Pertanika J. Sci. & Technol. 28 (1): 91 - 116 (2020)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

A Genetic Algorithm Approach for Discovering Fuzzy Hierarchical Censored Classification Rules (FHCCRs)

Renu Bala* and Saroj Ratnoo

Departmnent of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar, India

ABSTRACT

Most of the classification algorithms discover flat Fuzzy Classification Rules (FCRs) in '*If-Then*' form. The knowledge discovered in the form of FCRs allows us to deal with vague, inexact and incomplete premises, however, it ignores exceptions and hierarchies that may exist in data. The simple FCRs enlarge the size of Rule Bases (RBs) with the presence of duplicate clauses that can be removed by arranging the rules in a hierarchical fashion. Moreover, such rules infer incorrect conclusions in the presence of exceptional conditions. This paper proposes the discovery of accurate, interpretable and interesting rules in a novel form named as Fuzzy Hierarchical Censored Classification Rules (FHCCRs) using a Genetic Algorithm approach. The GA design for discovering FHCCRs includes designing of suitable encoding scheme, fitness function and genetic operators. The suggested approach works in three phases: i) fuzzifying a dataset in a pre-processing step, ii) applying a genetic algorithm for discovering FHCCRs and iii) merging FHCCRs into bigger hierarchies in a post-processing step. The proposed approach is applied to five benchmark datasets. It successfully discovers FHCCRs which contain exceptions (also referred as censors) as well as hierarchies. The knowledge discovered in the form of FHCCRs enriches rule bases in respect of interpretability and interestingness.

Keywords: Classification rule discovery, fuzzy censored classification rules (FCCRs), fuzzy hierarchical classification rules (FHCRs), fuzzy hierarchical censored classification rules (FHCCRs)

ARTICLE INFO

Article history:
Received: 28 June 2018
Accepted: 19 April 2019
Published: 13 January 2020

E-mail addresses:
renu0805@gmail.com (Renu Bala)
ratnoo.saroj@gmail.com (Saroj Ratnoo)
* Corresponding author

INTRODUCTION

In real world problem domains, a machine has to make predictions using inadequate, vague and uncertain information. Fuzzy

ISSN: 0128-7680
e-ISSN: 2231-8526

Rule Based Systems (FRBSs) are proficient systems for reasoning with ambiguous, imprecise and/or incomplete information (Zadeh, 1996). Most of the FRBSs consist of Fuzzy Classification Rules (FCRs) that ignore hierarchical relationships among the data/ class labels. Such conventional FRBSs are not suitable for domains, where knowledge can be better expressed at various levels of abstraction. Therefore, researchers have suggested knowledge structures like Hierarchical Production Rules (HPRSs) (Al-Maqaleh & Bharadwaj, 2005; Bharadwaj & Saroj, 2009; Bharadwaj & Kandelwal, 2007), Hierarchical Censored Production Rules (HCPRs) (Bharadwaj & Saroj, 2010; Bharadwaj & Jain, 1992; Jain & Bharadwaj, 1998) and Fuzzy Hierarchical Censored Production Rules (FHCPRs) (Neerja & Bharadwaj, 1996) to support the discovery of hierarchical fuzzy classification rules.

While designing hierarchical classification systems, we need to consider variable certainty and specificity as the two aspects of variable precision logic. A system that gives more certain answers given more time (resources) is called a variable certainty system and a system that gives more specific answers given more time (resources) is called variable specificity system (Michalski & Winston, 1986). Certainty refers to the degree of belief in a conclusion and specificity refers to the degree of details contained in a Rule Base (RB). The two aspects of variable precision logic (certainty and specificity) are associated with accuracy and interpretability - the two fundamental characteristics of classifiers/RBs in data mining. Accuracy is the capability of a RB to closely represent the real-world knowledge whereas interpretability of a RB is the power to state the behavior of a real system in an interpretable way (Gacto et al., 2011). In this context, accuracy provides a measure of certainty while specificity relates to interpretability.

Accuracy and interpretability are indeed important performance monitoring metrics but are not sufficient to measure the real interestingness of a RB. A RB is considered interesting if the knowledge represented by it is not only formerly unknown but it is also in contrast to the original beliefs of its users (Kontonasios et al., 2012). Although interestingness is a subjective term, yet unexpected and exceptional knowledge is considered interesting. Discovery of exceptions is interesting because exceptions are unusual conditions that contradict the prior knowledge about the domain, add curiosity and improve the quality of decision making in those rare circumstances when the default or generalized rules cease to work. Exceptions, also termed as censors in the literature (Bharadwaj & Al-Maqaleh, 2007a), have low support, and therefore, it is not possible to discover exceptions using the typical classification rule discovery methods. Most of the classification algorithms focus on the generality of the knowledge discovered, whereas exceptions make rules specific.

In this paper, we propose a genetic algorithm approach to discover Fuzzy Hierarchical Censored Classification Rules (FHCCRs) which are the integration of Fuzzy Censored

Classification Rules (FCCRs) and Fuzzy Hierarchical Classification Rules (FHCRs) (Bala & Ratnoo, 2018). The proposed system discovers accurate and interpretable fuzzy classification rules. In addition, these rules are interesting and have the ability to make predictions with variable certainty and specificity. We have employed two quantitative measures, the degree of subsumption and coefficient of similarity, to determine the hierarchical relationships in FHCCRs. The proposed approach is tested on benchmark data sets from the UCI machine learning repository. The approach initially generates FCRs which are, subsequently, converted into FHCCRs by appending exceptions/censors and hierarchies.

The rest of the paper has the following structure. Second part describes the supporting knowledge structures for discovering hierarchical rules with exceptions/censors and the related work to contextualize this research. Third part describes the proposed system to discover FHCCRs. It includes the fuzzification process, GA design, and a post-processing scheme to merge individual FHCCRs. It also explains the concepts of the degree of subsumption and degree of similarity. The suggested approach is employed in experimental design followed by the results obtained and lastly, conclusion and future scope of this research have been discussed.

RELATED WORK

Intelligent and reasoning based systems have to work in an environment of uncertainty and vagueness. Hence, the Rule Bases (RBs) of these systems consist of Fuzzy Classification Rules (FCRs). Researchers have been working to make these rule bases more concise and useful by accommodating exceptional clauses and hierarchies in these rule bases. This section describes novel rule structures and takes account of the research carried out for discovering knowledge in these various advanced rule forms to create RBs for intelligent systems.

Fuzzy Censored Classification Rules

Exceptions (also referred as Censors in this work) are the rare conditions which change the behavior of a default rule. Exceptions/ Censors pertain to limited instances in data and, hence, often get ignored as machine noise, if not tackled separately. An FCCR is a combination of an FCR and a Censor Production Rule (CPR) (Michalski & Winston, 1986). FCCRs have discussed in detail in Bala (2012).

FCCRs cover one aspect of variable precision, logic, i.e., variable certainty, but specificity remains constant here. Next Section describes FHCCRs which support variable specificity also in drawing conclusions.

Fuzzy Hierarchical Censored Classification Rules

Fuzzy Hierarchical Censored Classification Rules (FHCCRs) are the rules that contain censor conditions as well as hierarchical information. An FHCCRs is represented as:

If P is X Then C_kUnless E is Y: C_k'

Generality [C_g]

Specificity [C_{s1}, C_{s2}]

Here, 'E' represents the censor conditions, C_k represents the rule class, C_g is the general class and C_{s1}, and C_{s2} are the specific classes. The rule class C_k may change to C_k' when the censor condition E is true. The general class represents the most general concept and specific classes show the more specific concepts in the hierarchies. The following real world example shows the illustration of FHCCRs.

