World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:10, 2007

Evolutionary Approach for Automated Discovery of Censored Production Rules

Kamal K. Bharadwaj, and Basheer M. Al-Maqaleh

Open Science Index, Computer and Information Engineering Vol:1, No:10, 2007 publications.waset.org/14169/pdf

***Abstract*—**In the recent past, there has been an increasing interest in applying evolutionary methods to Knowledge Discovery in Databases (KDD) and a number of successful applications of Genetic Algorithms (GA) and Genetic Programming (GP) to KDD have been demonstrated. The most predominant representation of the discovered knowledge is the standard Production Rules (PRs) in the form **If P Then D**. The PRs, however, are unable to handle exceptions and do not exhibit variable precision. The Censored Production Rules (CPRs), an extension of PRs, were proposed by Michalski & Winston that exhibit variable precision and supports an efficient mechanism for handling exceptions. A CPR is an augmented production rule of the form:
If P Then D Unless C, where **C** (Censor) is an exception to the rule.

Such rules are employed in situations, in which the conditional statement '**If P Then D**' holds frequently and the assertion **C** holds rarely. By using a rule of this type we are free to ignore the exception conditions, when the resources needed to establish its presence are tight or there is simply no information available as to whether it holds or not. Thus, the **'If P Then D'** part of the CPR expresses important information, while the **Unless C** part acts only as a switch and changes the polarity of **D** to **~D**.
This paper presents a classification algorithm based on evolutionary approach that discovers comprehensible rules with exceptions in the form of CPRs.

The proposed approach has flexible chromosome encoding, where each chromosome corresponds to a CPR. Appropriate genetic operators are suggested and a fitness function is proposed that incorporates the basic constraints on CPRs. Experimental results are presented to demonstrate the performance of the proposed algorithm.

***Keywords*—**Censored Production Rule, Data Mining, Machine Learning, Evolutionary Algorithms.

I. Introduction

OVER the last decade there has been an increasing amount of research in the field of automated learning and discovery, in general, and Knowledge Discovery in Databases (KDD), in particular. KDD can be defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [8].

Data Mining is the process of discovering interesting knowledge from large amounts of data [14], which can be done using different kinds of algorithms depending mainly on the application domain and the user interest.

Genetic Algorithms (GAs) are based on Darwinian natural selection and Mendelian genetics, in which each point in the search space is a string called a chromosome that represents a possible solution. This approach requires a population of chromosomes representing a combination of features from the set of features and requires a cost function that calculates each chromosome's Fitness (this function is called evaluation function or Fitness function). The algorithm performs optimization by manipulating a finite population of chromosomes. In each generation, the GA creates a set of new chromosomes by crossover, inversion and mutation, which correlate to processes in natural reproduction [2], [13].

There has been increasing interest in applying evolutionary computation methods as data mining tasks to KDD [10], [19] [29]. A number of successful applications of GA [2], [6], [7], [9], [26], [28] and, lately, Genetic Programming (GP) [3], [4], [5], [27] to KDD have been reported in the literature.

The most predominant representation of the discovered knowledge is the standard Production Rules (PRs) in the form **If P Then D**. The PRs, however, are unable to handle exceptions and do not exhibit variable precision [1]. Exceptions, which focus on a very small portion of a dataset, have been ignored or discarded as noise in machine learning, but the goal of KDD is broader and it is always interesting to discover exceptions, as they challenge the existing knowledge and often led to the growth of knowledge in new directions [25].

In the past, some efforts have been made toward considering rules with exceptions for knowledge acquisition/discovery. Ripple-Down Rule (RDR) with exceptions for incremental development of a Knowledge-Based System (KBS) is considered in [12]. Learning rules with local exceptions using RDR is discussed in [18] that allow us to deal with exceptions for each rule separately by introducing exception rules, exception rules for each exception rule, etc. up to a constant depth. In [11] Exception Directed Acyclic Graph (EDAG) as a knowledge structure is presented that subsumes trees and rules but can be substantially more compact. An intuitive representation of decision tree using general rules and exceptions that reduce the complexity of the discovered knowledge substantially is discussed in [20]. A unified algorithm is given in [25] for

Kamal K. Bharadwaj, is a professor at the School of Computer and Systems Sciences (SC&SS), Jawaharlal Nehru University (JNU), New Delhi, India.(e-mail: kbharadwaj@gmail.com).

