# DISCOVERING FUZZY CENSORED CLASSIFICATION RULES (FCCRS): A GENETIC ALGORITHM APPROACH

Renu Bala[1] and Dr. Saroj[2]

[1]Ph.D Student, Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar, India
renu0805@gmail.com

[2]Associate Professor, Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar, India
ratnoo.saroj@gmail.com

## ABSTRACT

*Classification Rules (CRs) are often discovered in the form of 'If-Then' Production Rules (PRs). PRs, being high level symbolic rules, are comprehensible and easy to implement. However, they are not capable of dealing with cognitive uncertainties like vagueness and ambiguity imperative to real word decision making situations. Fuzzy Classification Rules (FCRs) based on fuzzy logic provide a framework for a flexible human like reasoning involving linguistic variables. Moreover, a classification system consisting of simple 'If-Then' rules is not competent in handling exceptional circumstances. In this paper, we propose a Genetic Algorithm approach to discover Fuzzy Censored Classification Rules (FCCRs). A FCCR is a Fuzzy Classification Rule (FCRs) augmented with censors. Here, censors are exceptional conditions in which the behaviour of a rule gets modified. The proposed algorithm works in two phases. In the first phase, the Genetic Algorithm discovers Fuzzy Classification Rules. Subsequently, these Fuzzy Classification Rules are mutated to produce FCCRs in the second phase. The appropriate encoding scheme, fitness function and genetic operators are designed for the discovery of FCCRs. The proposed approach for discovering FCCRs is then illustrated on a synthetic dataset.*

## KEYWORDS

*Data Mining, Fuzzy Classification Rules (FCRs), Fuzzy Censored Classification Rules (FCCRs), Genetic Algorithm.*

## 1. INTRODUCTION

The term Knowledge Discovery in Databases (KDD) signifies the complete process for discovering useful and actionable knowledge from huge amount of data existing in databases for decision making [1]. Data mining, a particular step within KDD process, automates the process of mining knowledge from large databases. Classification is a well known task of data mining where the aim is to build a model to predict the class of unseen data instance based on predictor attributes. The automated learning of classification model is accountable for discovering a set of Classification Rules (CRs). Such rules can subsequently be used both to infer properties of the class labels and to classify previously unclassified events from the data space. The generally agreed upon criteria among data mining community for evaluating a classifier are accuracy, comprehensibility and interestingness. Additionally, there has been a renewed focus on learning those classification models that can handle approximate interpretations, reason in an environment

DOI : 10.5121/ijaia.2012.3413

of uncertainty, accommodate exceptions and exhibit variable precision logic with respect to certainty and specificity. Many of these criteria depend on the underlying knowledge/rule structure employed while learning classification models. The Classification Rules (CRs) are most often represented as standard Production Rules (PRs) in the form of '*If* <*Premise*> *Then* <*Decision*>'. PRs, being high level symbolic rules, are considered comprehensible, easy to implement and several algorithms for reasoning using PRs already exist. However, there are also a number of limitations of PRs:

- Discovery of CRs in the form of PRs severely fragments the knowledge thereby resulting in a large number of rules.
- Although humans and computers both often have to reason using insufficient and incomplete or tentative premises, yet a rule base in the form of PRs is unable to handle the imperative uncertainties and vagueness that exists in decision making situations.
- An inference system is said to demonstrate variable precision logic if it is able to respond to our queries with a varying degree of certainty and specificity subject to time and other resource constraints [2]. The term 'certainty' refers to the degree of belief/confidence in a statement while 'specificity' refers to the degree of detail of a description. PRs as underlying rule structure for discovering classification rules do not exhibit variable precision logic.
- PRs simply ignore the exceptions as a noise and, hence, are unable to cope with exceptional/unusual conditions.

To deal with some of these aforesaid shortcomings of PRs, advanced underlying rule structure have been recommended in rule mining literature. Censored Production Rules (CPRs), proposed by Michalski and Winston [2], in the form of '*If* <*Premise*> *Then* <*Decision*> *Unless* <*Censors/Exceptions*>' as the underlying rule structure for the discovery of classification models not only accommodates exceptions but also exhibits variable precision logic with respect to certainty. A rule that has worked well in past might cease to work in some rare circumstances. In such situations, it is advisable to augment the rules with the exceptional conditions. Moreover, discovering a rule along with its exceptions is considered interesting. A genetic algorithm for discovery of CPRs from databases has been suggested in [3][4].

The limitation of a simple Classification Rule based systems (composed with conventional PRs and/or CPRs) is that they entail sharp cut-offs for conditional clauses. Such classification rule bases are unable to deal with uncertainty and vagueness vital to real world decision making. Fuzzy logic is a precise logic to deal with imprecision and approximate reasoning. More specifically, fuzzy logic has the capability to reason in an environment of imprecision, uncertainty, incompleteness of information, conflicting information and partial truth [5]. Fuzzy Classification Rules (FCRs) are produced by integrating fuzzy logic into CRs. FCRs are also in the form of '*If-Then*' rules, with the difference that their antecedents and consequents are composed of fuzzy constructs as an alternative to classical ones. As FCRs are considered more comprehensible and appropriate to a range of decision making situations, there have been several attempts to discover FCRs from real valued datasets [6][7][8][9][10].

The application of Evolutionary Algorithms in the domain of rule mining has shown promising results. As EAs cope better with large search spaces and problem of attribute interactions, over recent times, there has been an increasing interest in evolutionary approaches for discovering fuzzy rule bases from databases pertaining to various domains which mainly includes control and data mining applications. Genetic Algorithms (GAs), a class of Evolutionary Algorithms (EAs), are based on Darwinian natural selection and Mendelian genetics. In genetic algorithms, each point in the search space is represented by a string of alphabets called chromosome. Each chromosome is one of the possible solutions of the problem under consideration. The algorithm

performs optimization by manipulating a finite population of chromosomes. In each generation, a GA creates a population of new chromosomes by applying genetic operators like reproduction, crossover and mutation, correlating in parallel to the processes in natural reproduction [11].
This paper proposes a Genetic Algorithm approach to discover Fuzzy Censored Classification Rules (FCCRs) from datasets. The rule structure is basically an integration of fuzzy logic with CPRs. A classifier discovered in the form of FCCR shall provide an excellent mechanism for exception handling. In addition, it shall have flexibility of fuzzy logic with the further advantage of reasoning with variable certainty. The rest of the paper is organized as follows. Section 2 gives the related work for necessary contextualization of our work. The proposed system for discovery of FCCRs is described in Section 3. An illustration for discovery of FCCRs from an example dataset is given in Section 4. Concluding remarks and the future direction are presented in Section 5.

