ISSN:0975-9646
Amarnath Pathak et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 3023-3030

Classification Rule and Exception Mining Using Nature Inspired Algorithms

Amarnath Pathak
Dept. of Computer Science and Engineering
Guru Jambheshwar University of Sci.&Tech.
Hisar,Haryana,India

Jyoti Vashistha
Dept. of Computer Science and Engineering
Guru Jambheshwar University of Sci. & Tech.
Hisar, Haryana, India

***Abstract*— Classification is an important data mining task which facilitates list of decision rules that helps us to predict class of an unseen instance. Various traditional techniques like Decision tress, Neural Networks, SVMs have been used in past for rule mining. Nature Inspired Algorithms (NIAs) are class of algorithms that mimic natural processes and are capable of mining comprehensible and accurate rules. It is interesting to investigate Nature Inspired Algorithms (NIAs), exclusively GA and ACO, in context of rule mining. Classification model usually represents obvious information in form of decision rules and an unseen instance is liable to be misclassified if the model created using any of the above techniques do not account for exceptions present in the dataset. Instances having low support count and deviating from obvious behavior are termed as exceptions and they are less likely to be discovered using the usual rule discovery measures that account for generality of the discovered knowledge. In this paper we have investigated use of NIAs in rule mining and exception mining and we have also suggested possible modification in existing cAntMiner$_{pb}$ algorithm for mining exceptions.**

***Index Terms*—Nature Inspired Algorithms, Genetic Algorithm (GA), Ant Colony Optimization (ACO), Exceptions.**

I. INTRODUCTION

Classification means generating classification model (rule set) from given training data and using this model to classify unseen instances. An efficient classification model is characterized by four major features namely simplicity, comprehensibility, accuracy and interestingness. A tradeoff among all these features is desired while generating a model. Several different techniques have been suggested in past to discover classification model. Some of these techniques include- Decision trees, Artificial Neural Network, Naïve Bayesian Classifier, Support vector Machines and many more.

Recent research works are extensively employing nature inspired algorithms (NIAs) in rule mining and needless to mention, such attempts are furnishing promising results. NIAs take inspiration from nature for the development of novel problem solving techniques and employ natural materials (e.g. molecules, chromosomes) to compute. NIAs have two broad categories namely evolutionary algorithms (GA and GP) and swarm intelligence algorithms (ACO and PSO). Techniques of ACO and PSO are derived from collective behavior (ability) of ants and bees respectively to find optimal path from food source to their colony. Similarly the idea of Genetic Algorithm originates from natural process of evolution which says that individuals suited to the environment survive, reproduce and pass their genetic traits to offsprings. Research works reveal that these algorithms are capable of generating rules matching all our expectations and producing promising results.

Rule discovery techniques, traditional or NIAs, leave out the instances having low support and only discover what is obvious i.e. general rules. Such left out instances which deviate from the obvious behavior are often termed as exceptions.

Consider the rules,
If legs= 4 then class=mammal (i)

If legs = 4 Then class = mammal unless hair = FALSE (amphibian) (ii)

(i) represents an obvious information while (ii) represents exceptional information. This means that if an animal has 4 legs then it would be a mammal but in some exceptional cases it could be amphibian if it does not possess hair. Augmentation of term, *hair=FALSE* changes the class from mammal to amphibian and hence such a term is exception.
At many occasions it becomes essential to mine exceptions as decision makers might be interested in exceptional information rather than obvious one. Moreover discovering exceptions and appending them with general rules enhances their interestingness.

In this paper we try to explore nature inspired algorithms-their working and their applications in context of rule mining and exception mining. We will also investigate possible modifications in existing nature inspired algorithms which might enable them to mine exceptions along with general rules.

The rest of the paper is organized as follows: Section II includes a discussion on nature inspired algorithms exclusively GA and ACO. Section III discusses rule mining using GA and ACO. Detailed discussion on exceptions and exception mining using GA and ACO is included in section IV. Section V points directions for future research and suggests possible modifications in existing ACO algorithms for mining exceptions. Finally section VI concludes the paper.

II. NATURE INSPIRED ALGORITHMS

Traditional optimization techniques lack global perspective and often get stuck in local optima. They often require knowledge of first/second order derivatives of objective functions and constraints. Besides, we require different traditional methods for different types of problems. Nature inspired algorithms, on the other hand, are non-traditional and computationally intelligent optimization algorithms. NIAs have capability to avoid convergence to local optima and ability to perform a flexible robust search for a good combination of terms involving values of the predictor attributes [1]. These reasons inspire us to investigate NIAs and explore their applications in context of rule mining and exception mining.