Basheer Mohamad Al-Maqaleh is a Ph.D scholar at School of Computer and Systems Sciences (SC&SS), Jawaharlal Nehru University (JNU), New Delhi, India, on leave from Thamar University, Republic of Yemen. Mobile +91-9911071673; (e-mail: bmaa100@yahoo.com).

Open Science Index, Computer and Information Engineering Vol:1, No:10, 2007 publications.waset.org/14169/pdf

undirected discovery of exception rules. A method for mining exception rule is proposed by [16], which is based on a measure that estimates interestingness of discovered rules.

As an extension of PR, Michalski and Winston [21] have suggested Censored Production Rule (CPR) as an underlying representational and computational mechanism to enable logic based systems to exhibit variable precision, in which certainty varies, while specificity stays constant. A CPR has the form **If P Then D Unless C**, where **C** (censor) is the exception condition. Such rules are employed in situations, in which the conditional statement '**If P Then D**' holds frequently and the assertion **C** holds rarely. By using a rule of this type we are free to ignore the censor (exception) conditions, when the resources needed to establish its presence are tight or there is simply no information available as to whether it holds or does not hold. 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 For example, the following rule might be used to express the fact that I read the paper before going to work unless I oversleep, which occurs rarely [17]:

If Weekday-Morning **Then** Read-Paper **Unless** Oversleep.

A CPR may have more than one censor conditions, say, C_1, C_2, ... C_n and is denoted as:

If P Then D Unless ($C_1 \vee C_2 \vee \ldots \vee C_n$).

For example, "**If** Bird **Then** Fly **Unless** (Penguin $\vee$ broken-wings $\vee$ sick $\vee$ dead)".

In this paper, we have presented a classification algorithm based on evolutionary approach for the automated discovery of comprehensible rules with exception in the form of CPRs.

II. Censored Production Rules (CPRs)

Let us now give a more quantitative definition of a CPR:

$$P \rightarrow D \lfloor C, \tag{1}$$

where P is a premise, D is a decision and C is a censor. Although the unless operator, $\lfloor$, is logically equivalent to the commutative exclusive-or operator, the **Unless** operator has an expositive aspect, which is not commutative. In order to capture the asymmetry precisely, let us associate two parameters, γ_1 and γ_2, with rule (1)

$$P \rightarrow D \lfloor C: \gamma_1, \gamma_2. \tag{2}$$

Both γ_1 and γ_2 are point probabilities, one indicating the strength of the relationship between P and D and the other between P and C. Now, consider the following sets: Ω is a finite sample of events; Ω_P is the set of events, for which P holds; Ω_{PD} is the subset of events, for which both P and D hold; Ω_{PC} is a subset of events for, which both P and C hold [21].

Given these sets, the parameters γ_1 and γ_2 are defined as follows:

$$\gamma_1 = \frac{\Pr[P, D]}{\Pr[P]} = \Pr[D|P] \approx \frac{|\Omega_{PD}|}{|\Omega_P|},$$
$$\gamma_2 = \frac{\Pr[P, C]}{\Pr[P]} = \Pr[C|P] \approx \frac{|\Omega_{PC}|}{|\Omega_P|}, \tag{3}$$

where $|\Omega_i|$ denotes the cardinality of Ω_i. Also, we assume that $\Omega_D \cap \Omega_C = \varnothing$ and $\Omega_D \cup \Omega_C = \Omega_P$, thus Pr [P | D] + Pr [C | P] = 1. The main constraint on the CPR is

$$\gamma_1 >> \gamma_2 \tag{4}$$

To understand the implication of CPR, Michalaski and Winston [21] presented a quantitaive definition for it, where two parameters γ and δ have been introduced. A CPR is then written:

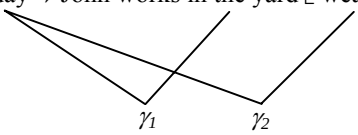
$$P \rightarrow D \lfloor (C_1 \vee UNK): \gamma, \delta. \tag{5}$$

The symbol UNK represents a disjunction of unknown conditions that could block the inference of D from P. Where γ = Pr [D | P], certainty of P→D when it is not known whether ($C_1 \vee$ UNK) holds. The implication P→D is certainty 1 when $C_1 \vee$ UNK is known to be false. Thus, the parameter γ is equivalent to the a priori certainty that none of the censors hold. When δ =Pr [D | P& ¬C_1], it is the certainty that P→D when ¬ C_1 is true. This is equivalent to the certainty that there is no implicit part of the censor that can hold when ¬ C_1 is true.