## 2. BACKGROUND AND RELATED WORK

### 2.1. Fuzzy Classification Rules (FCRs)

A classification problem involves learning a set of CRs from a training dataset and validating these rules on a test dataset. These CRs are extracted in the form of PRs ('*If* <***Premise***> *Then* <***Decision***>' or ***P→D***). A CR is represented as:

$$If\ (x_1\ op\ v_{1j}) \wedge (x_2\ op\ v_{2j}) \wedge ... \wedge (x_n\ op\ v_{nj})\ Then\ D_m$$

Assuming a dataset with *n* real valued attributes and *q* categorical class labels, $v_{ij}$ represents the $j^{th}$ value of $i^{th}$ attribute, $D_m$ is one of the class labels where $m \in 1\ ..\ q$. The values of *j* may vary depending on the number of values in the domain of attribute $x_i$ and op may take values =, <, <=, > or >=. An example of such a rule is-

$$If\ (Marks\ \geq 75)\ and\ (IQ > 80)\ Then\ Student\_status = Brilliant$$

This rule states that a student with marks more than or equal to 75 and an IQ greater than 80 is considered brilliant. However, according to this rule a student with marks 74.9 and IQ=79.9 is not classified as brilliant. Such sharp cut-offs are unreasonable and not suitable when modelling human intuitive reasoning. As fuzzy logic has the ability to use linguistic interpretations in a mathematical framework, it provides a natural way for constructing fuzzy rule based classification systems that are closer to the human decision making process [5]. A FCR takes the following form:

$$If\ (x_1 is\ A_{1k}) \wedge (x_2\ is\ A_{2k}) \wedge ... \wedge (x_n\ is\ A_{nk})\ Then\ D_m$$

In the above rule, the antecedent or the consequent, or both, may be fuzzy rather than crisp. $A_{1k}$, $A_{2k}$ … $A_{nk}$ are antecedent fuzzy sets on unit interval [0, 1] where *k* varies over number of fuzzy partitions or fuzzy linguistic labels (like *small*, *medium* and *large*) chosen for the attribute $x_i$. The value of *k* may be same for all the attributes or different for different attributes. Each $A_{ik}$ may assume one of the linguistic labels. The earlier example of the CR takes the following form as FCR which is a better and natural representation of the way human being comprehend and reason:

$$If\ (Marks = Excellent)\ and\ (IQ = High)\ Then\ Student\_status = Brilliant$$

Most of the approaches proposed in the literature for automated discovery of Fuzzy Classification Rules use predefined fuzzy membership functions or fuzzy partitions and there are a number of contributions where antecedent and consequent parts of simple PRs become fuzzy constructs [12][13][14]. In this context, a pioneering work has also been carried out by Thrift for learning of fuzzy rule bases [15] and Ishibuchi (1999) has proposed a hybrid genetics-based machine learning algorithm for designing a linguistic classification system that consists of small number of fuzzy

‘*If-Then’* rules [16]. Several other prominent and classical proposals by Ishibuchi et al. for discovering FCRs can be found in [6][17][18]. FCRs have demonstrated their ability in a wide spectrum of applications in the domain of control problems [19], modelling [20] and data mining [21][22].

## 2.2. Censored Production Rules

A Censored Production Rule (CPR), proposed by Michalski and Winston, is in fact a PR augmented with censor/exception conditions [2]. A CPR can be written in the form:

$$If\ (x_1\ op\ v_{1j}) \wedge (x_2\ op\ v_{2j}) \wedge ... \wedge (x_n\ op\ v_{nj})\ Then\ D_m\ \ Unless\ (x_3\ op\ v_{3j}) \vee (x_4\ op\ v_{4j}) : \gamma_1 : \gamma_2$$

$$P = (x_1\ op\ v_{1j}) \wedge (x_2\ op\ v_{2j}) \wedge ... \wedge (x_n\ op\ v_{nj})$$

$$C = (\mathrm{x}_3\ op\ \mathrm{v}_{3\mathrm{j}}) \vee (\mathrm{x}_4\ op\ \mathrm{v}_{4\mathrm{j}})$$

In this rule *P* is the premise part and is as usual a conjunction of attribute-value pairs. *C* may contain a single exception/censor or it may be a disjunction of exceptions/censors. It is important to note here that the attributes present in the premise and censor part are mutually exclusive. The ‘*If P Then D*’ part of a CPR holds frequently and the censor part *C* holds rarely. Therefore, the ‘*Unless’* operator acts like an XOR operator between decision and censor but it has an expositive aspect which is not commutative. To capture this asymmetry precisely, two parameters associated with this kind of rule structure are defined below:

$$\gamma_1 = \Pr\ ob.[D|P] = \frac{\Omega_{PD}}{\Omega_P} = \frac{|P \wedge D|}{|P|} \quad (1)$$

$$\gamma_2 = \Pr\ ob.[C\ |\ P\ ] = \frac{\Omega_{PC}}{\Omega_P} = \frac{|\ P \wedge C\ |}{|\ P\ |} \quad (2)$$

In the above equations, $\Omega_{PD}$ is subset of events (objects in the database) for which both *P* and *D* hold, $\Omega_{\mathrm{PC}}$ is subset of events for which both *P* and *C* hold, and $\Omega_{\mathrm{P}}$ is the subset of the events for which only *P* holds. The main constraints on CPR are that $\Omega_{PD} \cap \Omega_{PC} = \varphi$ , $\Omega_{PD} \cup \Omega_{PC} = \Omega_{P,}$ $\gamma_1 + \gamma_2 <= 1$ and $\gamma_1 >> \gamma_2$. We are free to ignore a censor/exception condition if the resources are scarce to establish its presence or simply no information is available as to whether the censor holds true or not. As time permits, the censor condition *C* is evaluated establishing the conclusion *D* with higher certainty if *C* does not hold, or simply changing the polarity of *D* to ~*D* if *C* holds. Let us take the following example of a CPR about birds:

> *If* (X = Bird) *Then* (Decision = Fly [$\gamma_1$= 0.9]) *Unless* (X=kiwi [γ2=0.05]) $\vee$(X=sick bird [γ2=0.006])

Now, consider a query “Does X fly?” The quickest possible answer is that if X= Bird then it flies with a certainty level of 90 percent. Given more time, exceptions can be evaluated. If this bird is a kiwi or it is sick then the decision is negated to ~Fly. If none of the exceptions hold then we can say with more certainty that this particular bird flies. The certainty of the decision increases by the factor γ2 in case an exception turns out to be false. For instance if X is not a kiwi then X flies with a certainty factor of 0.95. Further, if the bird is not sick, it flies with certainty factor of 0.956. That is how a CPR supports variable precision logic with respect to certainty of a decision and hence approximate reasoning under resource constraint. It is obvious that *If* (X=kiwi) *Then* ~Fly *Unless* (X=Bird) is not a CPR as it violates the constraints on $\gamma_1$ and $\gamma_2$. A CPR exercises control over certainty and the specificity of the decision remains constant. A classification algorithm based on evolutionary approach for discovering comprehensible rules with exceptions in the form of CPRs, is presented in [3]. Further, a parallel GA approach (island model) for automated discovery of CPRs has also been proposed in [4].