/* Level 0*/
If (S is household servant for family F) *Then* (S does household_work) *Unless* (S is Sick or S is on leave: S does not work) *Generality* [] *// The Rule Class itself is the most General class*
Specificity [S cooks for family F, S drives for family F]

/* Level 1*/
If (S has **good** cooking skills) *Then* (S cooks for family F) *unless* (Family F is outside: S does not cook for family F) *Generality* [S does household_work] *Specificity* [S cooks breakfast, S cooks lunch, S cooks dinner]
If (S has **good** driving skills) *Then* (S drives for family F) *Unless* (S is defaulter: S does not drive for family F)
Generality [S does household_work] *Specificity* [S drives bike, S drives car]

/* Level 2*/
If (time is **morning**) *Then* (S cooks breakfast) *Unless* (Family F wakes up late on Sundays: S does not cook breakfast) *Generality* [S cooks for family F] *Specificity*[]
If (time is **noon**) *Then* (S cooks lunch] *Unless* (Family F eats out: S does not cook lunch)
Generality [S cooks for family F] *Specificity*[]
If (time is **night**) *Then* (S cooks dinner] *Unless* ()
Generality [S cooks for family F] *Specificity*[]
If (S travels **short** distance) *Then* (S drives bike) *Unless* ()
Generality [S drives for family F] *Specificity* []
If (S travels **long** distance) *Then* (S drives car) *Unless* ()
Generality [S drives for family F] *Specificity* []

The above FHCCRs can further be merged into a single tree as shown in Figure 1. The root node in a FHCCR tree signifies the most general class and any child node is a specific case of its parent node. As we traverse towards the leaf nodes, the classes become more and more specific. Censors may be present at any level in the hierarchy.

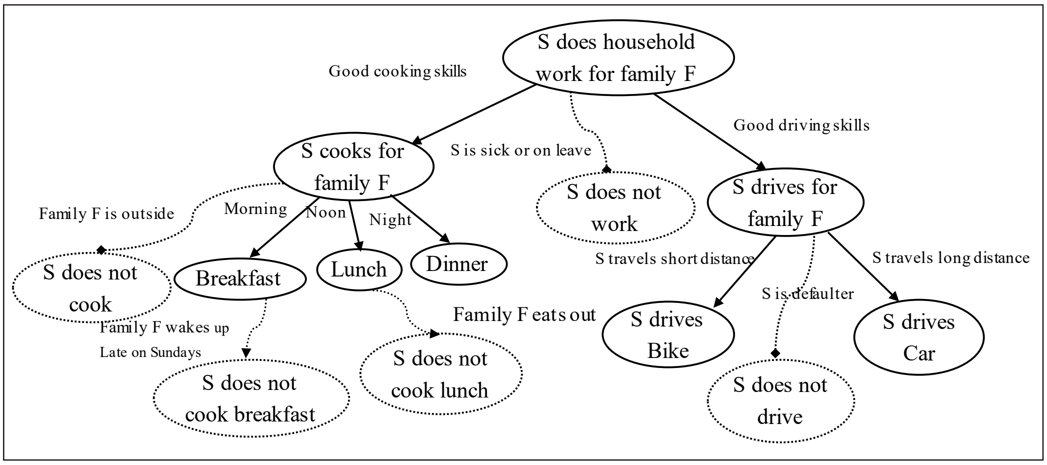

Figure 1. An example FHCCRs tree

Censors present at upper levels in the hierarchy are inherited at the lower levels as well. Since each rule in the hierarchy inherits all the properties of its parent FHCCR, it is not required to list all such properties repetitively. Hence, in the tree representation, redundancy is minimized in the listing of the properties. The solid circles represent the classes and solid arrows denote the properties. Censor conditions are represented along the dashed lines and dashed circles show the classes when censor/exception condition(s) is/are satisfied.

For discovering the best set of FHCCRs, one needs to search through all possible candidate FHCCRs for a dataset. In this context, it is an optimization problem that can be solved using a Genetic Algorithm. Applying an approach for discovering FHCCRs requires an evaluation/fitness function to guide the search towards an optimal solution. Therefore, we need to quantify the goodness of hierarchies. For this purpose, we have used degree of subsumption and coefficient of similarity- two quantitative measures- to decide about the levels of classes in the hierarchical framework (Al-Maqaleh & Bharadwaj, 2005; Bharadwaj & Saroj, 2010; Bharadwaj & Al-Maqaleh, 2007b). Degree of subsumption decides the hierarchical levels (Generality/ Specificity) among classes. If Class C_k subsumes class C_s then C_k is more general class than C_s, i.e. C_s is the specific class of C_k. Coefficient of similarity gives more comprehensible results if the degrees of subsumption between classes C_k and C_s are equal both ways. These measures are explained in detail in the proposed system.

Rule Mining using Genetic Algorithms

Genetic algorithms are random but guided search methods for solving complex optimization problems. These start with a random population of candidate solutions and apply genetic operators (selection, recombination and mutation) to form improved solutions from the better fit parents in successive generations by following Darwinian theory of evolution. The possible classification rules that can be learned from a dataset can be enormously large. Finding an optimal rule set that can perform effective classification is no less a challenging task. GAs, being global optimization tool, have been extensively used for classification rule mining (Freitas, 2002a; Freitas, 2002b). The block diagram for discovering classification rules is given in Figure 2.

Research for Discovering Knowledge in Various Rule Forms

FCRs have demonstrated their ability in a wide spectrum of applications in the domain of control (Palm et al., 1997), modeling (Pedrycz, 1996), and data mining problems (Ishibuchi et al., 2005b; Kuncheva, 2010). Therefore, there have been many attempts to discover FCRs from real-valued datasets (Cordón et al., 2000; Fernández et al., 2009; Herrera, 2008; Ishibuchi et al.,1995; Ishibuchi et al., 2005a; Ishibuchi et al., 2005b; Ishibuchi et al., 2011; Mendes et al., 2001). Most of these approaches discover the knowledge at a single conceptual level that results into RBs of large size. It may increase the accuracy of the RB but influences the interpretability adversely.

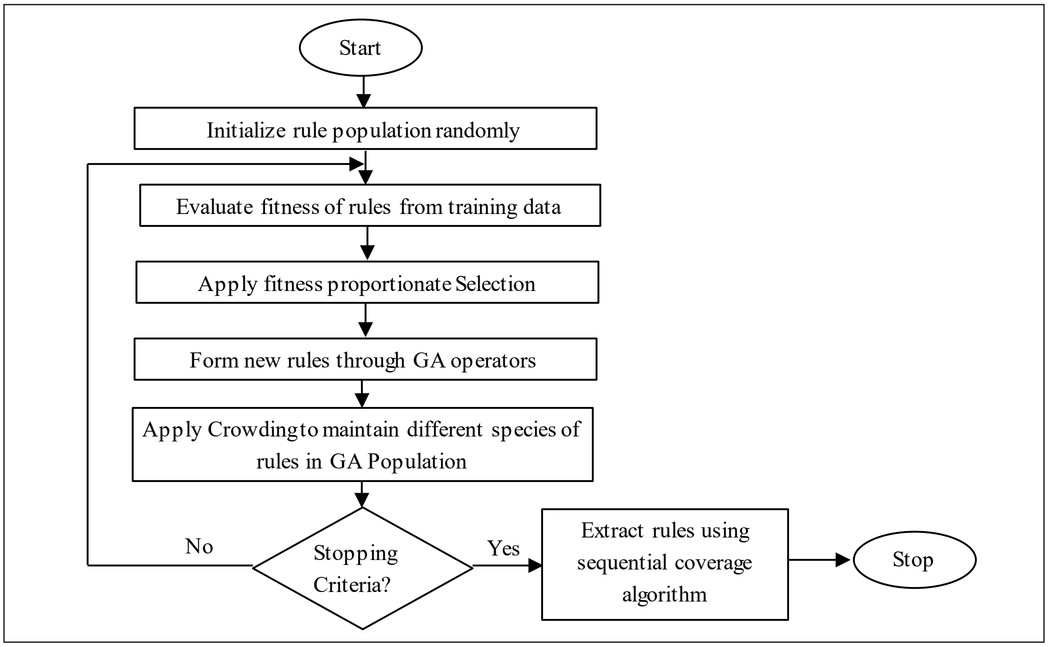

Figure 2. Block diagram of GA for discovering Classification Rules

Accuracy and interpretability are conflicting criteria; therefore, researchers depend on achieving the best trade-off between accuracy and interpretability. Several efforts have been made to reach an acceptable trade-off between accuracy and interpretability (Bala & Ratnoo, 2016; Gacto et al., 2011; Ishibuchi et al., 1997; Ishibuchi et al., 2001; Ishibuchi et al., 2011; Sanz et al., 2010). Ishibuchi and colleagues (1997, 2001, 2005a, 2005b & 2011) had proposed a GA for rule selection problem that tried to maximize accuracy and minimize the number of rules. In addition, multi-objective evolutionary algorithms have also been suggested for rule learning problems. These approaches discover non-dominated Pareto optimal solutions by considering accuracy and interpretability (in terms of the number of rules and the number of antecedent conditions per rule) as the optimization criteria (Ishibuchi et al., 1997; Ishibuchi et al., 2001). All these approaches discover knowledge in the form of FCRs which have the following shortcomings:

- FCRs severely fragment the knowledge, thereby, resulting in a large number of rules.
- FCRs, as an underlying rule structure for discovering classification rules, do not exhibit the two aspects of variable precision logic, i.e., variable certainty and variable specificity.
- FCRs simply ignore the exceptions as a noise and, hence, are unable to cope with exceptional/unusual conditions.