A detailed discussion on working of GA and ACO has been contained in this section.

(b) GENETIC ALGORITHM

Genetic algorithms (GAs) are among the most popular evolutionary algorithms in terms of the diversity of their applications. Evolutionary algorithms mimic the process of evolution and hence the name. GAs are the search algorithms based on the mechanics of natural selection and natural genetics. They are based on the survival of the fittest concept (Darwinian Theory) which says that only the fittest will survive, reproduce and procreate, and successive generations will become better and better compared to previous generations. Unlike traditional optimization algorithms GAs search for a population of points rather than a single point and while doing so they make use of stochastic transition rules in place of deterministic rules. GAs use objective function information and not the derivative or second derivative.

The evolution usually starts from a population of randomly generated individuals (chromosomes). Each of these individuals represents a point in search space. Fitness values of these individuals are calculated and better fit individuals are selected for next iteration. Selection of individuals on the basis of their fitness values is termed as fitness proportionate selection. Selected individuals then undergo crossover and mutation operations to generate new offspring. Crossover involves exchange of information between better fit individuals . Offspring generated as a result of crossover inherit characteristics of its parents. Different strategies of crossover are adopted based on the nature of optimization problem. Mutation involves altering some of the bits of an individual to allow search exploration. Usual practice is to keep the crossover rate moderate and mutation rate very low. Individuals generated after selection, crossover and mutation constitute the new population. These individuals are evaluated and the process continues. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. Working of GA is depicted in figure 1.1.

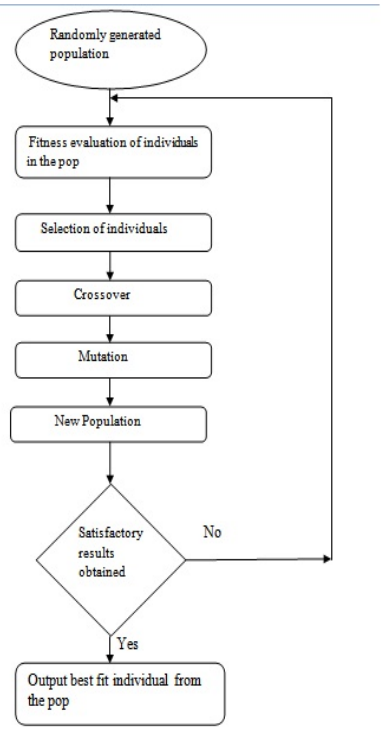

Figure 1.1

(a) ANT COLONY OPTIMIZATION

Ant Colony Optimization, a swarm intelligence technique, gets inspiration from ants' behavior for discovering rules from a given training data. Nature of ants is such that if we drop some food on ground, ants will find shortest (optimal) path between their nest and the food source. Many ant species even with limited visual capabilities or completely blind, are able to find the shortest path between food source and the nest by using pheromone as a communication mechanism [1]. Ants drop pheromone on the ground as they walk from nest to food source, thereby creating a pheromone trail on the used path. The pheromone concentration of a path influences the choice ants make and the more pheromone the more attractive a path becomes. Given that shorter paths are traversed faster than longer ones, these have a stronger pheromone concentration after a period of time, contributing to being selected and reinforced more often. Ultimately the majority of ants will be following the same path, most likely the shortest path. The pheromone evaporates over time to allow search exploration. ACO algorithms have been designed on similar grounds. Pheromone values and heuristic values are associated with each of the terms (attribute-value pairs) of training data. ACO algorithms use a colony of artificial ants, where ants build candidate solutions to optimization problems by iteratively selecting terms (attribute value pairs) based on their associated pheromone and heuristic information. The terms used to create good solutions will incur increase in their pheromone value while terms not used will incur gradual decrease in their pheromone values. At the end of this iterative procedure the artificial colony of ant will converge to an optimal or a near optimal solution. The optimal solution will comprise of terms having relatively high pheromone and heuristic values. Figure 1.2

best describes the ants' behavior and the idea used in ACO algorithms. This shows as to how colony of ants searches optimal path between food source and nest.