## 3. THE PROPOSED SYSTEM FOR DISCOVERY OF FCCRS

This section describes the genetic algorithm approach for automated discovery of FCCRs. A FCCR is an integration of FCR and CPR, and employs fuzzy constructs in its premise as well as censor part. A short notation is used in this paper to represent FCCR - $P \rightarrow D \oplus C$; where $\oplus$ stands for the '*Unless*' operator. A FCCR is specified as below:

$$(x_1 \text{ is } A_{1k}) \wedge (x_2 \text{ is } A_{2k}) \wedge \ldots \wedge (x_n \text{ is } A_{nk}) \rightarrow D_m \oplus (x_3 \text{ is } A_{3k}) \vee (x_4 \text{ is } A_{4k})$$

An example FCCR involving fuzzy terms is shown below.

$$\begin{aligned}&(years\_employed = medium \vee years\_employed = high) \wedge (income = high)\\ &\rightarrow Then\ (Credit = Approved)\\ &\oplus (Emp\_defaulter\_status = high \vee Emp\_health\_status = poor)\end{aligned}$$

The algorithm to discover FCCRs works in two phases. In the first phase real valued attributes are normalized and fuzzified. Then a GA is employed to extract FCRs in the form $P \rightarrow D$. In this work, the antecedent of the discovered rules is in CNF form allowing disjunction between the values of same attributes. These FCRs discovered in the first phase are taken as initial population for the second phase. FCCRs are discovered by adding clauses as censors in '*Unless*' part and mutating the class $D_m$ to some other class. The detail steps of the two phases are given below.

### Phase 1: Discovery of FCRs: A Genetic Algorithm Approach

1. **Normalization:** The dataset is normalized so that all the data values fall between 0 and 1. An attribute is normalized with the formula given below.

$$A' = \frac{A - \min(A)}{\max(A) - \min(A)} \qquad (3)$$

2. **Fuzzification**: The dataset is fuzzified in the pre-processing phase. The fuzzification is done by assigning a membership degree to the numeric values of an attribute into three fuzzy linguistic values (*small*, *medium*, and *large*). The membership degree of linguistic values can be decided either subjectively or by a membership function defined on the range of numeric values of the attribute. Although there are several fuzzy membership functions, we have used the triangular fuzzy membership function for the sake of simplicity and it is shown in Fig1.

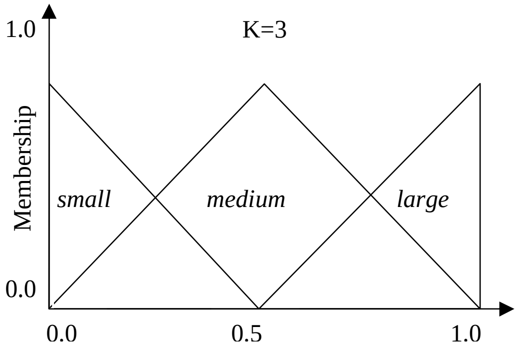

Figure 1. Triangular Membership Function

The formula used to fuzzify values of an attribute into three fuzzy partitions is as follows:

$$\mu_k^i = \max\{(1 - \frac{|X - a_k^i|}{b^k}), 0\} \qquad (4)$$

$$a_k^i = \frac{i-1}{k-1} \quad i = 1, 2, ..., k; k = 3 \text{ and } b^k = \frac{1}{k-1}$$

The value $k$ denotes the number of linguistic variables. An illustration to determine the membership degree of the value 0.12 into three fuzzy linguistic variables (*small*, *medium*, *large*) is given below:

$$\mu_3^{small} = \max\{(1 - \frac{|0.12 - 0|}{0.5}), 0\} = 0.76$$

$$\mu_3^{medium} = \max\{(1 - \frac{|0.12 - 0.5|}{0.5}), 0\} = 0.24$$

$$\mu_3^{large} = \max\{(1 - \frac{|0.12 - 1|}{0.5}), 0\} = 0$$

3. **Initializing Population**: This step generates an initial population of '*If-Then'* rules to be processed by GA. We have adopted the Michigan approach where each chromosome in the population represents a single rule. A chromosome (an individual) is divided into two parts: If part consists of the premise and a consequent part represents the decision class. We have used a pure binary string for encoding a chromosome. A block of three bits is used for each attribute. The three bits in a block correspond to three consecutive linguistic fuzzy variables- '*small'*, '*medium'* and '*large'*. Within a block, a '1' bit indicates the presence of the linguistic term whereas a '0' bit marks the absence of any value. A block with all three bits set to 1 or 0 is treated as a '*do not care*' state which indicates absence of an attribute from the rule. The consequent part contains the number of bits equal to the number of class labels i.e. for a three class classification problem the consequent part will have three bits. In the consequent part only one of the bits can be set to 1 so that the chromosome is assigned to only one of the classes. The encoding scheme and its mapping to rules are shown in Figure 2. The encoding scheme maps the chromosomes to rules in CNF form where there is a conjunction between different attributes and a disjunction within the different values of the same attribute.

| Genotype (Chromosomes) | Phenotype (Rules) |
|---|---|
| A1 \| A2 \| A3 \| A4 \| A5 \| Class<br>100 \| 000 \| 111 \| 000 \| 001 \| 100 | $(A_1\text{=Small}) \wedge (A_5\text{=Large}) \longrightarrow D_1$ |
| A1 \| A2 \| A3 \| A4 \| A5 \| Class<br>100 \| 111 \| 000 \| 011 \| 111 \| 010 | $(A_1\text{=Small}) \wedge (A_4\text{=Medium} \vee A_4\text{=Large})$<br>$D_2$ |

| A1 | A2 | A3 | A4 | A5 | Class |
|---|---|---|---|---|---|
| 100 | 000 | 111 | 000 | 001 | 100 |

| A1 | A2 | A3 | A4 | A5 | Class |
|---|---|---|---|---|---|
| 100 | 111 | 000 | 011 | 111 | 010 |