Obviously, the a priori certainty of ¬ ($C_1 \vee$ UNK) must be equal to or smaller than the a priori certainty that ¬UNK. Therefore, $\gamma <= \delta$. Note that δ = 1 if it is certain that there are no conditions in the censor other than C_1 [15].

For example, consider the CPR:

Sunday→ John works in the yard $\lfloor$ weather is bad (6)



The condition $\gamma_1 >> \gamma_2$ means the ratio of Sundays when John worked in the yard to all Sundays >> the ratio of Sundays with bad weather to all Sundays [21].

It is to be noted that PR is a special case of a CPR, when the **Unless** part is absent.

III. Applying GA to CPR Discovery

In this section an evolutionary approach is presented for the automated discovery of rules with exceptions using CPR as the underlying knowledge representation. The current version of the system handles only categorical attributes and also the proposed algorithm cannot cope with the missing values. That is why, in each dataset the few instances that contained missing values were simply removed.

Open Science Index, Computer and Information Engineering Vol:1, No:10, 2007 publications.waset.org/14169/pdf

A. Individual Representation

Suppose there are ***n*** predicting attributes and a single goal (class) attribute in the data being mined. A chromosome (sequence of genes) is divided into three parts: **If** part (***n*** genes corresponding to predicting attributes) consisting of a conjunction of conditions on the values of the predicting attributes, **Then** part (single goal attribute) representing decision and **Unless** part (***n*** genes corresponding to predicting attributes) consisting of a disjunction of conditions on the values of the predicting attributes representing censors. Accordingly, we have fixed length (***2n + 1)***, chromosome representation. It is to be noted that any of the ***n*** predicting attributes (one or more) can form conditions in the **If** part and similarly any of the ***n*** predicting attributes (zero or more) can form censor (exception) conditions in the **Unless** part. Further, for any rule the set of attributes forming **If** part and the set of attributes forming **Unless** part would be disjoint i.e., (set of attributes present in the **If** part) ∩ (set of attributes present in the **Unless** part)= Ø.

The ***n*** predicting attributes are represented as, Atr*1*, Atr*2*,….,Atr*i*,…,Atr*n*. Supposing that the *i*th attribute Atr*i* has *xi* values, the set of values corresponding to Atr*i* is denoted as Set-Val-Atr*i*= {Val-Atri_1, Val-Atri_2,……, Val-Atri_k ,…..,Val-Atri_{xi}}.

The condition (**If** part / **Unless** part) corresponding to the *i*th attribute Atri would be denoted as Atri= Val-Atri, where Val-Atri ∈ Set-Val-Atri.

The genes are positional i.e, the first gene represents the first attribute the second gene represents the second attribute and so on.

The structure of chromosome is shown in Fig.1.

If part	**Then** part	**Unless** part
n predicting attributes	single goal attribute	***n*** predicting attributes

Fig. 1 The structure of chromosome

The encoding corresponding to the *k*th value (Val-Atri$_k$) of Atr*i* is, a sequence of *xi* (number of values of Atri) bits such that Val-Atri$_k$ is set to 1 and the rest bits are all set to 0 as shown in Fig.2.

Val-Atri$_1$	Val-Atri$_2$	…	Val-Atri$_k$	…	Val-Atri$_{xi}$
0	0	…	1	…	0

Fig. 2 The encoding of predicting attribute Atr*i*

In case the *i*th attribute Atr*i* is absent in the conditional part, all the *xi* bits are set to zero.

A two-valued goal attribute can be encoded as single bit (1 representing class A and 0 representing class B).

Each gene corresponds to one condition in the **If** part or one censor condition in the **Unless** part of a rule. Thus, the entire chromosome (individual) corresponding to a single CPR, **If** conditions **Then** decision **Unless** censors, is encoded as shown in Fig.3.