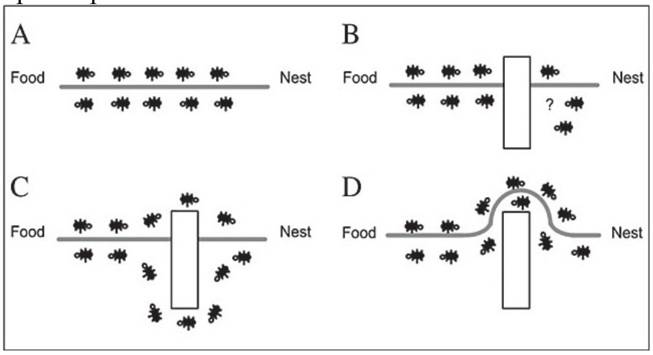

Figure 1.2

III. RULE MINING USING NIAS

In previous sections we have seen some of the advantages of NIAs over traditional rule mining techniques. Many NIAs particularly GA, ACO, PSO have gained immense popularity in the field of rule mining because of their simplicity, efficiency and computational intelligence. This section explores relevant rule mining works which have been performed using ACO and GA.

(b) CLASSIFICATION RULE MINING USING GA

For last few decades, GAs have been extensively employed in rule mining. GA has the capability of avoiding convergence to local optimal solutions and it also takes care of attribute interactions while evolving rules whereas most of the other rule induction methods tend to be quite sensitive to attribute-interaction problems. In case of GA, all the interactions among attributes are taken into account and fitness function evaluates the individual, as a whole.

Several GA designs, for discovering classification rules, have been proposed in the literature.

Comprehensible classification rules have been discovered by Fidelis et al.[1] by applying fixed encoding scheme to the chromosome and using specific design for mutation operator.

Jyoti et. al.[2] suggested a classification algorithm based on genetic algorithm approach that discovers comprehensible and interesting rules in CNF form in which along with conjunction in between various attributes there is disjunction among the values of an attribute. A flexible encoding scheme, genetic operators with appropriate syntactic constraints and a suitable fitness function to measure the goodness of rules are proposed for effective evolution of rule sets. A GA with entropy based filtering bias to initial population for automated rule mining has been proposed by Kapila et al. [3].

Dehuri & Mall [4] proposed a multi-objective algorithm for mining highly predictive & comprehensible classification rules from large databases. Goplan et al. [5] proposed a GA approach as a post-processing stage to discover accurate and interesting classification rules. Carvalho & Frietas [6] proposed a hybrid approach for rule discovery that combine decision trees and GA to automated discovery of small disjuncts rules. An accuracy-based learning system called DTGA(decision tree and GA) that aims to improve prediction accuracy over many classification problem proposed by Sarkar et al. [7].

Basheer et.al.[8] prove the worthiness of GAs in rule mining by discovering predictive, complete and comprehensible rules using GA. In the proposed work, GA scheme has been devised with flexible individual representation, appropriate genetic operators and effective fitness function. Predictive accuracy of proposed scheme has been tested against existing C4.5 and DTGA algorithm. The results proved that none of the selected learners improved the predictive accuracy on any dataset, as much as the proposed algorithm did.

However, the problem with the GA is high computational cost associated with fitness evaluations which discards the use of GAs for knowledge discovery from large datasets. Kapila et. al.[3] proposed an enhanced genetic algorithm for automated rule mining. It associates entropy based probabilistic initialization to reduce the search space and to reduce the number of fitness evaluations resulting in better fit rules and gain in run time. The enhanced GA has been applied on various datasets from UCI machine learning repository and has shown promising results.

(a) RULE MINING USING ACO

The first ACO algorithm for rule mining, commonly known as Ant-Miner, was proposed by Parpinelli, Lopes and Freitas[9] in year 2002. Ant-Miner algorithm has been referred by most of the algorithms that have been proposed afterwards. Following details with regard to Ant-Miner are worth mentioning [9]:

Path that an ant follows is assumed to be made up of attribute value pairs commonly known as terms. With all the terms we associate some pheromone value and heuristic value. All the terms are assigned an equal amount of pheromone value at the beginning of iteration. After a rule has been created, all the terms contained in it incur an increase in their pheromone value as per quality of rule and those not contained suffer evaporation of pheromone. Higher the pheromone value, higher will be the probability that term will be selected and added to the rule. Constructed rule is pruned before updating the pheromone value of terms contained in it. One approach to rule pruning is to remove one term at a time from the rule and measure the rule quality; if the rule quality increases the term is permanently dropped otherwise it is retained. Once the antecedents of a rule have been decided the next step is to decide rule consequent or the predicted class keeping in mind that it should maximize the rule quality. This is done by assigning to the rule consequent the majority class among the cases covered by the antecedents of the rule. Once the ants have discovered list of best rules, next step is to find class of an unseen instance using these rules. In order to decide class of an unseen instance, rules are applied in the same order in which they were discovered. Discovered rules are kept in an ordered list. First rule that covers the new test case is applied and the case is assigned the class predicted by the rule consequent. It is possible that no rule of the list covers the new case. In this situation the new case is classified by a default rule that simply predicts the majority class in the set of uncovered training cases.

In the same year 2002 a modification to above algorithm, Ant-Miner2, was suggested by Bo Liu et. al[10]. The algorithm proposed a new way of calculating heuristic function and hence a new way of selection of term because selection of term depends on heuristic function. Ant-Miner 2 proposed a density based heuristic function which ignored the accuracy of information contained in heuristic value, as it believed that small errors in heuristic value could be compensated by pheromone value. The new way of calculating heuristic function was computationally efficient and had same accuracy as that of heuristic function used in case of Ant-Miner 1.

Ant-Miner3 was the next modification proposed by Bo Liu et. al[11] in year 2003. Ant-Miner 3 proposed use of random numbers for selection of terms and a new method for updating the pheromone.

A new version of Ant-Miner, proposed by Frietas et.al.[12], for discovering unordered rule list, came in year 2006. New version discovered unordered rule set i.e. a set of rules which need not be applied to test data in the same order in which they were discovered. It was possible with some modification in high level algorithm, heuristic function and pheromone updating.

Parallel Ant-Miner algorithms[13][14], unlike previous works, were based on fixing consequent and discovering antecedents. Use of multiple processors, equal to number of consequents and division of ants into groups served the purpose of rule set discovery.

An important modification to original Ant-Miner, cAnt MinerPB algorithm, was proposed by Frietas in year 2012[15]. cAnt-MinerPB algorithm is given as:

```
Input: Training examples
Output: Best discovered rule list
Initialize pheromones();
list_gb <- ø;
m<- 0;
while m< maximum iterations and not stagnation do
    list_ib<- ø;
    for n<- 1 to colony_size do
        examples<- All training examples
        list_n<- ø;
        while |examples|> maximum uncovered do
            ComputeHeuristicInformaation(examples);
            rule<- CreateRule(examples);
                Prune(rule);
            examples<-examples-covered(rule,examples);
            list_n <- list_n+ rule;
        end while
        if Quality(list_n) > Quality(list_ib) then
            list_ib<- list_n;
        end if
    end for
    UpdatePheromones(list_ib);
    if Quality(list_ib) > Quality(list_gb) then
        list_gb<- list_ib;
    end if
    m<- m+1;
end while
return list_gb;
```

In case of original Ant-Miner, although, rules are discovered in an one-at-a-time fashion, the outcome of a rule (i.e. the examples covered by the rule) affects the rules that can be discovered subsequently since the search space is modified due to the removal of examples covered by previous rules. The sequential covering strategy of Ant-Miner performed a greedy search for a list (sequence) of rules which is not guaranteed to be the best list of rules that covers the training set, since the interaction between rules is not taken into account during the search. In order to mitigate the problem of rule interaction, the proposed strategy of Frietas incorporates the ideas of Pittsburgh approach into Ant-Miner's sequential covering strategy and hence the name cAnt-MinerPB algorithm.

cAnt-MinerPB algorithm works as follows: An ant in the colony(corresponding to an iteration of the for loop) starts with an empty list of rules and adds one rule at a time to that list, while the number of uncovered examples is greater than a user specified maximum value. After a rule is created and pruned, the training examples covered by the rule are removed and the rule is added to the current list of rules. Heuristic information associated with attribute value pairs is recalculated at each iteration of the list creation process (while loop) in order to reflect potential changes in the predictive power of the terms due to removal of training examples covered by previous rules. Once ant has discovered rule list, quality of this list is compared with iteration best rule list and if it has higher quality then iteration best rule list is replaced by it. So after every ant has discovered rule list we get the iteration best rule list. Pheromone values are updated based on iteration best rule list and updated pheromone values are used in upcoming iteration. The end of for loop yields iteration best rule list which is compared with global best rule list. Global best rule list is updated if iteration best rule list exceeds it in terms of quality of rules. The algorithm finally returns the global best rule list.