Figure2. Encoding Scheme

4. **Evaluating Fitness**: Several criteria are used to quantify the quality, or fitness, of a rule. Some of these criteria are highly qualitative and in some cases subjective. However in the context of generic search we must formulate a single numerical quantity that encapsulates

the desirable features. In the proposed approach the product of confidence and coverage of a fuzzy rule is considered as a measure of fitness. To compute the confidence and coverage we need a mechanism to measure the degree of match between a fuzzy rule and an object from the dataset. The process of computing the degree of match between a fuzzy rule r and an object of database u is based on *MAX_MIN* operators of Mamdani model and same as used in [20]. The match method is described as below.

The degree of match for an individual attribute $A_i$ between a rule r and an object u is given by

$$mA_i(r,u) = \begin{cases} 1 & if\, s(A_i) = \#, \\ \max_j( \min( s(T_i^k), \mu_{T_i^k}(u)) & otherwise \end{cases} \qquad (5)$$

In the above equation, $s(A_i)$ represents the binary string for ith attribute, '#' stands for 'don't care' state, $s(T_i^k)$ symbolizes the kth bit of the ith attribute and $\mu_{T_i^k}(u)$ gives the membership degree of corresponding term in kth fuzzy set. The degree of match between the premise ***P*** of a rule r and an object u of the dataset is given by

$$mP(r,u) = \underset{i}{Min}\,(mA_i(r,u)) \qquad (6)$$

As the conclusion part of the rule is not in a fuzzified form in our implementation, the conclusion of the rule r and the class of an object u can be directly matched. To make things clearer an illustration of the above matching process for the premise part follows here. Let us assume the premise part of a rule has five attributes and a fuzzified instance of an object u as shown below in Figure. 3.

| | **A1** | | | **A2** | | | **A3** | | | **A4** | | | **A5** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Encoded FCR(r) | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| object (u) | 0.82 | 0.12 | 0 | 0.9 | 0.1 | 0 | 0.62 | 0.38 | 0 | 0 | 0.84 | 0.16 | 0.72 | 0.68 | 0 |
| $mA_i(r,u)$ | $mA_1(r,u)$ = max[min(1, 0.82), min(1, 0.12), min(0, 0)] = 0.82 | | | $mA_2(r,u)$ = 1.0 (*do not care* state) | | | $mA_3(r,u)$ = max[min(1,0.62), min(0,0.38), min(1,0)]=0.62 | | | $mA_4(r,u)$= max[min(0,0), min(1,0.84), min(0,0.16)]=0.84 | | | $mA_5(r,u)$ = max[min(0,0.72), min(1,0.28), min(1,0)]=0.68 | | |
| *mP(r,u)* = min (0.82, 1.0, 0.62, 0.84, 0.68) = 0.62 | | | | | | | | | | | | | | | |

Figure 3: Matching Process for a rule r and an object u

In case of above illustration we can conclude that the premise part of the rule r and the instance u matches with a degree of 0.62. This degree of match is compared to significance level $\alpha$. If the value of *mP (r, u)* is greater than the $\alpha$ cut value, and the conclusion part of the rule r (c) and the class label of the instance u (c) matches, we can count this as one of the true positive cases. Now that we have a mechanism for matching a rule r and an instance u belonging to the dataset, we can define confidence factor and coverage of a rule at significance level $\alpha$ as follows [23].

$$cf = \frac{\sum_{u \in U} f_\alpha(mP(r,u)) \wedge (r(c) = u(c))}{\sum_{u \in U} f_\alpha(mP(r,u))} \qquad (7)$$

$$Cov = \frac{\sum_{u \in U} f_{\alpha}(mP(r,u)) \wedge (r(c) = u(c))}{|(r(c)|} \tag{8}$$

$$Fitness\ (R_i) = cf \times \mathrm{cov} \tag{9}$$

Here $f_{\alpha}$ is the $\alpha$ cut function with a value greater than 0.5. In our experiments we have taken the value of $\alpha$ as 0.6.

5. **Applying Genetic Operators:** Genetic operators are used to create new individuals from existing ones. There are two primary genetic operators: crossover and mutation. The operators employed in this GA are described as under:

   **Crossove**r:

   Crossover operator exchanges a random number of complete blocks from two parents to make two offspring. The random creation of crossover points takes place only at the start of the blocks. For example, two parents generating two offspring through crossover at the beginning of fourth blocks are shown below.

   Parent1: 100: 000:110:**000:111:100**
   Parent2: 001:110:111:**011:000:001**
   Child1: 100:000:110: **011:000:001**
   Child2: 001:110:111: **000:111:100**

   These chromosomes can be interpreted as following rules
   Parent1: ($A_1$=small) && ($A_3$=small || $A_3$= medium) $\rightarrow D_1$
   Parent 2: ($A_1$=large) && ($A_2$=small ||$A_2$=medium) && ($A_4$=medium ||$A_4$=large) $\rightarrow D_3$
   Child1; ($A_1$=small) && ($A_3$=small || $A_3$=medium) && ($A_4$=medium ||$A_4$=large) $\rightarrow D_3$
   Child2; ($A_1$=large) && ($A_2$=small || $A_2$=medium) && $\rightarrow D_1$
   The crossover operator is capable of inserting and dropping the whole attributes as well as the terms within an attribute block.

   **Mutation:**
   Mutation operator introduces a gene value into an individual that was not even present in the population thereby increasing the genetic diversity of the population. The mutation operator generates a new chromosome by mutating one of the blocks in the chromosome. Each block has equal probability of being mutated. For example, mutation on third block may generate one of the seven new chromosomes as given below.

   Parent: (001:110: **010:** 011: 000: 001)
   Mutated: (001:110: **000:** 011: 000: 001) or
   (001:110: **001:** 011: 000: 001) or
   (001:110: **011:** 011: 000: 001) or
   (001:110: **100:** 011: 000: 001) or
   (001:110: **101:** 011: 000: 001) or
   (001:110: **110:** 011: 000: 001) or
   (001:110: **111:** 011: 000: 001)

The detailed algorithm for evolving FCRs is given in Figure 4.