The problem of discovering flat classification rules has been extensively studied in the area of data mining and machine learning (Freitas, 2008). Hence, in the past few decades, discovery of the Hierarchical Classification Rules (HCRs) has become the priority of researchers in field of rule mining (Cerri et al., 2013; Cerri et al., 2012; Davies et al., 2007; Rousu et al., 2006; Secker et al., 2007; Secker et al., 2010; Sun et al., 2004; Sun & Lim, 2001; Tsumoto, 2003). A knowledge base organized in a hierarchical form is not only interpretable, it can also make predictions at multiple levels of abstractions, i.e., it can handle variable precision logic with respect to specificity. The most efficient and easy to understand underlying rule structures to support the discovery of the hierarchical classification rules are HPRs (Al-Maqaleh & Bharadwaj, 2005; Bharadwaj & Saroj, 2009; Bharadwaj & Kandelwal, 2007), HCPRs (Bharadwaj & Jain, 1992) and FHCPRs (Neerja & Bharadwaj, 1996). Hierarchical multi-label classification is another complex challenging task that requires discovery of rules in hierarchical form. In hierarchical multi-label classification problems, an instance is assigned to more than one classes out of hundreds or thousands of the classes (Cerri et al., 2012). Popular examples of the hierarchical multi-label classification problems are the task of text classification (Rousu et al., 2006; Sun et al., 2004; Sun & Lim, 2001) and protein function prediction (Cerri et al., 2013; Sun et al., 2004).

Another important aspect of real world knowledge is interestingness which has been given considerably focus in the data mining literature. Silberschatz and Tuzhilin (Kontonasios et al., 2012; Silberschatz & Tuzhilin, 1996; Silberschatz & Tuzhilin, 1995) had dealt with the issue of interestingness. They had considered the exceptions as interesting pieces of knowledge that challenged the common beliefs. Suzuki and colleagues (Suzuki, 2002; Suzuki, 2004; Suzuki & Shimura, 1996; Suzuki & Zytkow, 2000) had done extensive work in the domain of exception discovery. They had classified exceptions in several categories and discovered exceptions in the form of rule pairs and rule triplets. They had addressed the task of dependence modeling, and their algorithm discovered a large number of exceptions, which made the discovered knowledge unsuitable for human insight and analysis. A classification algorithm based on the evolutionary approach for discovering comprehensible rules with exceptions in the form of Censored Production Rules (CPRs) is presented in (Bharadwaj & Al-Maqaleh, 2007a). A genetic algorithm approach has also been proposed to discover Fuzzy Censored Classification Rules in (Bala, 2012). A genetic programming-based intelligent miner has also been proposed to mine rules with fuzzy hierarchies with exceptions at every level (Bharadwaj & Saroj, 2010; Bharadwaj & Al-Maqaleh, 2007b). Vashishtha et al. (2013) had discovered classification rules with intra and inter-class exceptions. However, their algorithm was designed to work with datasets containing discretized /nominal attributes only. Another contribution for discovering exceptions has been made to discover tuned fuzzy classification rules with Intra and inter-class exceptions (Bala & Ratnoo, 2016). In this work, each attribute is fuzzified by finding an optimal combination of the type of MFs (triangular, trapezium) and the number of linguistic terms (varying from 1 to 5). In this research work, we take up the task of discovering knowledge in the form of FHCCRs using a genetic algorithm approach.

THE PROPOSED SYSTEM

This section describes the proposed approach for discovering FHCCRs in a step by step manner. It includes a pre-processing step to fuzzify the attributes, a genetic algorithm to discover FHCCRs and a post-processing step to merge FHCCRs into FHCCR trees. The overall flow chart of the proposed system is shown in Figure 3.

Fuzzification Process

The dataset is normalized by using the following formula in Equation 1.

$$A_{ij}' = \frac{A_{ij} - min(A_j)}{max(A_j) - min(A_j)} \qquad [1]$$

A'_{ij} is the normalized value of the j^th^ attribute of the i^th^ instance. After normalization the dataset is fuzzified by applying algorithm given by Bala and Ratnoo (2016). This algorithm

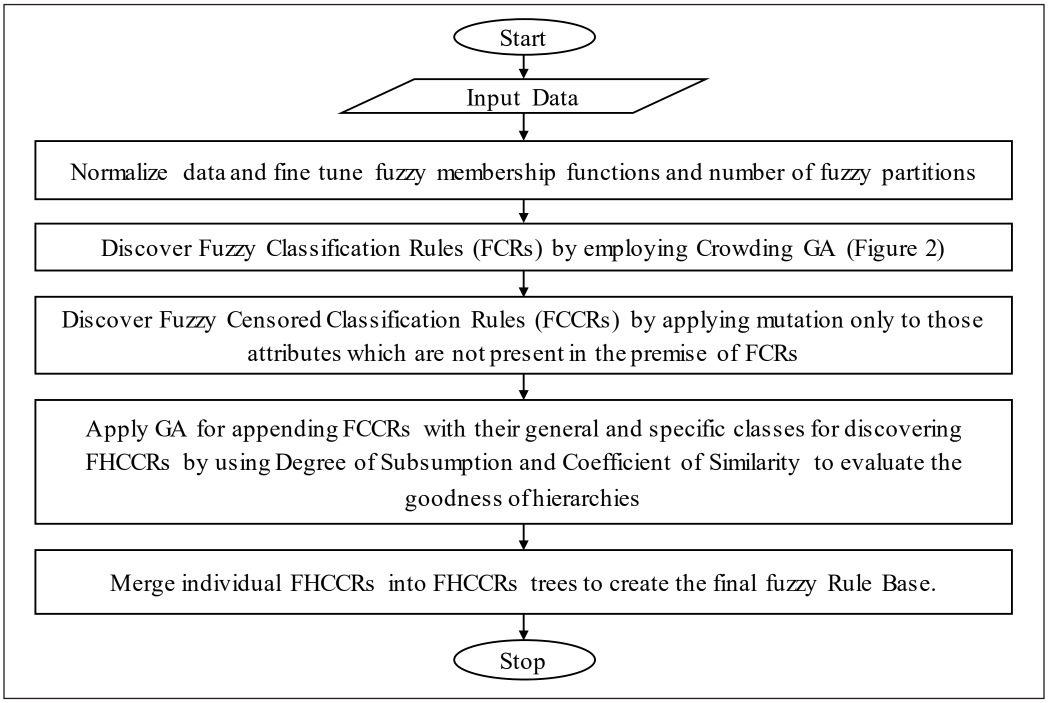

Figure 3. Flowchart for the overall process of the proposed system

produces optimal fuzzy partitions for attributes in relation to class labels. For each attribute, it selects an optimal combination of a Fuzzy Membership Function (FMF) and the number of linguistic partitions out of a given set of FMFs (i.e., Triangular and Trapezium) and a given set of linguistic partitions (i.e.,2:(small, medium), 3:(small, medium large), 4: (small medium, medium large), 5:(small, small medium, medium, medium Large, Large)). The algorithm computes the gain ratio for all the possible combinations of FMFs and number of linguistic partitions for an attribute using the following Equation 2:

$$GR(A^k) = \frac{IG(A^k)}{SplitInfo(A^k)} \qquad [2]$$

Measuring Goodness of Hierarchies

This section explains the use of the degree of subsumption and the coefficient of similarity to quantify the goodness of hierarchies.