Note that the above encoding is quite flexible with respect to the length of the rules. A "traditional" GA is very limited in this aspect, since it can only cope with fixed-length rules. In this proposed algorithm, although each chromosome has a fixed length, the genes are "interpreted" in such a way that the individual phenotype (the rule) has a variable length. Hence, different individuals correspond to rules with different number of conditions and different number of censor conditions. This kind of representation gives a lot of flexibility to the rules being discovered [9].

As an example, consider object classification training set (Table I) with three predicting attributes as shown below:

Attribute	Possible values
Bird	Yes
Kiwi	No, Yes
Dead	No, Yes

and there is one goal attribute:

Attribute	Possible values
Decision	Fly, ~Fly

Corresponding to the above training dataset, a CPR:

If Bird = Yes **Then** Decision = Fly **Unless** Kiwi = Yes would be encoded as:

Bird	Kiwi	Dead	Decision	Bird	Kiwi	Dead
Gene1	Gene2	Gene3	Class	Gene1	Gene2	Gene3
1	0 0	0 0	1	0 0	1 0	0 0

and a CPR:

If Bird = Yes **Then** Decision = Fly **Unless** (Kiwi = Yes ∨ Dead =Yes) would be encoded as:

Bird	Kiwi	Dead	Decision	Bird	Kiwi	Dead
Gene1	Gene2	Gene3	Class	Gene1	Gene2	Gene3
1	0 0	0 0	1	0 0	1 0	1 0

Notice that any attribute present will have exactly one bit set to1 and rest all bits are 0 in the corresponding gene and any attribute absent in the rule will have all the bits zero in the corresponding gene.

B. Genetic Operators

We used conventional genetic operators of selection and crossover. More precisely, we used fitness proportional selection and one-point crossover, with probability 0.7. We also used an elitist reproduction strategy, where the best individual of each generation was passed unaltered to the next generation.

Note that crossover points can fall only between genes and not inside a gene. Hence, crossover swaps entire rule conditions between individuals, but it cannot produce new rule conditions. The mutation operator accomplishes the creation of new rule conditions.

Mutation is an operator that acts on a single individual at a time. Mutation replaces the value of a gene with a randomly

Gene*1*	…	Gene *i*	…	Gene *n*	Class	Gene*1*	…	Gene *i*	…	Gene *n*
Val-Atr*1*	…	Val-Atr*i*	…	Val-Atr*n*	0/1	Val-Atr*1*	…	Val-Atr*i*	…	Val-Atr*n*

Fig. 3 Chromosome encoding

Open Science Index, Computer and Information Engineering Vol:1, No:10, 2007 publications.waset.org/14169/pdf

generated value, which belongs to the domain of the attribute. We developed three mutation operators tailored for our chromosome representation, namely *insert*, *swapping* and *dropping*. We used mutation rates of 0.1 for reach kind of mutation.

The *insert* mutation randomly chooses Val-Atr*i* in the **If** part or Val-Atr*j* in the **Unless** part and replaces it by a randomly chosen value from Set-Val-Atr*i* or Set-Val-Atr*j* in **If** and **Unless** part, respectively. While applying the mutation operator, *insert* to the **If** part or **Unless** part, it must be ensured that the mutated chromosome is legal i.e., the condition (set of attributes present in the **If** part) ∩ (set of attributes present in the **Unless** part) = Ø is satisfied. In case the above condition is not satisfied, the produced mutated chromosome is rejected as illegal and the operator *insert* is applied again to produce legal mutant.

The *swapping* mutation swaps randomly chosen Val-Atr*i* from the **If** part and Val-Atr*j* from the **Unless** part.

The *dropping* mutation randomly deletes (replace 1 by 0) a condition in the **If** part or a censor condition in the **Unless** part.

C. Fitness Function

The most difficult and most important concept of evolutionary algorithm is the Fitness function. It varies greatly from one type of problem to another. Clearly many criteria are used to quantify the quality or, Fitness, of a rule over the database. Some of these criteria are highly qualitative and in some cases subjective. However, in the context of genetic search we must formulate a single numerical quantity that encapsulates the desirable features [22].