Extension of the cAnt-MinerPB algorithm to create unordered rules was proposed by Alex A. Freitas et. al in year 2013[16]. In the same paper he also proposed a new measure to evaluate the size of the discovered model and a new measure to characterize the interpretability of the discovered rules.

III. EXCEPTIONS

Exceptions contradict the prior knowledge about the domain hence their discovery is considered interesting. Discovery of exceptions adds curiosity and improves the quality of decision making in those rare circumstances where rules cease to work. Exceptions have low support and cannot be discovered using the usual rule discovery measures that account for generality of the discovered knowledge. Therefore, it would be useful to augment the default/general 'If-Then' decision rule (a rule with high support and confidence) with exceptions.

Jyoti et.al. [17] have nicely categorized exceptions into two types-intra class and inter class exceptions. The attribute value pair which does not change the class of decision rule when augmented with antecedent part is called intra class exception. On the contrary, inter class exceptions change the class of decision rules when augmented with the antecedent part. The intra-class exceptions relate to the unique and interesting features of an object within the class to which it belongs while the inter-class exceptions are the rare features that change the class of an object.[17] To further elaborate the types of exceptions we consider a dataset having 4 predicting attributes A, B, D and E and a class attribute C. Next we consider a decision rule of the form:

IF (A=a) and (B=b1) and (E=e) THEN (C=c1)

Suppose, very few instances in the dataset are of the form (A=a), (B=b2), (E=e) and (C=c1). Such an instance disagrees to the above rule. Even though the value of B changes from b1 to b2, class C remains the same. Such an attribute value pair namely (B=b2) is intra class exception and it should be appended with the general decision rule to make the rule more interesting. On the contrary, we could also have dropped the term B=b1 and form a rule:

IF (A=a) and (E=e) THEN (C=c1).

However, this will become a more generalized and possibly less interesting rule and hence such an attempt should be avoided.

Suppose the dataset also contained a very few instances of the form (A=a), (B=b1), (E=e), (D=d) and (C=c2). Appending an extra term namely (D=d) changes the class of object from c1 to c2. Such a term is an inter class exception. Decision rule augmented with such an inter class exception can be written as:

IF (A=a) and (B=b1) and (E=e) THEN (C=c1) UNLESS (D=d).

Obviously a rule of this form possesses more interestingness than our general decision rule and our future work aims at discovering such inter class exceptions using ACO.

(a) EXCEPTION DISCOVERY USING GA

Many decision rule formats have been suggested in past to facilitate exception discovery using NIAs (GAs in particular). This sub-section reviews all such rule formats and discusses exception mining using GA.

Measures of interestingness used in data mining literature have been reviewed by Jyoti et.al[18]. The main objective of this paper is to improve the understanding of interestingness measure for discovery of knowledge and identify the unresolved problems to set the direction for future research in this area. It clarifies the meaning of interestingness and discusses several forms of it namely- objective measure, subjective measure and semantic measure. It also discusses several novel ideas which can be used to accommodate interesting knowledge in form of exception. Some of these ideas are- **Ripple Down Rules (RDRs), Rule Pair, Rule Triplet, Censored Production Rule (CPR), Hierarchical Production rule (HPRs), Hierarchical Censored Production Rules (HCPRs)** etc. In Ripple Down Rules (RDRs) exceptions are encapsulated with general rules. It has the following structure:

If condition Then conclusion Except
If ..Then..Except
If.....
Else if

If premise is true then conclusion is taken only if except part is not true. If premise part of a RDR is true and the condition in the Except part is also true then the decision will be made on the basis of Except part. If premise is false then Else part will be considered for decision.[18].

Discovery of exceptions in form of rule pair has been suggested by Suzuki et.al. [19] and reviewed by Jyoti et. al[18] . Such a rule pair is given by:

If Yμ then x (strong rule)
If Yμ Λ Z then x' (exception)

In the rule pair above, Yμ and Z are conjunction of attribute values and, x and x' are class attribute values. Strong rule is a rule which has high recall and precision i.e. it accounts for most of the instances of dataset. The one which are not covered by this rule are taken care of, by exceptions. So exceptions have low support but high confidence. The attribute value pair Z appended to Yμ changes the polarity of the class x to x', making Z an example of inter class exception. Some of the examples of interesting rule pairs discovered by Suzuki (2004)[20] from real world datasets are given below.