Degree of Subsumption. Degree of subsumption is a quantitative measure to decide the hierarchical relation of generality/specificity among the classes. To compute the degree of subsumption between two classes C_k and C_s, first, we need to spot their defining properties. The defining properties for the class C_k are the distinct linguistic labels (i.e., small, medium and large), for the attributes of a given dataset, which have recalled more than a user-defined threshold value (θ_t) with respect to the class C_k (Bharadwaj & Saroj, 2009; Bharadwaj & Kandelwal, 2007). The values for recall, denoted by x and y, for the i[th] property in the class C_k and j[th] property in the class C_s are computed as below in equations 3 and 4:

$$x_i = \frac{|(\mu(P_i) \quad _\alpha \wedge C_k)|}{|C_k|} \qquad [3]$$

$$y_j = \frac{|(\mu(P_j) \quad _\alpha \wedge C_s)|}{|C_s|} \qquad [4]$$

The value of α needs to be greater than 0.51. This value has been chosen through experimental tuning. The set of defining properties *S(C*$_k$*)* and *S(C*$_s$*)* for the classes C_k and C_s are captured as Equation 5 and 6 below:

$$S(C_k) = \forall P_i \in U; \;\; x_i > \theta_t \qquad [5]$$

$$S(C_s) = \forall P_j \in U; \;\; y_j > \theta_t \qquad [6]$$

The subsumption between i[th] and j[th] properties of defining sets *S(C*$_k$*)* and *S(C*$_s$*)* is computed by using following Equation 7, 8 and 9.

$$subsume(S(C_k(i)), S(C_s(j)) = 1 \qquad if(x \leq y)\ and\ (i = j) \qquad [7]$$

$$subsume(S(C_k(i)), S(C_s(j)) = y \qquad if(x > y)\ and\ (i = j) \qquad [8]$$

$$subsume(S(C_k(i)), S(C_s(j)) = 0 \qquad if\ (i \neq j) \qquad [9]$$

The overall subsumption between the two classes is given by Equation 10:

$$deg\,r\,ee\_subsumption(C_k, C_s) = \sum_{i=1}^{|S(C_k)|} \sum_{j=1}^{|S(C_s)|} \frac{subsume(S(C_k(i)), S(C_s(j)))}{|S(C_k)|} \qquad [10]$$

Here, we are calculating overall subsumption degree between every pair of the classes. Therefore, double summation is applied.

A subsumption matrix can be prepared using Equation 11. θ_s denotes the threshold value for the degree of subsumption which is kept at 0.6.

$$\underset{i,j=1\dots n}{Subsumption\_matrix}[C_i,C_j] = \begin{cases} -1 \; if \; i = j \\ 0 \; Subsumption(C_i,C_j) < \theta_{ts} \\ 0 \; Subsumption(C_j,C_i) > Subsumption(C_i,C_j) \\ Subsumption(C_i,C_j) \; otherwise \end{cases}$$

[11]

Coefficient of Similarity

A coefficient of similarity is required to decide the hierarchy among those classes whose subsumption degree is same both ways. The coefficient of similarity (S) of attributes between two classes C_k and C_s is defined on the basis of a 2 × 2 contingency table (Table 1).

Table 1
Contingency table

S(C_k)(C₁) ↓	*S(C_S)(C₂)* →	
	Observed	Not observed
Observed	p	q
Not observed	r	s

$$\chi^2 = \frac{N(ps - qr)^2}{M} \quad [12]$$

$$N = p + q + r + s \quad [13]$$

$$M = (p + q)\ (r + s)\ (p + r)\ (q + s) \quad \|14\|$$

$$S = \sqrt{\frac{\chi^2}{N(k-1)}} \quad [15]$$

In the above Equation 12, 13, 14 and 15, N is the total number of properties present; k is the degree of freedom. Degree of freedom is the number of independent quantities in the final calculation of a stastical distribution and it is calculated by $k = (Number\ of\ Rows\ - 1) \times (Number\ of\ Col\ - 1)$. For a two by two contigency table the degree of freedom is 1. The value of S shall always lie between 0 and 1. Higher the value of S, more is the similarity between the classes involved (Bharadwaj & Saroj, 2009; Bharadwaj & Kandelwal, 2007).

A similarity matrix can be prepared by using Equation 16. θ_{sim}, in Equation 16 represents the threshold for coefficient of similarity which is taken as 0.6.

$$\underset{i,j=1\ldots.n}{Similarity\_matrix[C_i,C_j]} = \begin{cases} 1\ if\ \ i == j \\ 0\ \ Similarity(C_i, C_j) < \theta_{tsim} \\ Similarity(C_i,C_j)\ \ otherwise \end{cases} \quad [16]$$

The coefficient of similarity is helpful in deciding the hierarchy between the classes when a general class subsumes two classes with the same degree of subsumption.

The Genetic Algorithm Design

This section describes the proposed GA approach to discover FHCCRs. It illustrates encoding scheme, generation of the initial population, fitness function and genetic operators.

Individual's Encoding. A set of four consequent blocks encodes the hierarchical chromosome with exceptions, as a candidate solution. The first block represents default rule, i.e., FCR. The second block captures the censor/exceptional condition(s) along with the class which gets predicted when censor conditions are satisfied. The third block contains the general class and the fourth block holds the specific classes. Two constraints that need to be enforced on individual FHCCRs are – 1) The properties present in the premise and exceptional parts are mutually exclusive and; 2) The rule class, the general class and the specific classes need to be all distinct.

Figure 4 shows the encoding of two chromosomes and their mapping to the corresponding FHCCRs. We have adopted the Michigan approach with a pure binary string for encoding the premise part of a chromosome. A block of n bits signifies n consecutive linguistic fuzzy variables in the order- 'small', 'small-medium', 'medium', 'medium-large' and 'large'. Within a block, a '1' bit marks the presence of the corresponding linguistic term, whereas a '0' bit denotes its absence. A block with all bits set to 1 or 0 is treated as a '*don't care*' state which indicates the absence of an attribute from the rule. The consequent

Block 1						**Block 2**			**Block 3**	**Block 4**
A1	A2	A3	A4	A5	Class	A6	A7	Censor class	Generality	Specificity
10	010	0010	10	00000	2	00010	0100	7	1	4 5 0
A1	A2	A3	A4	A5	Class	A6	A7	Censor	Generality	Specificity
10	010	0010	01	00000	3	00001	1000	8	1	6 00

Genotype (Chromosomes)

If (A_1 is small) $\wedge$ (A_2 is medium) $\wedge$ (A_3 is medium-large) $\wedge$ (A_4 is small)$\rightarrow C_2 \neg$ (A_6 is medium-large) $\wedge$ (A_7 is small-medium) : C_7; Generality[C_1] Specificity[C_4,C_5]
If (A_1 is small) $\wedge$ (A_2 is medium) $\wedge$ (A_3 is medium-large) $\wedge$ (A_4 is large) $\rightarrow C_3$ $\neg$ (A6 is large) $\vee$ (A7 is small) : C_8; Generality[C_1] Specificity [C_6]

Phenotype (FHCCR)

Figure 4. Encoding scheme

part contains the class label of the rule. The second block follows a similar strategy to represent exceptions and their corresponding classes. The attributes in block 1 and block 2 are mutually exclusive as per the first constraint. The Generality part contains the label for the general class of the rule. The next block consists of three bits – each for one of the specific classes - which indicates that there are at most three specific classes.

Initial Population. Initially, a set of FCRs is generated from data using the algorithms "Evolving FCRs" and "Crowding" given in Bala and Ratnoo (2016). This first phase of the design discovers a set of FCRs which occupies the first block of the FHCCRs. The second phase transforms the FCRs into a set of FCCRs based on the algorithm given in Saroj and Bharadwaj (2007) , and modified for discovery of fuzzy rules.

The third phase of the design creates an initial population of the FHCCRs by appending classes randomly to the generality and specificity parts of the pre-discovered FCCRs. The number of possible combinations in the generality and specificity parts increases with the rise in the number of classes present in the dataset. Hence, we have applied a genetic algorithm to fix the general and specific classes of FHCCRs in an optimal way.

Fitness Evaluation. Fitness function gives the quantitative measure to evaluate the quality of FHCCRs in the population. In the proposed approach, the degree of Subsumption and the similarity coefficient, already described in section of measuring goodness of hierarchies are used to evaluate the fitness of an individual. The fitness function is given below:

If (Generality = empty and Specificity= empty) then

$$Fitness = \frac{2 \times precision \times recall}{precision + recall}$$

Else

$$Fitness1 = \frac{2 \times precision \times recall}{precision + recall}$$ // Fitness of the first block of FHCCR

$Fitness2 = precision \times recall$ // Fitness of the second block of FHCCR

// Fitness of generality and specificity parts

$$Fitness3 = Subsumption\left(C_g, C_k\right) \times Similarity\left(C_g, C_k\right) + \sum_{i=1}^{3} Subsumption(C_k, C_{si} \times Similarity(C_k, C_{si})$$

$$Fitness = Fitness1 + Fitness2 + Fitness3$$

We have considered a maximum of three classes in the specificity part. The subsumption and similarity matrices are computed in advance from the set of FCCRs in the initial population to save the overhead of computation of these two measures again and again for each FHCCR generated during the evolutionary process. Two FHCCRs that can be generated from the FCCRs given in the initial population are shown in the Figure 5. The fitness computations are also depicted in this Figure 5.

