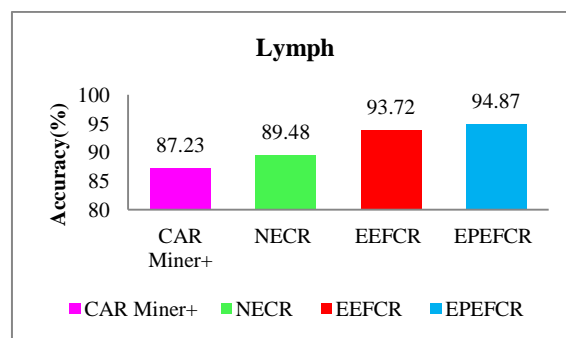


Figure 7. Runtimes comparison of tree techniques for Vehicle dataset

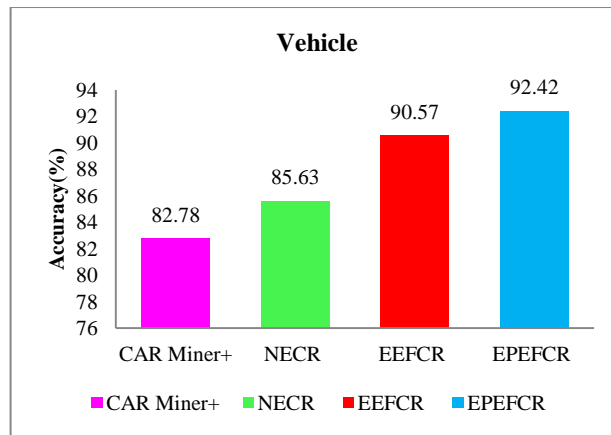
Likewise, take the vehicle dataset with minSup = 0.2% (Figure 7). The mining times of CAR-Miner+ are 0.86s, 0.97s, 1.15s, and 1.29s for classes 1, 2, 3, and 4, correspondingly, compared to 0.48s, 0.65s, 0.72s, and 0.56s of proposed EPEFCR correspondingly. Though the execution time of proposed EPEFCR technique is smaller while matched up with CAR-Miner+ as the proposed work proportion of class value is measured throughout classification task.



(a) Accuracy comparison of breast cancer (minSup = 0.2%, minFConf = 80%)



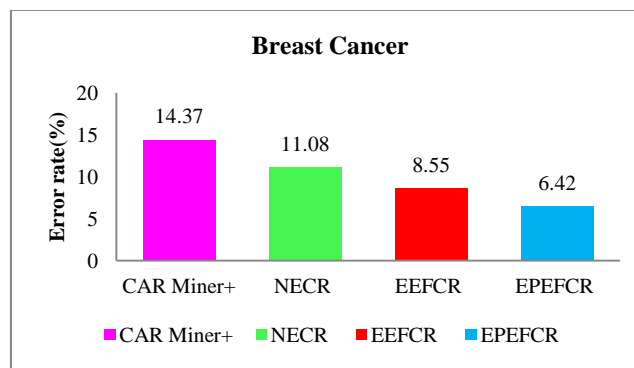
(b) Accuracy comparison of Lymph dataset (minSup = 0.2%, minFConf = 80%)



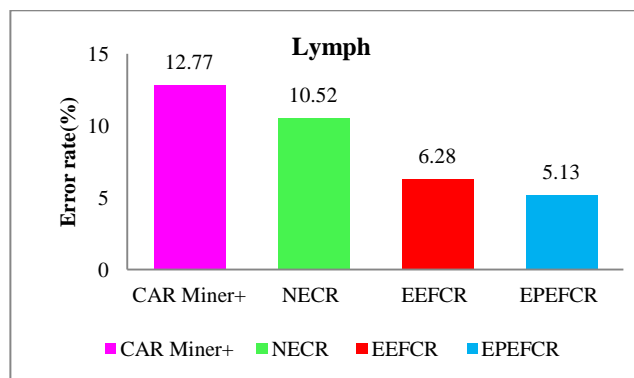
(c) Accuracy comparison of Vehicle dataset (minSup = 0.2%, minFConf = 80%)