### Phase 2: Discovering FCCRs from FCRs

In this phase the population of the FCRs discovered in the first phase is mutated over a number of generations to produce FCCRs. As the attributes occurring in the *premise* and *censor* part are mutually exclusive, the mutation operator in this phase is restricted only to the blocks of attributes which do not figure in the premise part of the pre-discovered FCRs. The detailed algorithm for the second phase is given in Figure 5.

```
Algorithm Evolving_FCRs
Input: T: Training Dataset
$N_{replace}$ // Crowding factor
$N_{worst}$  //Crowding sub-population size
Output: A set of FCRs
Begin
1.  Normalize the dataset
2.  Fuzzify values of each attribute into small, medium and large fuzzy terms.
3.  Generate an initial population of FCRs by randomly generating bits over the chromosome
    length
4.  While stopping criterion not met do
    4.1. Evaluate the fitness of each rule in the population based on training dataset T
    4.2. Apply proportionate selection
    4.3. Apply crossover and mutation to create new rules
    4.4. Replace most similar $N_{replace}$ ruless among a randomly selected  $N_{worst}$ rules from the
         current population
    End While
Stop.
```

Figure 4. Algorithm evolving FCRs in the first phase

```
Algorithm Discovering FCCRs
    Input: Pre-discovered FCRs
    Output: A set of FCCRs
Begin
1.  Initializing FCCRs Population: FCCRs=[]
2.  For each rule $R_i$ ε FCRs (evolved in first phase) with confidence < 1 repeat the following steps
    2.1. Create a set TP of true positive examples covered by the rule $R_i$ (If P Then D) or $P \longrightarrow D$.
    2.2. Create a set FP of false positive examples covered by If P Then ~ D or $P \longrightarrow$ . ~D.
    2.3. Repeat steps 2.3.1 to 2.3.9 over a number of generations
         2.3.1.  Insert new conditions through mutation in rule $R_i$ keeping the conditions already present
                 in the rule fixed. Mutate the class D to ~D. This shall produce the rules in the form of
                 FCPRs: $R_i$'= $p \wedge C_j \longrightarrow \sim D$. Here, C represents censor condition.
         2.3.2.  Compute $\gamma_{1,}$ $\gamma_2$ and confidence of $R_i$' from the data subset containing TP+ FP.
         2.3.3.  TP'=| $p \wedge C_j \wedge \sim D$|
         2.3.4.  FP'=| $p \wedge C_j \wedge D$|
         2.3.5.  $\gamma_1$= TP/TP+FP // for the rule $R_i$ $(P \longrightarrow D)$
         2.3.6.  $\gamma_2$=TP'/TP+FP // for $p \wedge C_j \longrightarrow \sim D$
         2.3.7.  Conf($R_i$') = TP'/TP'+FP'
         2.3.8.  If ($\gamma_1$>> $\gamma_2$ && conf($R_i$') ==1 && (($A_i$ =$V_{ij}$) $\in$ C) $\notin$ TP) // none of the censor
                                                                        // conditions can
                                                                        //appear anywhere in the
                                                                        //true positive examples
                                                                        //covered by the rule if P
                                                                        //Then/D
       2.3.9.  FCCRs[]=FCCRs[] $\cup$ $R_i$
End Repeat
Stop.
```

Figure 5. Algorithm for discovering FCCRs in the Second Phase

## 4. EXPERIMENTATION AND AN ILLUSTRATION

The proposed approach is demonstrated through a synthetic dataset. The dataset represents a three class classification problem involving five continuous attributes. The dataset containing 34 instances is shown in Table A-1 in the Appendix- A. It is to be noted that dataset is already in normalized form. Each attribute of Table 1 is fuzzified according to three linguistic labels – *small*, *medium* and *large* as described in the previous section. This fuzzified dataset is given in Table1. Although the sum of membership degrees of all the linguistic terms for an attribute is equal to 1 in this table below, it is pertinent to mention here that membership degrees of linguistic terms are not probabilities and we can very well use type of membership functions and this sum needs not at all to be 1.