A1	A2	A3	A4	A5	Class	A6	A7	Censor Class	Generality	Specificity
10	010	0010	10	00000	2	00010	0100	7	1	4 5 0
Precision=0.36 Recall=0.66 Fitness1=0.465						Censor_precision=0.09 Censor_recall=1 Fitness2=0.09			Fitness3=1.9455 Total Fitness= 2.5005	
A1	A2	A3	A4	A5	Class	A6	A7	Censor Class	Generality	Specificity
10	010	0010	01	00000	3	00001	1000	8	1	6 0 0
Precision= 0.5 Recall=1 Fitness1=0.667						Censor_precision=0.083 Censor_recall=1 Fitness2=0.083			Fitness3=1.4613 Total Fitness=2.2113	

Figure 5. FHCCRs generated from intial FCCR and their fitness evaluation

Genetic Operators. Tournament selection is used as the selection operator. The selection is restricted to Intra-species (individuals of same class) individuals since inter-species (individuals of different classes) individuals create low fitness offspring after applying crossover. New offspring are created from the selected ones by employing one point crossover and flip mutation. Cross breeding is done on the generality and specificity parts of the individuals of the same species. The mutation operator generates a new offspring by mutating generality and specificity blocks of an individual. The algorithm uses pre-selection technique to maintain diversity, i.e., the better fit offspring replace their parents. Figure 6 shows the overall genetic algorithm approach for discovering FHCCRs.

Post-processing Scheme

The average number of defining properties in a rule is given by following Equation 17:

$$ADP = \frac{1}{N}\sum_{i=1}^{N} |prop(R_i)| \qquad [17]$$

Where N is the number of rules and *prop* (R_i) is the number of defining properties for the i^{th} rule.

Here, we are presenting a post processing scheme in which small and related hierarchies can be merged together to form bigger hierarchies in order to reduce redundancy and increase interpretability. The scheme is as follows:

1. If an FHCCR is already a part of a bigger hierarchy, then the bigger FHCCR is retained and the smaller one is dropped.

Algorithm: Discovering Fuzzy Hierarchical Censored Classification Rules (FHCCRs)
Input: Set of FCCRs, Population Size = 500; Mutation Rate = 0.1; Crossover Rate = 0
Output: A set of FHCCRs
Begin
1. Evolve FCRs from data using Algorithms 'Evolving FCR' and 'Crowding'..
2. Discover FCCRs from the pre-discovered set of FCRs.
3. Construct Subsumption and Similarity Matrices from the pre-discovered set of FCCRs
4. Initialize random population P_0 of FHCCRs using set of FCCRs and generate random classes for generality and specificity parts
5. Evaluate Fitness of the population P_0
6. **While** stopping criteria not satisfied

Begin
6.1 Select two individuals i_1 and i_2 of same species
6.2 Apply crossover and mutation to produce new individuals i'_1 and i'_2//keeping the rule and censor parts fixed
6.3 Compute fitness of i'_1 and i'_2
6.4 **If** fitness of i'_1 > fitness of i_1
6.4.1 Replace i_1 in P_i with i'_1
End if
6.5 **If** fitness of i'_2 > fitness of i_2
Replace i_2 in P_i with i'_2
End if
End while
End.

Figure 6. The Genetic Algorithm to discover *FHCCRs*

2. If two FHCCRs have a common general class, but different specific classes, then these two FHCCRs are merged into a single hierarchy.
3. Two different hierarchies will be merged only and only if the threshold criteria for the degree of subsumption and coefficients of similarity are satisfied. These threshold values are kept at 0.6 for our experimentation.

After post processing step the related hierarchies are merged together and the redundant rules have been subsumed by the general rules.

The post processing scheme reduces the average number of defining properties significantly and hence the knowledge represented by the FHCCR tree is more compact and interpretable.

EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

The proposed approach was validated on five datasets. The Land-Transport dataset was taken from Bala and Ratnoo (2018), and it was further extended to include exceptions as well. Rests of the four datasets were from UCI machine learning repository. Table 2 shows the description of the datasets with respect to the number of attributes, number of classes and total instances. All the datasets had continuous attributes without any missing values.

Table 2
Description of datasets

Dataset	Attributes	Classes	Instances
Land-Transport	7	11	151
Hglass	9	6	214
E.coli	7	8	336
Yeast	8	10	1484
Vehicle	18	4	846

The datasets were normalized and fuzzified as described in the pre-processing step. Fuzzification of attributes had been carried out using the algorithm proposed in Bala & Ratnoo (2016). The resulting fuzzy partitions obtained for the attributes of different datasets are given in Table 3.

After fuzzifying the attributes, FHCCRs are discovered in three phases as described below:

1. In the first Phase, FCRs are discovered by using the algorithms *'Evolving_FCRs'* and *'Crowding'* proposed in (Bala & Ratnoo, 2016) as mentioned earlier. These FCRs occupy the first part of FHCCRs representation. We have compared the accuracy and number of rules discovered by the classifiers containing FCRs with those obtained from Decision table, JRIP, PART and J48 classifiers available in WEKA. (Witten et al., 2011). These classifiers are chosen since they produce rules in "*If-Then*' form which are comparable with the rules discovered in the form of FCRs. All these classifiers are applied with their default settings provided in WEKA. The results have been taken with tenfold cross validation sampling method across all the datasets and classifiers. Table 4 compares the results averaged over ten folds.

   The table shows that FCRs perform significantly better than the other classifiers on four datasets (*Land-Transport*, Hglass, E-coli and Yeast), but it fails to do better on one dataset (Vehicle). For Vehicle dataset, the PART algorithm obtains the highest accuracy, however, it discovers many rules (29) as compared to the number of rules (7) discovered by the classifier consisting of FCRs. This reflects the tarde off between accuracy and number of rules discovered. Overall, JRip and the classifier containing FCRs discover lesser number of rules. As RBs with less number of rules are considered more interpretable (García et al., 2008), JRip and FCRs perform significantly better on this metric.
2. In the second phase of the design, FCRs have been converted to FCCRs based on algorithm given in (Saroj & Bharadwaj, 2007). The algorithm is modified for fuzzy rules. The exceptions/censors discovered in this phase are accommodated in the second part of the FHCCRs representation. We have compared the accuracy of

Table 3
Attribute-wise Fuzzy partitions for datasets

Attributes	FMF	Linguistic terms	Attributes	FMF	Linguistic terms
Dataset: Land_Transport					
1. Weight	△	Low, Medium, High	5. Boot-Space	⏢	Low, Low-medium, Medium-high, High
2. Mileage	⏢	Low, Medium, High	5. Fuel-capacity	△	Low, Medium, High
3. Width	△	Low, Medium, High	6. Power	△	Low, low-medium, Medium-high, High
4. Length	△	Low, Medium, High			
Dataset: Hglass					
1. Refractive Index (RI)	⏢	Low, High	6. Potassium (K)	△	Low, High
2. Sodium(Na)	△	Low, High	7. Calcium (Ca)	⏢	Low, Low-medium, Medium-high, High
3. Magnesium (Mg)	⏢	Low, High	8. Barium (Ba)	⏢	Low, Low-medium, Medium-high, High
4. Aluminium (Al)	△	Low, High	9. Iron (Fe)	△	Low, High
5. Silicon (Si)	△	Low, Medium, High			
Dataset:E-coli					
1. mcq	⏢	Low, High	5. Vac	⏢	Low, Medium, High
2. Gvh	△	Low, High	6. Alm1	⏢	Low, High
3. Lip	△	Low, High	7. Alm2	⏢	Low, High
4. Cb	△	Low, High			
Dataset: Yeast					
1. Mcq	⏢	Low, High	5. Eri	△	Low, High
2. Gvh	⏢	Low, High	6. Pox	⏢	Low, High
3. Alm	△	Low, Medium, High	7. Vac	△	Low, Low-medium, Medium-high, High
4. Mit	△	Low, High	8. Mv	⏢	Low, Low-medium, Medium-high, High
Dataset: Vehicle					
1. Radius Ratio	⏢	Low, Medium, High	7.Scaled Variance along major axis	△	Low, High
2. PR Axis aspect Ratio	△	Low, High	8. Scaled Variance along minor axis	△	Low, Medium, High
3. Max length Aspect Ratio	△	Low, Low-medium, Medium, Medium-high, High	9. Skewness about major axis	⏢	Low, Medium, High
4. Scatter Ratio	⏢	Low, Low-medium, Medium-high, High	10. Skewness about minor axis	⏢	Low, High
5. Elongatedness	⏢	Low, High	11. Kurtosis about minor axis	△	Low, High
6. PR Axis Rectangularity	⏢	Low, Medium, High			