IF (used_seat_belt = 'yes') THEN (injury = 'no')
IF ((used_seat_belt = 'yes') Λ (passenger = 'child')) THEN (injury = 'yes')

Rule triplet [21] [18] is an extension to the rule pair structure. Syntax of rule triplet is given by:

If Yμ*Then* x (commonsense rule)
If Yμ /\ Z *Then* x' (exception)
If Z *Then* x' (reference rule)

CPR is an excellent rule structure that supports an efficient mechanism for handling exceptions. It is another framework for discovery of exceptions. A Censored Production Rule (CPR) is of the form:

If P Then D Unless C

where P is the premise part which is a conjunction of attribute-value pairs. C may contain a single exception/censor or it may be disjunction of exceptions/censors. A classification algorithm based on evolutionary approach for discovering comprehensible rules with exceptions in the form of CPRs is presented by Saroj and Bharadwaj (2007)[21].

An HPR (Hierarchical Production Rule) , a standard production rule augmented with generality and specificity information, is of the following form:

<Decision> If < conditioin
Generality <general-information>
Specificity <specific-information>

Hierarchical representation used in HPR allows us to easily manage the complexity of knowledge, to view the knowledge at different levels of details, and to focus our attention on the interesting aspects only [18].
Further, a HCPR based system provides a general framework for intelligent systems that supports variable precision logic, excellent mechanism for handling exceptions, various machine learning paradigms and various inference mechanisms [48]. A HCPR has the following form:

< Decision> If <precondition>
Unless <censor-conditions>
Generality <general-information>
Specificity <specific-information>

HCPR not only accommodates exceptions but it also contains the general and specific concepts related to the class under consideration[18].
Silberschatz et. al.[23,24] address exception as rules that are contrary to the users' held beliefs. They also proposed a measure assessing how much a new pattern changes the degrees of a belief system. On one hand, directed methods are useful in discovering rules each of which is unexpected from the user point of beliefs, on the other it suffers from users' subjective biases or their lack of expertise about the domain. In an undirected approach no background knowledge is provided and the data mining algorithm alone discovers rule pairs.
Hussain et al. [25] have provided the objective and unbiased (from users' belief point of view) measure of the interestingness of a newly discovered rule in relation to already mined default rules. The formula for the Relative Interestingness (RI) measure is derived based on information theory and statistics. The measure has two components – interestingness based on the rule's support and interestingness based on the rule's confidence.
Lots of research works focus on discovery of exceptions and other interestingness measures using genetic algorithm. A genetic algorithm based approach is proposed for the discovery of production rules in CNF form that allows conjunction in between various attributes and disjunction among the values of an attribute. The crowding genetic algorithm scheme has been devised with flexible chromosome encoding, appropriate crossover and mutation operators, and appropriate fitness function. The underlying rule encoding in CNF form successfully captures the interestingness in form of exceptions that have relatively low support and high confidence [26].

The proposed work by Jyoti et.al [17] uses technique of genetic algorithm for mining intra class exceptions. Proposed design comprises of two stages: In the first stage the default rule set is discovered using GA and in the second stage there is inclusion of intra and inter class exceptions to the rules. The 'Michigan-style' approach has been used to encode each solution in the population. Besides, a crowding GA has been implemented with the intention of discovering a set of classification rules. This strategy maintains the diversity in the population and avoids convergence of the traditional GA to the single best solution.
A wide variety of interestingness measures are available in data mining literature, exception is one of them. It is difficult for users to select appropriate measure in a particular application domain. Garima et.al.[27]compare these interestingness measures on diverse datasets by using GAs and select the best one according to situation.
Works of Saroj et.al [28] deal with discovery of fuzzy censored classification rules (FCCRs) from datasets using GA. Such FCCCRs are good at handling vagueness, uncertainty and offer an excellent exception handling mechanism.
Above research works suffice to prove effectiveness of GA in mining exceptions and other interestingness measures from a given dataset.