Table 1. Synthetic Dataset in Fuzzified from

| **Sr. No** | **Attribute1** | | | **Attribute2** | | | **Attribute3** | | | **Attribute4** | | | **Attribute5** | | | **Class** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **S** | **M** | **L** | **S** | **M** | **L** | **S** | **M** | **L** | **S** | **M** | **L** | **S** | **M** | **L** | **D1** | **D2** | **D3** |
| 1 | 0.76 | 0.24 | 0 | 0.78 | 0.22 | 0 | 0.9 | 0.1 | 0 | 0.98 | 0.02 | 0 | 0 | 0.9 | 0.1 | 1 | 0 | 0 |
| 2 | 0.64 | 0.36 | 0 | 0.52 | 0.48 | 0 | 0.92 | 0.08 | 0 | 0.4 | 0.6 | 0 | 0 | 0.86 | 0.14 | 1 | 0 | 0 |
| 3 | 0.74 | 0.26 | 0 | 0.64 | 0.36 | 0 | 0.84 | 0.16 | 0 | 0.6 | 0.4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 4 | 0.68 | 0.32 | 0 | 0.98 | 0.02 | 0 | 0.9 | 0.1 | 0 | 0.9 | 0.1 | 0 | 0 | 0.82 | 0.18 | 1 | 0 | 0 |
| 5 | 0.64 | 0.36 | 0 | 0.9 | 0.1 | 0 | 0.7 | 0.3 | 0 | 0.92 | 0.08 | 0 | 0 | 0.92 | 0.08 | 1 | 0 | 0 |
| 6 | 0.66 | 0.34 | 0 | 0.84 | 0.16 | 0 | 0 | 0.82 | 0.18 | 0 | 0.88 | 0.12 | 0 | 0.2 | 0.8 | 1 | 0 | 0 |
| 7 | 0.62 | 0.38 | 0 | 0.8 | 0.2 | 0 | 0 | 0.8 | 0.2 | 0 | 0.8 | 0.2 | 0 | 0.4 | 0.6 | 1 | 0 | 0 |
| 8 | 0.9 | 0.1 | 0 | 0.96 | 0.04 | 0 | 0 | 0.88 | 0.12 | 0 | 0.78 | 0.22 | 0 | 0.24 | 0.76 | 1 | 0 | 0 |
| 9 | 0.88 | 0.12 | 0 | 0.86 | 0.14 | 0 | 0 | 0.92 | 0.08 | 0 | 0.92 | 0.08 | 0 | 0.26 | 0.74 | 1 | 0 | 0 |
| 10 | 0.82 | 0.18 | 0 | 0.84 | 0.16 | 0 | 0 | 0.96 | 0.04 | 0 | 0.9 | 0.1 | 0 | 0.22 | 0.78 | 1 | 0 | 0 |
| 11 | 0 | 0.92 | 0.08 | 0.76 | 0.24 | 0 | 0.9 | 0.1 | 0 | 0.8 | 0.2 | 0 | 0 | 0.92 | 0.08 | 0 | 1 | 0 |
| 12 | 0 | 1 | 0 | 0.74 | 0.26 | 0 | 0.9 | 0.1 | 0 | 0.98 | 0.02 | 0 | 0 | 0.88 | 0.12 | 0 | 1 | 0 |
| 13 | 0 | 0.86 | 0.14 | 0.72 | 0.28 | 0 | 0.84 | 0.16 | 0 | 0.96 | 0.04 | 0 | 0.02 | 0.98 | 0 | 0 | 1 | 0 |
| 14 | 0 | 0.98 | 0.02 | 0.7 | 0.3 | 0 | 0.88 | 0.12 | 0 | 0.94 | 0.06 | 0 | 0.04 | 0.96 | 0 | 0 | 1 | 0 |
| 15 | 0 | 0.96 | 0.04 | 0 | 0.96 | 0.04 | 0.9 | 0.1 | 0 | 0.92 | 0.08 | 0 | 0 | 0.82 | 0.18 | 0 | 1 | 0 |
| 16 | 0 | 0.88 | 0.12 | 0 | 0.8 | 0.2 | 0 | 0.96 | 0.04 | 0 | 0.8 | 0.2 | 0 | 0.16 | 0.84 | 0 | 1 | 0 |
| 17 | 0.04 | 0.96 | 0 | 0 | 0.84 | 0.16 | 0 | 0.92 | 0.08 | 0 | 0.78 | 0.22 | 0 | 0 | 1 | 0 | 1 | 0 |
| 18 | 0.02 | 0.98 | 0 | 0 | 0.42 | 0.58 | 0 | 0.96 | 0.04 | 0 | 0.8 | 0.2 | 0 | 0.22 | 0.78 | 0 | 1 | 0 |
| 19 | 0.08 | 0.92 | 0 | 0 | 0.4 | 0.6 | 0 | 0.8 | 0.2 | 0 | 0.96 | 0.04 | 0 | 0.26 | 0.74 | 0 | 1 | 0 |
| 20 | 0 | 0.94 | 0.06 | 0 | 0.36 | 0.64 | 0 | 0.78 | 0.22 | 0 | 0.84 | 0.16 | 0 | 0.24 | 0.76 | 0 | 1 | 0 |
| 21 | 0 | 0.3 | 0.7 | 0.6 | 0.4 | 0 | 0.98 | 0.02 | 0 | 0.8 | 0.2 | 0 | 0 | 0.92 | 0.08 | 0 | 0 | 1 |
| 22 | 0 | 0.28 | 0.72 | 0.92 | 0.08 | 0 | 0.9 | 0.1 | 0 | 0.94 | 0.06 | 0 | 0 | 0.9 | 0.1 | 0 | 0 | 1 |
| 23 | 0 | 0.2 | 0.8 | 0.8 | 0.2 | 0 | 0.64 | 0.36 | 0 | 0.84 | 0.16 | 0 | 0 | 0.82 | 0.18 | 0 | 0 | 1 |
| 24 | 0 | 0 | 1 | 0 | 0.92 | 0.08 | 0.72 | 0.28 | 0 | 0.72 | 0.28 | 0 | 0 | 0.8 | 0.2 | 0 | 0 | 1 |
| 25 | 0 | 0.32 | 0.68 | 0 | 0.88 | 0.12 | 0.82 | 0.18 | 0 | 0.64 | 0.36 | 0 | 0 | 0.86 | 0.14 | 0 | 0 | 1 |
| 26 | 0 | 0.2 | 0.8 | 0 | 0.84 | 0.16 | 0 | 0.86 | 0.14 | 0 | 0.9 | 0.1 | 0 | 0.02 | 0.98 | 0 | 0 | 1 |
| 27 | 0 | 0.32 | 0.68 | 0 | 0.96 | 0.04 | 0 | 0.92 | 0.08 | 0 | 0.92 | 0.08 | 0 | 0.04 | 0.96 | 0 | 0 | 1 |
| 28 | 0 | 0.16 | 0.84 | 0 | 0.44 | 0.56 | 0 | 0.96 | 0.04 | 0 | 0.88 | 0.12 | 0 | 0.22 | 0.78 | 0 | 0 | 1 |
| 29 | 0 | 0.12 | 0.88 | 0 | 0.32 | 0.68 | 0 | 0.8 | 0.2 | 0 | 0.86 | 0.14 | 0 | 0.3 | 0.7 | 0 | 0 | 1 |
| 30 | 0 | 0.26 | 0.74 | 0 | 0.2 | 0.8 | 0 | 0.9 | 0.1 | 0 | 0.8 | 0.2 | 0 | 0.04 | 0.96 | 0 | 0 | 1 |
| 31 | 0.8 | 0.2 | 0 | 0 | 0.88 | 0.12 | 0 | 0.2 | 0.8 | 0 | 0.84 | 0.16 | 0.92 | 0.08 | 0 | 0 | 0 | 1 |
| 32 | 0.72 | 0.28 | 0 | 0 | 0.26 | 0.74 | 0 | 0.3 | 0.7 | 0.82 | 0.18 | 0 | 0.88 | 0.12 | 0 | 0 | 0 | 1 |
| 33 | 0 | 0.84 | 0.16 | 0.76 | 0.24 | 0 | 0 | 0.2 | 0.8 | 0.76 | 0.24 | 0 | 0.7 | 0.3 | 0 | 0 | 0 | 1 |
| 34 | 0 | 0.92 | 0.08 | 0.78 | 0.22 | 0 | 0 | 0.26 | 0.74 | 0.74 | 0.26 | 0 | 0.68 | 0.32 | 0 | 0 | 0 | 1 |

We have implemented the proposed system for discovery of FCCRs in GAlib247 on Ubuntu platform version 11.6. GAlib is a C++ library of genetic algorithm objects (developed at Massachusetts Institute of Technology, USA http://web.mit.edu/) which includes tools for applying genetic algorithms using a number of encoding schemes and genetic operators. We have used a crowding GA instead of a simple genetic algorithm because our target is to get a solution consisting of a set of best rules and not the single best rule. A crowding GA maintains the various species in the GA population where the offspring replace the most similar individuals among a number of worst individuals selected from an overlapping population. A binary encoded chromosome of length 18, single point crossover and flip mutation as described in previous section is used. The parameters of the genetic algorithm were tuned in the few initial runs of the genetic algorithm. The final results are obtained with a population size of 50, maximum number of generation equals 300, crossover probability equals to 0.6, a mutation rate of 0.1, crowding sub-population size of 4 and a crowding factor equals to 3. In the first phase our GA discovers three FCRs as given in Table 2.