FCRs and FCCRs (Fuzzy Classification Rules with Censors/Exceptions) in Table 5. The results illustrate that the accuracy slightly increases when exceptions/censors get appended to the default rules. This is so because exception/censor conditions decrease false positive (FP) rate and increase True Negative (TN) rate. The results in Table 5 show that exceptions have been discovered for all the datasets except E-coli. Discovering exceptions may not increase accuracy to a great extent, but their discovery is interesting because knowing exceptions gives us a chance to revise our decisions in those rare circumstances where the default rules become inapplicable.

3. FCCRs are converted into FHCCRs in the third phase by appending the respective general and specific information to the FCCRs. The Initial population of FHCCRs is generated by appending general and specific classes randomly among the FCCRs. The fitness of individuals in the population is calculated from the subsumption and similarity matrices. The crossover and mutation operators are applied only to the generality and specificity parts of the initial FHCCRs. Table 6 gives the GA parameters employed to discover FHCCRs.

The additional stopping criteria adopted was of no change in the fitness of individuals for last 10 generations. The FHCCRs discovered for the '*Land transport*' dataset along with their fitness values are shown in Table 7.

Table 4
Comparisonon the basis of accuracy and number of rules

Algorithm/ Dataset	Decision table		JRip		PART		J48		Classifier (FCR)	
	Acc.	No of Rules	Acc.	No of Rules	Acc.	No of Rules	Acc.	No of Rules	Acc.	No of Rules
Land-Transport	39.07	15	43.7	**8**	49.67	17	54.30	25	**67.62**	**13.2**
Hglass	68.22	38	69.62	**8**	68.22	14	66.82	30	**83.59**	**7.5**
E.coli	75.29	18	80.35	**8**	79.46	13	84.22	22	**87.59**	**8.1**
Yeast	53.77	109	57.82	**15**	53.5	132	55.86	185	**72.57**	**10**
Vehicle	61.82	62	63.71	**16**	**68.68**	29	68.20	61	60.58	**7.1**

Table 5
Comparison of accuracy of FCCRs with accuracy of FCRs

Dataset	Accuracy without Exceptions	Accuracy with Exceptions	No of Exceptions
Land-Transport	67.62	**71.63**	2
Hglass	83.59	**84.25**	1
E.coli	87.59	**87.59**	0
Yeast	72.57	**73.45**	1
Vehicle	60.58	**60.70**	1

In this table, the first column lists FHCCRs, the second column gives the fitness of the FCRs, the third column denotes the fitness of censor part and the fourth column gives the fitness of FHCRs. Finally, the fifth column provides the fitness of entire FHCCRs. The proposed algorithm has discovered 11 FHCCRs. The set of FHCCRs discovered are shown pictorially in Figure 7.

Table 6
GA parameters used for discovery of FHCCRs

Population Size	20*no. of FCCRs discovered
Crossover rate	0.6
Mutation rate	0.1
Maximum number of generations	50

Table 7
FHCCRs discovered for 'Land-transport' dataset

FHCCRs	*Fitness1*	*Fitness2*	*Fitness3*	*Total Fitness*
If (Weight is low) $\wedge$ (Width is low) $\wedge$ (Length is low) $\wedge$ (Boot_space is low) $\neg$ (Fuel_capacity is low) *Then* Two-wheeler $\neg$ (Power is medium-high) : (Sports-bike); *Generality* [] *Specificity* [Scooter, Bike]	0.5	0.085	1.592	2.177
If (Weight is medium) $\wedge$ (Mileage is low) $\neg$ (Width is medium) *Then* Car $\neg$ (Length is medium $\vee$ power is high) : (Sports-car); *Generality*[] *Specificity* [Small-car, Big-car]	0.376	0.037	1.541	1.954
If (Weight is low) $\wedge$ (Mileage is medium) $\wedge$ (Width is low) $\wedge$ (Length is low) $\wedge$ (Boot_space is low) $\wedge$ (Fuel_capacity is low) $\wedge$ (Power is low) *Then* Scooter; *Generality* [Two-wheeler] *Specificity*[]	0.733	0	0.796	1.530
If (Weight is low) $\wedge$ (Mileage is high) $\wedge$ (Width is low) $\wedge$ (Length is low) $\wedge$ (Boot_space is low) $\wedge$ (Fuel_capacity is low) $\wedge$ (Power is low) *Then* Bike; *Generality* [Two-wheeler] *Specificity* []	0.8	0	0.796	1.596
If (Weight is medium) $\wedge$ (Mileage is low) $\wedge$ (Width is medium) $\neg$ (Length is medium) *Then* Small-car $\neg$ (Power is high) : (Sports-car); *Generality* [Car] *Specificity* [Hatchback]	0.52	0.108	2.497	3.125
If (Weight is medium) $\wedge$ (Mileage is low) $\wedge$ (Width is medium) $\wedge$ (Length is high) $\wedge$ (Power is low-medium)*Then* Big-car; *Generality* [Car] *Specificity* [Sedan, SUV]	0.455	0	1.575	2.031
If (Weight is medium) $\wedge$ (Mileage is low) $\wedge$ (Width is medium) $\wedge$ (Length is medium) $\wedge$ (Fuel_capacity is low-medium) *Then* Hatchback; *Generality*[Small-car] *Specificity*[]	0.733	0	0.820	1.553

Table 7 *(continue)*

FHCCRs	*Fitness1*	*Fitness2*	*Fitness3*	*Total Fitness*
If (Weight is medium) ∧ (Mileage is low) ∧ (Width is medium) ∧ (Length is high) ∧ (Boot_space is medium-high) ∧ (Power is low-medium) *Then* Sedan; *Generality* [Big-car] *Specificity*[]	0.722	0	0.887	1.609
If (Weight is medium) ∧ (Mileage is low) ∧ (Width is medium) ∧ (Length is high) ∧ (Boot_space is Low-medium) ∧ (Power is low-medium) *Then* SUV *Generality* [Big-car] *Specificity* []	0.571	0	0.837	1.408
If (weight is low) ∧ (Mileage is low) ∧ (Width is low) ∧ (Length is low) ∧ (Boot-space is low) ∧ (Fuel-capacity is low) ∧ (Power is medium-high) *Then* Sports-bike; *Generlity* [Two-wheeler] *Specificity* []	0.667	0	0.796	1.463
If (weight is medium) ∧ (Mileage is low) ∧ (Width is medium) ∧ (Length is medium) ∧ (Power is high) *Then* Sports-car; *Generality*[Car] *Specificity* []	0.5	0	0.870	1.370

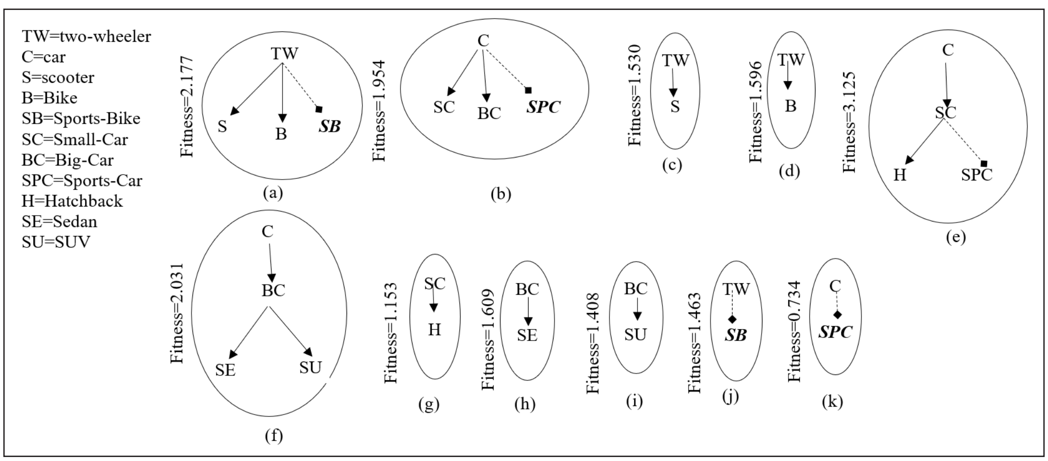

Figure 7. FHCCRs discovered for *Land-Transport* dataset

The post-processing step consolidates these FHCCRs into bigger hierarchies to further reduce the redundancy as described in post-processing scheme. Figure 8 shows the complete hierarchy for '*Land-Transport*' dataset along with the defining properties. The bold dashed circle at the top level of the hierarchy represents the most abstract class which can be added by human intervention.