(a) EXCEPTION DISCOVERY USING ACO

To the best of our knowledge, the use of ACO algorithms for discovering exceptions along with classification rules, in the context of data mining, is a research area still unexplored. However, exception mining using ACO has been attempted in context of fuzzy rules. Work of Carmona, P., Castro, J., Zurita, J [29] in this regard is worth mentioning who have proposed algorithm for mining fuzzy rules along with exceptions. An enhancement to the above approach has been suggested by the same author [30], where they have used the Ant Colony Optimization plug-in to enhance the interpretability of fuzzy rule bases with exceptions. In this paper they have proposed an extension on the syntax of fuzzy rules by including new predicates and exceptional rules and, on the other hand, the use of an ant colony optimization algorithm to obtain an optimal set of such rules that describes an initial fuzzy model. Further improvement in the above work has been proposed by the same authors [31] where they have suggested several extensions on that algorithm in order to improve the interpretability of the obtained fuzzy model, as well as the computational cost of the algorithm. The first extension consists of replacing the AS model used in the original method with the more advanced ACS model. The second extension adds a local search to refine each solution, a common practice in this and other metaheuristics. Finally, a third extension proposes to use a candidate list in order to restrict the number of steps considered to select the next one, trying that way to diminish the computational cost of the algorithm. Some other works which deal with the issue of interestingness and exceptions in bioinformatics are [32] and [33]. James Smaldon et.al [32] propose a modification in rule induction algorithms aimed at improving the interpretability of the discovered rules. This modification is proposed in the context of sparse bioinformatics data sets where the presence of a feature is much less common than its absence, so that rule conditions with positive values of the feature tend to be more informative than rule conditions with negative values of that feature. The proposed modification consists of inducing only rules having positive values of the features, rather than rules using both positive and negative values of the features. The central idea of this paper is to modify two rule induction algorithms to discover rules having in their antecedent only conditions of the form "IF a protein has biological motif X", and not conditions of the form "IF a protein does not have biological motif X", in order to improve the interpretability of the discovered rules and to increase their interestingness.

Gisele et.al [33] address the problem of predicting whether or not a protein has post-synaptic activity. This problem is of great intrinsic interest because proteins with postsynaptic activities are connected with functioning of the nervous system.

IV. FUTURE SCOPE

We have seen that exception discovery along with classification rules, using ACO, is a research area still unexplored. Thus it would be interesting to extend existing ACO algorithms to mine exceptions by making suitable modifications in existing ACO algorithm and by using parameters gamma1 and gamma2 [17]. cAnt-MinerPB algorithm can be modified to accommodate exception discovery module. Once a general decision rule has been discovered, the module would traverse the entire data set looking for possible interclass exceptions corresponding to this rule. CPR framework can be used to represent discovered rules plus exceptions.

Second, it would be interesting to extend existing ACO algorithms to cope with continuous attributes, rather than requiring that these kinds of attributes be discretized in a preprocessing step.

V. CONCLUSION

In this paper we have investigated nature inspired algorithms namely ACO and GA-their working and their applications in rule mining and exception mining. We conclude that these algorithms are quite efficient, simple, computationally intelligent and able to discover classification model which is comparable (in terms of accuracy) to model designed using traditional techniques. Finally we suggested that discovery of rules plus exceptions using ACO, an unexplored research topic, can be achieved by modifying cAntMiner$_{pb}$ algorithm- an existing ACO algorithm.

ACKNOWLEDGEMENT

Authors would like to thank Dr. Saroj Ratnoo for her immense support and valuable suggestions.

REFERENCES

[1] Fidelis, M.V., Lopes, H.S., Freitas, A.A. and Grossa, P. 2000. Discovering comprehensible classification rules with a genetic algorithm. In Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, IEEE, vol. 1, pp. 805-810.

[2] Jyoti Vashishtha, Dharminder Kumar, Saroj Ratnoo, Kapila Kundu (2011), "Mining Comprehensible and Interesting Rules: A Genetic Algorithm Approach", International Journal of Computer Applications (0975 – 8887) Volume 31– No.1, October 2011

[3] Kaplia, Saroj, Kumar D. and Kanika. 2010. "A genetic algorithm with entropy based initial bias for automated rule mining. In Proceeding of the IEEE International Conference on Computer & Communication Technology(ICCCT"10), pp. 491-495.

[4] Dehuri, S. and Mall, R. 2006. "Predictive and comprehensible rule discovery using a multi objective genetic algorithms". Knowledge Based Systems, vol. 19, pp. 413-421.

[5] Goplan J., Alhajj R. and Barker, K. 2006. Discovering accurate and interesting classification rules using genetic algorithm. In Proceedings of the International Conference on Data Mining(DMIN06), Las Vegas, Nevada, USA , pp. 389-395.