The Fitness function to evaluate each individual CPR takes into account the logical interpretation of the rule and the basic constraint $\gamma_1 >> \gamma_2$.

Under the Dempster-Shafer interpretation of CPR, four belief values are associated with each CPR [23],[24]:

$$P \rightarrow D \lfloor C: \alpha, \beta, \gamma, \delta \quad (7)$$

such that

i. $P \wedge \neg C \rightarrow D$
ii. $P \wedge C \rightarrow \neg D$
iii. $P \rightarrow D$
iv. $P \rightarrow \neg D$

Four indicators α, β, γ and δ corresponding to rule (i), (ii), (iii) and (iv), respectively, are computed as shown below:

$$\alpha = \frac{|P \wedge \neg C \wedge D|}{|P \wedge \neg C|} \quad (8), \quad \beta = \frac{|P \wedge C \wedge \neg D|}{|P \wedge C|} \quad (9)$$

$$\gamma = \frac{|P \wedge D|}{|P|} \quad (10), \quad \delta = \frac{|P \wedge \neg D|}{|P|} \quad (11)$$

$$where \quad |p \wedge \neg c|, |p \wedge \dot{c}| \; and \; |P| \neq \dot{0}$$

And the basic constraint on CPR would be:

$$\gamma_1 = \frac{|P \wedge D|}{|P|} \quad >> \quad \gamma_2 = \frac{|P \wedge C|}{|P|} \quad (12)$$

where $\gamma_1 + \gamma_2 = 1$ (when all the censor conditions C are known).

From the equations (8),(9),(10),(11) and (12) we deduce that:

- If $\gamma_1 \geq 0.75$ – threshold defined by user- (i.e., $\gamma_1 >> \gamma_2$ is satisfied), then the discovered rule is considered as CPR.
- If $\gamma_1 < 0.75$ (i.e., $\gamma_1 >> \gamma_2$ is **not** satisfied), then the discovered rule cannot be designated as CPR. However, the discovered rule is a Production Rule with Exceptions (PRE). It is to be noted that these exception conditions are not censors and, therefore, cannot be ignored under resource constraints and must always be evaluated, while reasoning with such rules.

Finally, the Fitness function is defined as under:

$$\text{Fitness} = \begin{cases} \alpha \text{ if } \gamma 1 \geq 0.75 \text{ i.e., condition } \gamma_1 >> \gamma_2 \text{ is satsified } (CPR) \\ \beta \text{ if } \gamma 1 < 0.75 \text{ i.e., condition } \gamma_1 >> \gamma_2 \text{ is not satsfied (PRE)} \end{cases} \quad (13)$$

Also, the proposed algorithm would generate standard PRs, $P \rightarrow D$ & $P \rightarrow \neg D$ with the accuracy γ and δ, respectively, as Fitness.

Note that the Fitness function will produce values in the range [0..1] and the goal is to maximize the value.

As an illustration of the Fitness function, consider a CPR:

If Bird=Yes **Then** Decision = Fly **Unless** Kiwi=Yes

and the training dataset for object classification given in Table I.

TABLE I
OBJECT CLASSIFICATION TRAINING SET

No.	Bird	Kiwi	Dead	Decision
1	Yes	No	No	Fly
2	Yes	No	No	Fly
3	Yes	No	No	Fly
4	Yes	No	No	Fly
5	Yes	No	Yes	~ Fly
6	Yes	No	No	Fly
7	Yes	No	No	Fly
8	Yes	No	No	Fly
9	Yes	Yes	No	~ Fly

First of all the values of γ_1 and γ_2 are computed as under:

$$\gamma_1 = \gamma = \frac{|P \wedge D|}{|P|} = \frac{|Bird = Yes \wedge Decision = Fly|}{|Bird = Yes|} = \frac{|7|}{|9|} = 0.778$$

$$\gamma_2 = \frac{|P \wedge C|}{|P|} = \frac{|Bird = Yes \wedge Kiwi = Yes|}{|Bird = Yes|} = \frac{|1|}{|9|} = 0.111$$

Clearly, the constraint on the CPR, $\gamma_1 >> \gamma_2$ is satisfied (0.778 >> 0.111) and, therefore, the above CPR is valid.