Table 2. FCRs discovered in first phase

| **Sr. No.** | **FCRs** | **$\gamma_1$ (confidence)** | **Coverage** | **Fitness** |
|---|---|---|---|---|
| 1 | $(A_1$= Small$) \rightarrow D_1$ | 0.8333 | 1 | 0.8333 |
| 2 | $(A_1$= Medium$) \rightarrow D_2$ | 0.8333 | 1 | 0.833 |
| 3 | $(A_1$=Large$) \rightarrow D_3$ | 1 | 0.7143 | 0.7143 |

In the table above, there are two rules that have confidence factors less than 1. These are the fuzzy rules that can possibly have exceptions or censors.

Assuming 0.8 and 0.2 as the predefined thresholds for $\gamma_1$ and $\gamma_2$ respectively, let us consider the first rule for the second phase. For this rule the set of true positive TP examples contains the first 10 objects of the Table 1. and the set of false positive FP contains only two cases (serial number 31 and 32). Therefore the value of $\gamma_1$ evaluates to 10/12=0.8333. In the second phase, some bits of the rule under consideration can undergo mutation. It is to be noticed that the bits that represent the pre-discovered FCR are not allowed to change during mutation and kept fixed. A mutated version of the FCR under consideration may be 100111001000000001 which maps to a rule R'= $(A_1$= Small$) \rightarrow D_1 \oplus A_3$= (Large) $(D_3)$. Let us check if this rule meets the criteria of a FCCR. The value of TP' for R' is 2 and FP'=0; the value of $\gamma_2$ for this FCCR computes to TP'/TP+FP=2/12=0.167. The confidence factor of rule R' is 1. Hence R' is a perfect FCCR. This FCCR is an interesting rule in the sense that it asserts that in most cases when $A_1$ is small then the decision is $D_1$. However, if along with $A_1$ being small, $A_3$ takes the value large, which happens very rarely, and then the class of this rule becomes $D_3$. This rule gives us an opportunity to change the decision in rare circumstances.

For the second phase mutation rate was kept at 0.2 and mutation was applied over 50 generations. For the dataset used in this experimentation we discovered the FCCRs as shown in Table 3.There is no censor to the third rule in Table 2. Hence it is included in the set of FCCRs with its *Unless* part empty.

Table 3: FCCRs Discovered

| **Sr. No.** | **FCCRs** | **$\gamma_1$:$\gamma_2$** |
|---|---|---|
| 1. | $(A_1$=Small$) \rightarrow D_1 \oplus (A_3$=Large $\vee$ $A_5$ = small$)$: $D_3$ | 0.883:0.167 |
| 2. | If $(A_1$=Medium$) \rightarrow D_2 \oplus (A_3$=Large $\vee$ $A_5$=Small$)$: $D_3$ | 0.883:0.167 |
| 3. | If $(A_1$=Large$) \rightarrow D_3 \oplus ()$ | 1:0 |

## 5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented a genetic algorithm approach for discovery of Fuzzy Censored Classification Rules from datasets. The discovery of FCCRs provides the advantages of Fuzzy Classification Rules (FCRs) (like dealing with uncertain, imprecise and vague conditions inherent in real world situations), as well as it offers an excellent exception handling mechanism. Discovery of a classifier in the form of FCCRs makes it more interesting as the classifier is now capable of giving right predictions even in exceptional cases. Learning of FCCRs may also enhance the capabilities of automated and expert systems. The proposed discovery may also prove to be very useful for fuzzy control applications to predict the behavior of a system in rare circumstances. Another benefit of a classification model consisting of FCCRs is that it can make predictions with variable certainty. The major limitation of the proposed system is that fuzzifying the attributes in pre-processing phase employing the same membership function is a very simple idea. This fuzzification technique may not suit the distribution of data values across all the predicting attributes with respect to the class attribute. In fact, use of a single fuzzy membership function with same number of linguistic labels for all the predicting attributes may lead to the discovery of a classifier with unacceptable predictive accuracy.  As the future work, the proposed system needs to be applied and tested on some real world datasets in the fields of medical diagnosis and fuzzy controllers. For real world datasets, the fuzzy membership functions to be employed and number of fuzzy partitions to be induced shall definitely require separate tuning for different attributes.

## 6. REFERENCES

[1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy, (1996) Advances in Knowledge Discovery and Data Mining. AAAI Press.

[2] R. S. Michalski and P. H. Winston, (1986)  “Variable precision logic,” Artificial Intelligence, Vol. 29, No. 2, pp. 121–146.

[3] K. K. Bharadwaj & B. M. Al-Maqaleh, (2005) “Evolutionary approach for automated discovery of censored production rules,” Trans Eng Comput Technol, Vol. 10, pp. 147–152.

[4] S. Saroj & K. Bharadwaj, (2007) “A parallel genetic algorithm approach for automated discovery of censored production rules,” in Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications, pp. 435–441.

[5] L. A. Zadeh, “Fuzzy logic = computing with words, (May 1996)” IEEE Transactions on Fuzzy Systems, Vol. 4, No. 2, pp. 103–111.

[6] H. Ishibuchi, K. Nozaki, N. Yamamoto, & H. Tanaka, (Aug 1995)  “Selecting fuzzy if-then rules for classification problems using genetic algorithms,” IEEE Transactions on Fuzzy Systems, Vol. 3, No. 3, pp. 260–270.

[7] F. Herrera, (2008) “Genetic fuzzy systems: taxonomy, current research trends and prospects,” Evolutionary Intelligence, Vol. 1, No. 1, pp. 27–46.

[8] A. Fernandez, S. Garcia, M. Deljesus, & F. Herrera (Sep. 2008) “A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets,” Fuzzy Sets and Systems, Vol. 159, No. 18, pp. 2378–2398.

[9] Cordon O., Herrera F., & Villar P. (2000)  “Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing,” International Journal of Approximate Reasoning, Vol. 25, No. 3, pp. 187–215.