Figure 8. Accuracy comparison vs. tree techniques at different datasets

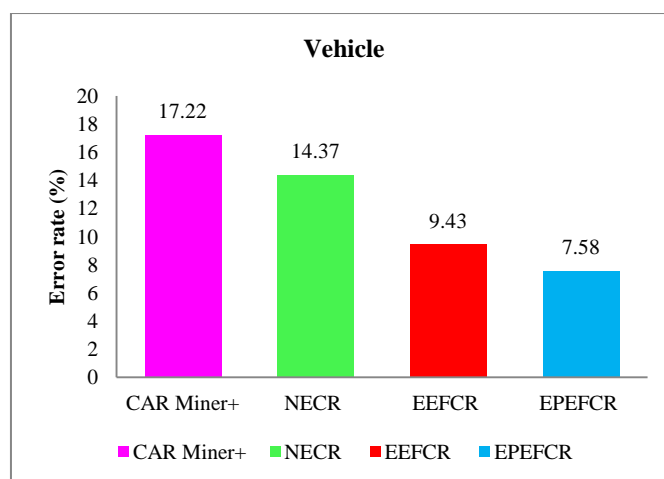
Figure 8 prove the accurateness results of the diverse tree techniques. From the results it proves that the proposed EPEFCR technique performs better to CAR-Miner+, NECR and EEFCR technique in lot of dataset experimentations. For instance, take the Breast dataset with minSup = 0.2% (Figure 8(a)). The accurateness results of the proposed EPEFCR technique is 93.58 % that is 2.27%, 4.97% and 8.49% greater while matched up with EEFCR, NECR and CAR miner+ techniques correspondingly. For instance, take the Lymph dataset with minSup = 0.2% (Figure 8(b)). The accurateness results of the proposed EPEFCR technique is 94.87 % that is 1.21%, 5.68% and 8.05% greater while matched up with EEFCR, NECR and CAR miner+ techniques correspondingly. For instance, take the vehicle dataset with minSup = 0.2% (Figure 8(c)). The accurateness results of the proposed EPEFCR technique is 92.42 % that is 2.00 % , 7.346% and 10.430% greater when compared to EEFCR, NECR and CAR miner+ techniques correspondingly.



(a) Error rate comparison of breast cancer (minSup = 0.2%, minFConf = 80%)



(b) Error rate comparison of Lymph dataset (minSup = 0.2%, minFConf = 80%)



(c) Error rate comparison of Vehicle dataset (minSup = 0.2%, minFConf = 80%)

Figure 9. Error rate comparison vs. tree techniques at different datasets

For instance, take the Breast dataset with minSup = 0.2% (Figure 9(a)). The error rate results of the proposed EPEFCR technique is 6.42 % which is 7.95%, 4.66% and 2.13% lesser while matched up with CAR-Miner+, NECR and EEFCR techniques correspondingly. For instance, take the Lymph dataset with minSup = 0.2% (Figure 9(b)). The error rate results of the EPEFCR technique is 7.64%, 5.39% and 1.15% smaller while matched up with CAR-Miner+, NECR and EEFCR techniques correspondingly. For instance, take the vehicle dataset with minSup = 0.2% (Figure 9(c)). The error rate outcomes of the EPEFCR technique is 7.58 % that is 9.64%, 6.79% and 1.85% compared with CAR-Miner+, NECR and EEFCR techniques correspondingly.

## 5. Conclusion and future work

This research work proposed an Enhanced Proportion Equivalence Fuzzy Class Rule tree (EPEFCR-tree) technique for mining PECARs. Initially the technique could calculate the support of itemsets rapidly by utilizing the list of Obidset. Next, it could directly find the position pos of certain nodes without considering the fuzzy support. The fuzzy confidence of a candidate rule is identified dependent upon this information. Then, rules in the ruleset are pruned by using the PFOCBA technique. In conclusion in EPEFCR-tree technique solves the issue of proportions of constraint class estimation for unlabeled testing data by utilizing Proportion of Constraint Class Estimation (PPCE) technique. The proposed EPEFCR-tree technique could be utilized to effective mines all class association rules to decrease the mining time and memory usage. In addition, it could be used to prune rules fast. Mining itemsets from incremental databases are designed in modern years. It could be seen that it is time and memory saving process and while matched up with mining from integration databases. Consequently, in future work we will study how to utilize this incremental database for mining PEFCARs.

## 6. References

1. H. Duong, T. Tin, B. Vo (2014). An efficient method for mining frequent itemsets with double constraints. *Engineering Applications of Artificial Intelligence*, 27, pp. 148-154.
2. B. Vo, T.P. Hong, B. Le (2013). A lattice-based approach for mining lot of generalization association rules. *Knowledge-Based Systems*, 45, pp. 20–30.
3. N. Abdelhamid, A. Ayesh, F. Thabtah, S. Ahmadi, W. Hadi (2012). MAC: A multiclass associative classification technique. *Journal of Information & Knowledge Management*, 11(2), pp. 1-10.
4. D. Nguyen, B. Vo, B. Le (2014). Efficient strategies for parallel mining class association rules. *Expert Systems with Applications*, 41(10), pp. 4716-4729.