Open Science Index, Computer and Information Engineering Vol:1, No:10, 2007 publications.waset.org/14169/pdf

Next, the Fitness (α) is computed as under:

$$Fitness = \alpha = \frac{|P \wedge \neg C \wedge D|}{|P \wedge \neg C|}$$
$$= \frac{|Bird = Yes \wedge Kiwi = No \wedge Decision = Fly|}{|Bird = Yes \wedge Kiwi = No|} = \frac{|7|}{|8|} = 0.875$$

IV. Experimental Results

Each Evolutionary Algorithm run consisted of a population of 40 individuals evolving during 50 generations. The proposed algorithm was terminated when the best Fitness did not change continually throughout 10 generations. The performance of the proposed algorithm on different datasets is demonstrated below:

Example 1: Let us consider the training dataset for object classification given in Table II.

TABLE II
OBJECT CLASSIFICATION TRAINING SET (BIRD)

No.	Bird	Kiwi	Dead	Broken-Wings	Decision
1	Yes	No	No	No	Fly
2	Yes	No	No	No	Fly
3	Yes	No	No	No	Fly
4	Yes	No	No	No	Fly
5	Yes	No	No	No	Fly
6	Yes	No	No	No	Fly
7	Yes	No	No	No	Fly
8	Yes	No	No	No	Fly
9	Yes	No	No	No	Fly
10	Yes	No	No	No	Fly
11	Yes	No	No	No	Fly
12	Yes	No	No	No	Fly
13	Yes	No	No	No	Fly
14	Yes	No	No	No	Fly
15	Yes	No	No	No	Fly
16	Yes	No	No	No	Fly
17	Yes	No	No	No	Fly
18	Yes	No	No	No	Fly
19	Yes	No	No	No	Fly
20	Yes	No	No	No	Fly
21	Yes	No	No	No	Fly
22	Yes	No	No	No	Fly
23	Yes	No	No	No	Fly
24	Yes	No	No	No	Fly
25	Yes	No	No	No	Fly
26	Yes	No	No	No	Fly
27	Yes	No	No	No	Fly
28	Yes	No	No	No	Fly
29	Yes	No	No	No	Fly
30	Yes	No	No	No	Fly
31	Yes	Yes	No	No	~Fly
32	Yes	No	No	Yes	~Fly
33	Yes	No	Yes	No	~Fly
34	Yes	Yes	Yes	No	~Fly
35	Yes	Yes	No	Yes	~Fly
36	Yes	No	Yes	Yes	~Fly
37	Yes	Yes	Yes	Yes	~Fly

The proposed algorithm discovered four rules shown in Table III.

Notice that the first three rules (Rule1 & Rule2 & Rule3) satisfy the constraint $\gamma_1 >> \gamma_2$ and are designated as CPRs. But the Rule4 does not satisfy the condition $\gamma_1 >> \gamma_2$ and, therefore, cannot be designated as CPR. However, this rule can be considered as PRE.

Example 2: This experiment was carried out on the Monk-1 dataset [30]. This dataset has 432 examples, 6 predicting attributes and a goal attribute, which can take on 2 classes. The predicting attributes were nominal.

In this case, six CPRs and three PRs are generated as in Table IV.

V. Conclusions and Future Work

In the present work, an evolutionary approach is proposed for the discovery of Censored Production Rules (CPRs) that can efficiently handle exceptions and deal with uncertain, incomplete and imprecise knowledge with resource constraints. The proposed scheme has flexible chromosome encoding and appropriate crossover and mutation operators are suggested. Suitable mutation operators are designed in the form of *inset, swapping* and *dropping*. Keeping in the view the basic constraints on CPRs, appropriate Fitness functions are formulated corresponding to different forms of the discovered rules i.e, PRs, CPRs and PREs.

Experimental results have demonstrated the effectiveness of the evolutionary scheme proposed in providing the user with compact and comprehensible classification rules in the form of PRs & CPRs.

In case of data being mined does not satisfy the constraints on the CPRs, the proposed algorithm would still discover PREs. However, in the later case the discovered rules would not support reasoning under time constraint and, therefore, all the exception conditions need to be evaluated during reasoning process.

The current version of the system handles only discrete attributes. Future enhancement will allow the use of continuous attributes. One of the most important future research directions would be the discovery of Hierarchical Censored Production Rules (HCPRs) [1] from large datasets using evolutionary algorithm.