[10] R. R. F. Mendes, F. B. Voznika, A. A. Freitas, & J. C. Nievola, (2010) “Discovering fuzzy classification rules with genetic programming and co-evolution,” Proceedings of the Genetic and Evolutionary Computation Conference GECCO2001, Vol. 2168, pp. 314–325.

[11] D. E. Goldberg, (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, 1st ed. Addison-Wesley Professional.

[12] O. Cordon, F. Herrera, & P. Villar, (Aug 2001) “Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base,” IEEE Trans. Fuzzy Syst., Vol. 9, No. 4, pp. 667–674.

[13] O. Cordón, F. Herrera, L. Magdalena, & P. Villar,(Aug 2001) "A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base," Information Sciences, Vol. 136, No. 1–4, pp. 85–107.

[14] H. Ishibuchi & T. Murata, (1996) "A genetic-algorithm-based fuzzy partition method for pattern classification problems," Physica-Verlag, vol., in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing. No. 8 in Studies in Fuzziness and Soft Computing, pp. 555–578.

[15] P. Thrift, (1991) "Fuzzy logic synthesis with genetic algorithms," in Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 509–513.

[16] H. Ishibuchi, T. Nakashima, & T. Kuroda, (1999) "A hybrid fuzzy genetics-based machine learning algorithm: hybridization of Michigan approach and Pittsburgh approach," in 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings, Vol. 1, pp. 296–301.

[17] H. Ishibuchi, Y. Nakashima, & Y. Nojima, (2011) "Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning," Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 15, no. 12, pp. 2415–2434.

[18] H. Ishibuchi, T. Yamamoto, & T. Nakashima, (Apr. 2005) "Hybridization of fuzzy GBML approach

[18] H. Ishibuchi, T. Yamamoto, & T. Nakashima, (Apr. 2005) "Hybridization of fuzzy GBML approaches for pattern classification problems," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 35, No. 2, pp. 359–365.

[19] R. Palm, D. Driankov, & H. Hellendoorn (1997), Model based fuzzy control: fuzzy gain schedulers and sliding mode fuzzy controllers. Springer.

[20] W. Pedrycz (1996), Fuzzy modelling: paradigms & practice, vol. 7. Kluwer Academic Pub.

[21] H. Ishibuchi, T. Nakashima, & M. Nii (2005), Classification and modeling with linguistic information granules: advanced approaches advanced approaches to linguistic data mining. Springer.

[22] L. I. Kuncheva (2000), Fuzzy classifier design, vol. 49. Physica Verlag.

[23] Y. Yuan & H. Zhuang, (Nov 1996) "A genetic algorithm for generating fuzzy classification rules," Fuzzy Sets Syst., Vol. 84, No. 1, pp. 1–19.

# Appendix –A

Table A-1. Synthetic dataset in normalized from

| Att1 | Att2 | Att3 | Att4 | Att5 | -Class |
|---|---|---|---|---|---|
| 0.12 | 0.11 | 0.05 | 0.01 | 0.55 | 1 |
| 0.18 | 0.24 | 0.04 | 0.3 | 0.57 | 1 |
| 0.13 | 0.18 | 0.08 | 0.2 | 0.5 | 1 |
| 0.16 | 0.01 | 0.05 | 0.05 | 0.59 | 1 |
| 0.18 | 0.05 | 0.15 | 0.04 | 0.54 | 1 |
| 0.17 | 0.08 | 0.59 | 0.56 | 0.9 | 1 |
| 0.19 | 0.1 | 0.60 | 0.6 | 0.8 | 1 |
| 0.05 | 0.02 | 0.56 | 0.61 | 0.88 | 1 |
| 0.06 | 0.07 | 0.54 | 0.54 | 0.87 | 1 |
| 0.09 | 0.08 | 0.52 | 0.55 | 0.89 | 1 |
| 0.54 | 0.12 | 0.05 | 0.1 | 0.54 | 2 |
| 0.50 | 0.13 | 0.05 | 0.01 | 0.56 | 2 |
| 0.57 | 0.14 | 0.08 | 0.02 | 0.49 | 2 |
| 0.51 | 0.15 | 0.06 | 0.03 | 0.48 | 2 |
| 0.52 | 0.52 | 0.05 | 0.04 | 0.59 | 2 |
| 0.56 | 0.6 | 0.52 | 0.6 | 0.92 | 2 |
| 0.48 | 0.58 | 0.54 | 0.61 | 1.0 | 2 |
| 0.49 | 0.79 | 0.52 | 0.6 | 0.89 | 2 |
| 0.46 | 0.8 | 0.6 | 0.52 | 0.87 | 2 |
| 0.53 | 0.82 | 0.61 | 0.58 | 0.88 | 2 |
| 0.85 | 0.2 | 0.01 | 0.1 | 0.54 | 3 |
| 0.86 | 0.04 | 0.05 | 0.03 | 0.55 | 3 |
| 0.9 | 0.1 | 0.18 | 0.08 | 0.59 | 3 |
| 1.0 | 0.54 | 0.14 | 0.14 | 0.6 | 3 |
| 0.84 | 0.56 | 0.09 | 0.18 | 0.57 | 3 |
| 0.90 | 0.58 | 0.57 | 0.55 | 0.99 | 3 |
| 0.84 | 0.52 | 0.54 | 0.54 | 0.98 | 3 |
| 0.92 | 0.78 | 0.52 | 0.56 | 0.89 | 3 |
| 0.94 | 0.84 | 0.6 | 0.57 | 0.85 | 3 |
| 0.87 | 0.9 | 0.55 | 0.6 | 0.98 | 3 |
| 0.1 | 0.56 | 0.85 | 0.58 | 0.04 | 3 |
| 0.14 | 0.87 | 0.95 | 0.09 | 0.06 | 3 |
| 0.58 | 0.12 | 0.9 | 0.12 | 0.15 | 3 |
| 0.54 | 0.11 | 0.87 | 0.07 | 0.16 | 3 |

## Authors

Dr. Saroj is working as an Associate Professor with the Department of Comp uter Science & Engineering, Guru Jambheshwar University of Science and Technology, Hisar. She has earned her Ph.D Degree in Computer Science in 2010 from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. Her areas of research interest include Knowledge Discovery & Data Mining, Evolutionary Algorithms and Machine learning.

Renu Bala is Ph.D Student in Department of Computer Science & Engineering, Guru Jambheshwar University of Science & T echnology, Hisar, India.