5. R. Agrawal, R. Srikant (1994). Fast techniques for mining association rules. In: Proc. Of VLDB'94, San Francisco, CA, USA, pp. 487–499.
6. P.T. La, B. Le, B. Vo (2014). Incrementally building frequent closed itemset lattice. *Expert Systems with Applications* 41(6), pp. 2703-2712
7. B. Vo, T. Le, T.P. Hong, B. Le (2014). An effective approach for maintenance of pre-large-based frequent-itemset lattice in incremental mining. *Applied Intelligence*.
8. Yin, X., & Han, J. (2003). CPAR: Classification based on predictive association rules. In *SIAM international conference on data mining (SDM'03)* , pp. 331–335.
9. Thabtah, F., Cowling, P., & Peng, Y. (2004). MMAC: A new multi-class, multi-label associative classification approach. In *4<sup>th</sup> IEEE international conference on data mining*, pp. 217–224.
10. Thabtah, F., Cowling, P., & Peng, Y. (2005). MCAR: Multi-class classification based on association rule. In *3<sup>rd</sup> IEEE international conference on computer systems and applications*, pp. 33–39.
11. Thonangi, R., & Pudi, V. (2005). ACME: An associative classifier based on maximum entropy principle. In *16<sup>th</sup> International conference technique learning theory*, pp. 122–134.
12. Vo, B., & Le, B. (2008). A novel classification technique based on association rule mining. In *Pacific Rim knowledge acquisition workshop (held with PRICAI'08)*, pp. 61–75.
13. Veloso, A., Meira, Jr., W., & Zaki, M. J. (2006). Lazy associative classification. In *IEEE international conference on data mining (ICDM'06)*, pp. 645–654.
14. Nguyen, L.T. and Nguyen, T.N., 2010. An Efficient Pruning Approach for Class Association Rule Mining. In *Advances in Intelligent Decision Technologies*, pp. 559-570.
15. Azmi, M. and Berrado, A., Class-Association Rules Pruning Using Regularization., *IEEE 13<sup>th</sup> International Conference of Computer Systems and Applications (AICCSA)*, 2016 .
16. Gonzales, E., Taboada, K., Shimada, K. and Hirasawa, K., 2010. Efficient pruning of class association rules using statistics and genetic relation technique. *SICE Journal of Control, Measurement, and System Integration*, 3(5), pp.336-345.
17. Mohamed Azmi; AbdelazizBerrado, “Class-association rules pruning using regularization”, *IEEE International conference of computer systems and applications*, 2016.
18. Subbulakshmi, B. and Monisha, M., 2016, Incremental constraint class association rule mining of student performance dataset. *International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 1-5.
19. Nguyen, L.T. and Nguyen, N.T., 2015. Updating mined class association rules for record insertion. *Applied Intelligence*, 42(4), pp.707-721.
20. Nguyen, L., Vo, B. and Hong, T.P., 2015. CARIM: An Efficient Technique for Mining Class-Association Rules with Interestingness Measures. *International Arab Journal of Information Technology (IAJIT)*, 12.
21. Shimada, K., Hirasawa, K. and Hu, J., 2006, Association rule mining with chi-squared test using alternate genetic network programming. In *Industrial Conference on Data Mining* , pp. 202-216..
22. Sarno, R. and Sinaga, F.P., 2015, Business process anomaly detection using ontology-based process modelling and multi-level class association rule learning. *International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 12-17.
23. Mabu, S., Chen, C., Lu, N., Shimada, K. and Hirasawa, K., 2011. An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(1), pp.130-139.
24. Kharche, D. and Patil, R. “Use of Genetic Technique with Fuzzy Class Association Rule Mining for Intrusion Detection”, *International Journal of Computer Science and Information Technologies*, Vol. 5, no. 6 , pp. , 7082-7087, 2014.
25. Jia, R. and Yibo, Z., 2007, Fuzzy System Based on Class Association Rules. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 2007( FSKD 2007)*, vol. 2, pp. 155-159.

26. Abu Mansour, H.Y., 2012. Rule pruning and prediction methods for associative classification approach in data mining (Doctoral dissertation, University of Huddersfield).
27. Nguyen, D., Nguyen, L.T., Vo, B. and Hong, T.P., A novel method for constrained class association rule mining. *Information Sciences*, vol.320, pp.107-125, 2015.
28. Yaghoobi, M.A. and Tamiz, M., A note on article “A tolerance approach to the fuzzy goal programming problems with unbalanced triangular membership function”. *European Journal of Operational Research*, vol.176, no.1, pp.636-640, 2007.
29. Berenji, H.R., Fuzzy logic controllers. In *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, pp. 69-96, 1992.
30. Chi, Z., Yan, H. and Pham, T., Fuzzy techniques: with applications to image processing and pattern recognition, World Scientific, vol. 10, pp.1-225, 1996.
31. Deza, M.M. and Deza, E., Encyclopedia of distances. In *Encyclopedia of Distances*, pp. 1-583, 2009.
32. G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11, pp. 2973–3009, 2010.
33. Scott C., G. Blanchard, and G. Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Proc. 2013 Conference on Learning Theory, JMLR W&CP 30*, pp. 489–511, 2013.
34. Baştanlar, Y. and Özuysal, M., Introduction to machine learning. *miRNomics: MicroRNA Biology and Computational Analysis*, pp.105-128, 2014.
35. Yang, X.S. and Hossein Gandomi, A., 2012. Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5), pp.464-483.
36. Yang, X.S. and He, X., 2013. Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*, 5(3), pp.141-149.