TABLE III
RESULT FROM THE BIRD DATASET

No.	Discovered Rules	γ_1	γ_2	Fitness
1	**If** Bird =Yes **Then** Decision= Fly **Unless** Kiwi =Yes (CPR)	0.811	0.108	0.909
2	**If** Bird = Yes **Then** Decision= Fly **Unless** (Kiwi =Yes ∨ Dead =Yes) (CPR)	0.811	0.162	0.968
3	**If** Bird = Yes **Then** Decision= Fly **Unless** (Kiwi =Yes ∨ Dead =Yes ∨ Broken-Wings =Yes) (CPR)	0.811	0.189	1.000
4	**If** Bird = Yes **Then** Decision= ¬Fly **Unless** ¬(Kiwi =Yes ∨ Dead =Yes ∨ Broken-Wings =Yes) (PRE)	0.189	0.811	1.000

TABLE IV
RESULT FROM THE MONK-1 DATASET

No.	Discovered Rules	γ_1	γ_2	Fitness
1	**If** a1= 1 ∧ a2 = 2 **Then** class =0 **Unless** a5 = 1 (CPR)	0.750	0.250	1.000
2	**If** a1= 1 ∧ a2 = 3 **Then** class =0 **Unless** a5 = 1 (CPR)	0.750	0.250	1.000
3	**If** a1= 2 ∧ a2 = 1 **Then** class =0 **Unless** a5 = 1 (CPR)	0.750	0.250	1.000
4	**If** a1= 2 ∧ a2 = 3 **Then** class =0 **Unless** a5 = 1 (CPR)	0.750	0.250	1.000
5	**If** a1= 3 ∧ a2 = 1 **Then** class =0 **Unless** a5 = 1 (CPR)	0.750	0.250	1.000
6	**If** a1= 3 ∧ a2 = 2 **Then** class =0 **Unless** a5 = 1 (CPR)	0.750	0.250	1.000
7	**If** a1= 1 ∧ a2 = 1 **Then** class =1 (PR)	….	….	1.000
8	**If** a1= 2 ∧ a2 = 2 **Then** class =1 (PR)	…..	….	1.000
9	**If** a1= 3 ∧ a2 = 3 **Then** class =1 (PR)	….	….	1.000

Open Science Index, Computer and Information Engineering Vol:1, No:10, 2007 publications.waset.org/14169/pdf

REFERENCES

[1] K.K. Bharadwaj and N.K Jain, "Hierarchical censored production rules (HCPRs) system," *Data & Knowledge Engineering, North Holland*, vol. 8, pp. 19-34, 1992.

[2] K.K. Bharadwaj, N.M. Hewahi and M.A. Brando, "Adaptive hierarchical censored production rule-based system: A genetic algorithm approach," *Advances in Artificial Intelligence,* SBIA '96, Lecture Notes in Artificial Intelligence, No. 1159, Berlin, Germany, Springer-Verlag, pp. 81-90, 1996.

[3] Basheer M. Al-Maqaleh and Kamal K. Bharadwaj "Genetic programming approach to hierarchical production discovery," *in Proc. 5th International Conference on Databases and Data Mining (DBDM2005),* vol. 6, Istanbul, Turkey, June 2005, pp. 271-274.

[4] M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Transaction on Evolutionary Computations* 5(1), pp. 17-26, 2001.

[5] I. De Falco, A. Della Cioppa and E. Tarantino, "Discovering interesting classification rules with genetic programming," *Applied Soft Computing*, 1, pp. 257-269, 2002.

[6] K. A. De Jong, W.M. Spears and D. F. Gordon, "Using genetic algorithms for concept learning," *Machine Learning*, vol. 13, pp. 161-188, 1993.

[7] W. Romao, A.A. Freitas and I.M. de S. Gimenes, "Discovering interesting knowledge from a science and technology database with a genetic algorithm," *Applied Soft Computing*, 4, pp. 121-137, 2004.

[8] U. M. Fayyad, G. P. Shapiro and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communication of ACM.* Nov. vol. 39 (11), pp. 27-34, 1996.