Table 8 gives the number of FHCCRs discovered, average number of defining properties before and after post-processing, size of hierarchy and number of exceptions discovered.

The fifth column of the table shows the size of hierarchies along with the number of classes encountered at each level. For '*Land-transport*' dataset, two classes have been encountered at first level, five classes at second level and four classes at third level. The last column shows the number of censors discovered at each level. The results show that merging of FHCCRs into bigger hierarchies reduces the average number of defining properties per rule significantly and hence reduces the redundancy in rule representation making them more interpretable. This reduction depends upon the size of hierarchies. Bigger a hierarchy is, more is the reduction in the average number of defining properties.

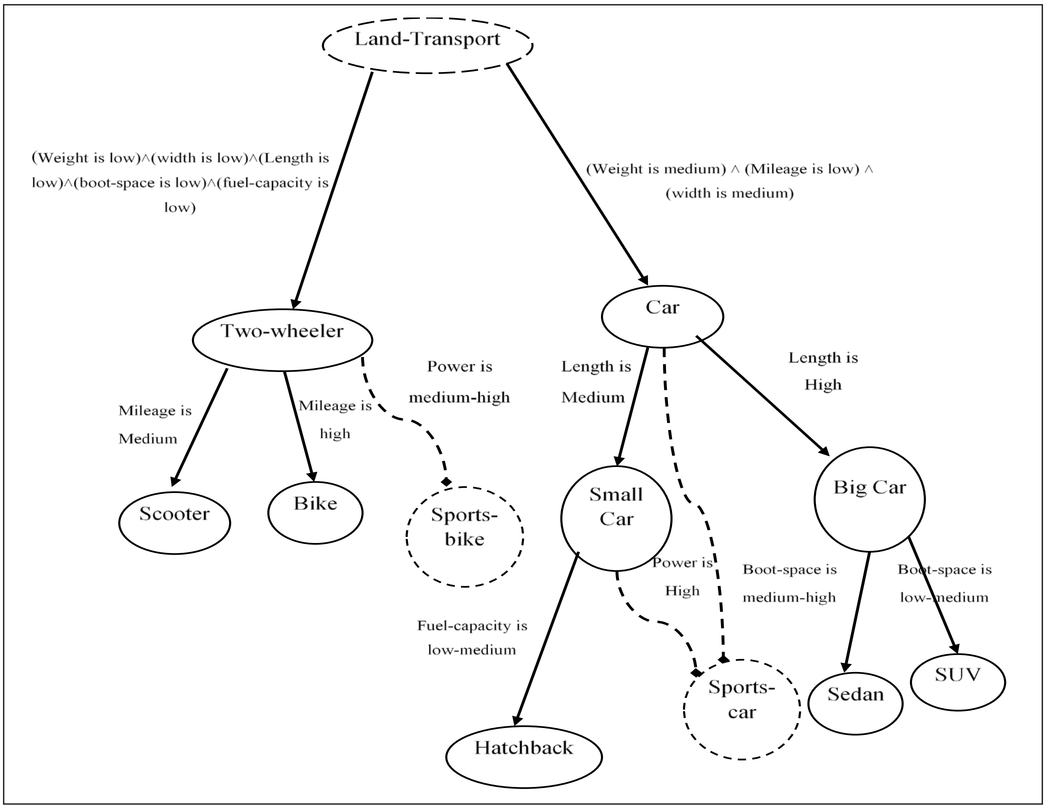

Figure 8. Complete hierarchical structure among classes of *Land-transport* dataset

Table 8
Properties of FHCCRs for different datasets

Dataset	No. of FHCCRs	Avg. Defining properties before Post-processing	Avg. Defining properties after Post-processing	Size of Hierarchy	No of censors/ Exceptions
Land-transport	11	5.8	**1.8**	3: (2, 5, 4)	0, 2, 1
Hglass	6	7.6	**4.6**	2: (3, 3)	1, 0
Ecoli	8	5.8	**5.2**	2: (7, 1)	0, 0, 0
Yeast	10	6.8	**3.9**	2: (5, 5)	2, 1
Vehicle	4	6.0	**6.0**	0	0

The proposed algorithm discovers concise, complete and interpretable rules in a hierarchical form for FRBs. It can classify objects at different levels of specificity. It also contains exceptions to the rules which enhance confidence in making decisions.

CONCLUSION AND FUTURE SCOPE

In this paper, we have proposed discovering Fuzzy Hierarchical Censored Classification Rules (FHCCRs) through genetic algorithm approach. The suggested approach has discovered FHCCRs in three phases. The first phase discovers FCRs which are, subsequently, converted into FCCRs and FHCCRs in the second and third phases respectively. The suggested approach has successfully captured the exceptions and hierarchical relationship in the class labels in the form of FHCCRs. Two measures- degree of subsumption and coefficient of similarity- have been used in the fitness function to measure the goodness of hierarchical relations. We have merged the individual FHCCRs to form larger hierarchies in a post processing step. This step has further organized the discovered knowledge in a more concise and non-redundant form because common properties among rules do not need to be repetitively represented and the specific classes inherit properties of the general classes.

Discovery of FHCCR trees is a contribution to form Fuzzy Rule Base Systems (FRBS) that are more accurate, interpretable and interesting. In addition, such FRBSs support variable certainty and specificity while drawing inference, i.e., these RBs are capable of taking corrective measures in the presence of censor/exceptional conditions, and classifying knowledge at multiple levels of abstraction. Since, FHCCRs have the capabilities to reason with incomplete and uncertain premises, to classify examples/objects at different levels of specificity and to take appropriate actions in exceptional circumstances; these have applications in the domains like robotics where human like adaptive decision making is required. This work can as well be extended to make multi-label classifications in the fields such as Bioinformatics and disease diagnosis where knowledge is commonly organized in a hierarchal manner.

ACKNOWLEDGEMENT

We sincerely thank Chairperson of the Department of the Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, who provided us an opportunity to join the department as JRFscholar, and who gave us access to the laboratory and research facilities. Our thanks also go to our friends and colleagues from the Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology. Finally, We would like to gratefully acknowledge the financial support received from The UGC.

REFERENCES

Al-Maqaleh, B. M., & Bharadwaj, K. K. (2005). Genetic programming approach to hierarchical production rule discovery. *Transactions on Engineering, Computing and Technology, 6*, 271-274.

Bala, R. (2012). Discovering fuzzy censored classification rules (fccrs): A genetic algorithm approach. *International Journal of Artificial Intelligence and Applications, 3*(4), 175-188.

Bala, R., & Ratnoo, S. (2016). A genetic algorithm approach for discovering tuned fuzzy classification rules with intra- and inter-class exceptions. *Journal of Intelligent Systems, 25*(2), 263-282.

Bala, R., & Ratnoo, S. (2018). Discovery of fuzzy hierarchical classification rules (FHCRs): A genetic algorithm approach. In N. R. Shetty, L. M. Patnaik, N. H. Prasad, & N. Nalini (Eds.), *Emerging research in computing, information, communication and applications* (pp. 595-604). Basel, Swetzerland: Springer Nature.

Bharadwaj, K. K., & Al-Maqaleh, B. M. (2007). Evolutionary approach for automated discovery of censored production rules. *International Journal of Computer, Electrical, Automation, Control and Information Engineering, 10*(1), 147-152.

Bharadwaj, K. K., & Al-Maqaleh, B. M. (2007, March 21-23). Evolutionary approach for automated discovery of censored production rules with fuzzy hierarchy. In *International Multi-conference of Engineers and Computer Scientists* (pp. 716-721). Hong Kong.