[9] M. V. Fidelis, H. S. Lopes and A. A. Freitas, "Discovering comprehensible classification rules with a genetic algorithm," *in Proc. Congress on Evolutionary Computation-2000 (CEC'2000)*, La Jolla, CA, USA, IEEE, July 2000, pp. 805-810.

[10] A.A. Freitas, "A survey of evolutionary algorithms for data mining and knowledge discovery," In: A. Ghosh and S. Tsutsui (Eds.) *Advances in Evolutionary Computation,* Springer-Verlag, 2002.

[11] B.R. Gaines, "Transforming rules and trees into comprehensible knowledge structures," AAAI, MIT Press, 1995.

[12] B.R. Gaines and P. Compton, "Induction of ripple down rules applied to modeling large database," *Journal of Intelligent Information System*. 5(3), pp. 211-228, 1995.

[13] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," *Addison-Wesley*, 1989.

[14] Han and Kamber, "Data mining: concepts and techniques," *Academic Press*, 2001.

[15] N. M. Hewahi and K. K. Bharadwaj, "Bucket brigade algorithm for HCPRs based system," *International Journal of Intelligent Systems*, 11, pp. 197-226, 1996.

[16] F. Hussain, H. Liu and E. Suzuki, "Exception rule mining with a relative interestingness measure," *in Proc. 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2000, pp. 86-97.

[17] N. K. Jain, K.K. Bharadwaj and N. Marranghello, "Extended of hierarchical censored production rules (EHCPRs) system: An approach toward generalized knowledge representation," *Journal of Intelligent System*, 9(3, 4), pp. 259-295, 1999.

[18] J. Kivinen, H. Mannila and E. Ukkonen, "Learning rules with local exceptions," *EuroCOLT*, 1994, pp. 35-46.

[19] W. Kwedlo and M. Kretowski, "Discovery of decision rules from databases: an evolutionary approach," *in Proc. 2nd European Symp. On Principles of Data Mining and Knowledge Discovery (PKDD'98),* Lecture Notes in Computer Science 1510, Springer, 1998, pp. 370-378.

[20] B. Liu, M. Hu and W. Hsu, "Intuitive representation of decision tress using general rules and exceptions," AAAI-2000, 2000.

[21] R. S. Michalski and P. H. Winston, "Variable precision logic," *Artificial Intelligence*, vol. 29, pp. 121-146,1986.

[22] N. J. Radcliffe and P. D. Surry, "Co-operation through hierarchical competition in genetic data mining," *EPCC-TR94*-09, 1994.

[23] K.K. Bharadwaj, Neerja and G.C. Goel, "Hierarchical censored production rules system employing Dampster-Shafer uncertainty calculus", *Information and Software Technology*, 36, pp.155-164, 1994.

[24] P. Haddaway, "A variable precision logic inference system employing the Dempster-Shafer uncertainty calculus", MS Thesis (UILU-ENG-86-1777), University of Illinois, Urbana-Champaign, IL, USA, 1987.

[25] E. Suzuki, and J.M. Zytkow, "Unified algorithm for undirected discovery of exception rules," *International Journal of Intelligent Systems*, vol.20, pp. 673-691, 2005.

[26] B. Alatas and A. Arslan, "Mining of interesting prediction rules with uniform two-level genetic algorithm," *International Journal of Computational Intelligence,* vol. 1, no. 4, pp. 276-281,2004.

[27] M. C. J. Bot and W. B. Langdon, " Application of genetic programming to induction of linear classification trees," *Genetic Programming: Proc. of the 3rd European Conference (EuroCP'2000),* Lecture Notes in Computer Science, 1802,Springer, 2000, pp. 247-258.

[28] K.K. Gundogan, B. Alatas, A. Karci and Y. Tatar, "Comprehensible classification rule mining with two-level genetic algorithm," *in Proc. 2nd FAE International Symposium*, Cyprus, 2002, pp. 373-377.

[29] S. Bhattacharyya, "Evolutionary algorithms in data mining: multi-objective performance modeling for direct marketing," *in Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2000)*, ACM, 2000, pp. 465-473.

[30] UCI Repository of Machine Learning Databases. Department of Information and Computer Science University of California, 1994. Available: http://www.ics.uci.edu/~mlearn/MLRepositry.html.