Bharadwaj, K. K., & Jain, N. K. (1992). Hierarchical censored production rules (HCPRs) system. *Data and Knowledge Engineering, 8*(1), 19-34.

Bharadwaj, K. K., & Kandelwal, R. (2007). Cumulative learning based on dynamic clustering of hierarchical production rules (HPRs). *International Journal of Computer and Information Engineering, 1*(8), 2615-2620.

Bharadwaj, K. K., & Saroj, S. (2009). Parallel genetic algorithm approach to automated discovery of hierarchical production rules. In J. Mehnen, M. Köppen, A. Saad, & A. Tiwari (Eds.), *Applications of Soft Computing* (pp. 327-336). Heidelberg, Germany: Springer.

Bharadwaj, K. K., & Saroj, S. (2010). A parallel genetic programming based intelligent miner for discovery of censored production rules with fuzzy hierarchy. *Expert Systems with Applications, 37*(6), 4601-4610.

Cerri, R., Barros, R. C., & de Carvalho, A. C. P. L. F. (2012, March 26-30). A genetic algorithm for hierarchical multi-label classification. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 250-255). Trento, Italy.

Cerri, R., Barros, R. C., de Carvalho, A. C., & Freitas, A. A. (2013, June 20-23). A grammatical evolution algorithm for generation of hierarchical multi-label classification rules. In *2013 IEEE Congress on Evolutionary Computation* (pp. 454-461). Cancun, Mexico.

Cordón, 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, 25*(3), 187-215.

Davies, M. N., Secker, A., Freitas, A. A., Mendao, M., Timmis, J., & Flower, D. R. (2007). On the hierarchical classification of G protein-coupled receptors. *Bioinformatics, 23*(23), 3113-3118.

Fernández, A., del Jesus, M. J., & Herrera, F. (2009). Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *International Journal of Approximate Reasoning, 50*(3), 561-577.

Freitas, A. A. (2002a). *Data mining and knowledge discovery with evolutionary algorithms*. Heidelberg, Germany: Springer-Verlag.

Freitas, A. A. (2002b). Data mining tasks and concepts. In A. A. Freitas (Ed.), *Data mining and knowledge discovery with evolutionary algorithms* (pp. 13-43). Heidelberg, Germany: Springer.

Freitas, A. A. (2008). A Review of evolutionary algorithms for data mining. In O. Maimon & L. Rokach (Eds.), *Soft Computing for Knowledge Discovery and Data Mining* (pp. 79-111). Boston, MA: Springer.

Gacto, M. J., Alcalá, R., & Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: an overview of interpretability measures. *Information Sciences, 181*(20), 4340-4360.

García, S., Fernández, A., Luengo, J., & Herrera, F. (2008). A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing, 13*(10), 959-977.

Herrera, F. (2008). Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence, 1*(1), 27-46.

Ishibuchi, H., Murata, T., & Türkşen, I. B. (1997). Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems, 89*(2), 135-150.

Ishibuchi, H., Nakashima, T., & Murata, T. (2001). Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences, 136*(1-4), 109-133.

Ishibuchi, H., Nakashima, T., & Nii, M. (2005b). *Classification and modeling with linguistic information granules: Advanced approaches to linguistic data mining*. Heidelberg, Germany: Springer-Verlag.

Ishibuchi, H., Nakashima, Y., & Nojima, Y. (2011). Performance evaluation of evolutionary multi-objective optimization algorithms for multi-objective fuzzy genetics-based machine learning. *Soft Computing, 15*(12), 2415-2434.

Ishibuchi, H., Nozaki, K., Yamamoto, N., & Tanaka, H. (1995). Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems, 3*(3), 260-270.

Ishibuchi, H., Yamamoto, T., & Nakashima, T. (2005a). Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35*(2), 359-365.

Jain, N. K., & Bharadwaj, K. K. (1998). Some learning techniques in hierarchical censored production rules (HCPRs) system. *International Journal of Intelligent Systems, 13*(4), 319-344.

Kontonasios, K. N., Spyropoulou, E., & De Bie, T. (2012). Knowledge discovery interestingness measures based on unexpectedness. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2*(5), 386-399.

Kuncheva, L. I. (2010). *Fuzzy Classifier Design* (1st Ed.). Heidelberg, Germany: Physica-Verlag HD.

Mendes, R. R. F., Voznika, F. B., de Freitas, A. A., & Nievola, J. C. (2001). Discovering fuzzy classification rules with genetic programming and co-evolution. In L. D. Raedt & A. Siebes (Eds.), *Principles of Data Mining and Knowledge Discovery* (pp. 314-325). Heidelberg, Germany: Springer.

Michalski, R. S., & Winston, P. H. (1986). Variable precision logic. *Artificial Intelligence, 29*(2), 121-146.

Neerja, & Bharadwaj, K. K. (1996). Calculus of fuzzy hierarchical censored production rules (FHCPRs). *International Journal of Intelligent Systems, 11*(1), 1-25.

Palm, R., Driankov, D., & Hellendoorn, H. (1997). *Model based fuzzy control: Fuzzy gain schedulers and sliding mode fuzzy controllers*. Heidelberg, Germany: Springer.

Pedrycz, W. (Ed.). (1996). *Fuzzy modelling* (Vol. 7). Boston, MA: Springer US.

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research, 7*, 1601-1626.

Sanz, J. A., Fernández, A., Bustince, H., & Herrera, F. (2010). Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. *Information Sciences, 180*(19), 3674-3685.

Saroj, S., & Bharadwaj, K. K. (2007, February 12-14). 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). Innsbruck, Austria.

Secker, A. D., Davies, M. N., Freitas, A. A., Timmis, J., Mendao, M., & Flower, D. R. (2007). An experimental comparison of classification algorithms for hierarchical prediction of protein function. *Expert Update (Magazine of the British Computer Society's Specialist Group on AI), 9*(3), 17-22.

Secker, A., Davies, M. N., Freitas, A. A., Clark, E. B., Timmis, J., & Flower, D. R. (2010). Hierarchical classification of G-protein-coupled receptors with data-driven selection of attributes and classifiers. *International Journal of Data Mining and Bioinformatics, 4*(2), 191-210.

Silberschatz, A., & Tuzhilin, A. (1995, August 20-21). On subjective measures of interestingness in knowledge discovery. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 275-281). Montréal, Québec, Canada.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 970-974.

Sun, A., & Lim, E. P. (2001, November 29-December 2). Hierarchical text classification and evaluation. In *Proceedings of IEEE International Conference on Data Mining, ICDM 2001* (pp. 521-528). San Jose, CA, USA.

Sun, A., Lim, E. P., Ng, W. K., & Srivastava, J. (2004). Blocking reduction strategies in hierarchical text classification. *IEEE Transactions on Knowledge and Data Engineering, 16*(10), 1305-1308.

Suzuki, E. (2002). Undirected discovery of interesting exception rules. *International Journal of Pattern Recognition and Artificial Intelligence, 16*(08), 1065-1086.

Suzuki, E. (2004, September 20-24). Discovering interesting exception rules with rule pair. In *Proceedings of the ECML/PKDD Workshop on Advances in Inductive Rule Learning* (pp. 163-178). Pisa, Italy.

Suzuki, E., & Shimura, M. (1996, August 2-4). Exceptional knowledge discovery in databases based on information theory. In *Proceeding KDD'96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 275-278). Portland, Oregon.

Suzuki, E., & Zytkow, J. M. (2000). Unified algorithm for undirected discovery of exception rules. In D. A. Zighed, J. Komorowski & J. Zytkow (Eds.), *Principles of data mining and knowledge discovery* (pp. 169-180). Heidelberg, Germany: Springer.

Tsumoto, S. (2003). Automated extraction of hierarchical decision rules from clinical databases using rough set model. *Expert Systems with Applications, 24*(2), 189-197.

Vashishtha, J., Kumar, D., & Ratnoo, S. (2013). An evolutionary approach to discover intra- and inter-class exceptions in databases. *International Journal of Intelligent Systems Technologies and Applications, 12*(3/4), 283-300.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: practical machine learning tools and techniques* (3rd Ed.). Burlington, MA: Morgan Kaufmann.

Zadeh, L. A. (1996). Fuzzy logic computing with words. *IEEE Transactions on Fuzzy Systems, 4*(2), 103-111.