[6] Carvalho, D. R. and Freitas, A. A. 2002. "A genetic-algorithm for discovering small-disjunct rules in data mining". Applied Soft Computing, vol. 2, pp.75-88.

[7] Sarkar, B. K., Sana, S.S. and Chaudhuri, K. 2012. A genetic algorithm-based rule extraction system. Applied Soft Computing. vol. 12, pp. 238-254.

[8] Basheer M. Al-Maqaleh, Hamid Shahbazkia, "A Genetic Algorithm for Discovering Classification Rules in Data Mining", International Journal of Computer Applications (0975 – 8887) Volume 41– No.18, March 2012

[9] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas(Aug. 2002), "Data mining with an ant colony optimization algorithm," *IEEE Trans. Evol. Comput*, vol. 6, no. 4, pp. 321–332.

[10] Bo Liu, Hussein A.Abbass, Bob McKay(2002), "Density_based Heuristic for Rule Discovery with Ant_Miner," The 6th Australia-Japan Joint Workshop on Intelligent and Evolutionary System, page 180-184

[11] Bo Liu, Hussein A.Abbass, Bob McKay(2003), "Classification Rule Discovery with Ant Colony Optimization," International Conference on Intelligent Agent Technology, 2003,IAT 2003, IEEE/WIC, October 13-16, 2003, page 83-88.

[12] J. Smaldon, A. A. Freitas(2006), " A new version of the Ant_miner algorithm for discovering unordered rule sets," Genetic and Evolutionary Computation Conference (GECCO-2006), pp.43-50, ACM Press, New York.

[13] Chen H., Chen L.,Li,T.(2006),"Parallel Ant Colony Algorithm for mining Classification Rules," 2006 IEEE International Conference on Granular Computing, May 10-12,2006,pp. 85-90.

[14] Omid Roozmand, Kamran Zamanifar(2008), "Parallel Ant Miner 2," Artificial Intelligence and Soft Computing – ICAISC 2008, Volume 5097, 2008, pp 681-692.

[15] F.E.B Otero,, A.A. Freitas, C.G Johnson(Feb. 2012.)," A New Sequential Covering Strategy for Inducing Classification Rules With Ant Colony Algorithms,", *IEEE Trans. Evol. Comput*, vol. 17, pp. 64 – 76.

[16] F.E.B Otero,, A.A. Freitas(2013),"Improving the Interpretability of Classification Rules Discovered by an Ant Colony Algorithm" Genetic and Evolutionary Computation Conference (GECCO-2006), pp.73-80, ACM Press, New York.

[17] Jyoti Vashishtha, Dharminder Kumar and Saroj Ratnoo(2013) "An evolutionary approach to discover intra- and inter-class exceptions in databases" Int. J. Intelligent Systems Technologies and Applications, Vol. 12, Nos. 3/4 pp 283-300.

[18] Jyoti Vashishtha, Dharminder Kumar, Saroj Ratnoo(2012)," Revisiting Interestingness Measures for Knowledge Discovery in Databases" 2012 Second International Conference on Advanced Computing & Communication Technologies

[19] Suzuki, E.(2002) "Undirected discovering of interesting exception rules", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 16, No. 8, pp.1065–1086.

[20] Suzuki, E. (2004) "Discovering interesting exception rules with rule pair", ECML/PKDD-04: Proceedings of the Workshop on Advances in Inductive Rule Learning, Pisa, Italy, pp.163–178.

[21] Suzuki, E. and Żytkow, J.M. (2005)"Unified algorithm for undirected discovery of exception rules", International Journal of Intelligent Systems, Vol. 20, No. 7, pp.673–691.

[22] Saroj, S. and Bharadwaj, K.K. (2007) "A parallel genetic algorithm approach for automated discovery of censored production rules", AIA-07: Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and application, Innsbruck, Austria, pp.435–441.

[23] Silberschatz, A. and Tuzhilin, A. (1995) "On subjective measures of interestingness in knowledge discovery", KDD-95: Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, pp.275–281.

[24] Silberschatz, A. and Tuzhilin, A. (1996) "What makes patterns interesting in knowledge discovery systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No.6, pp.970–974.

[25] Hussain, F., Liu, H., Suzuki, E. and Lu, H. (2000) "Exception rule mining with a relative interestingness measure", in PAKDD-2000: Proceedings of Pacific Asia Conference on Knowledge Discovery in DataBases, Kyoto, Japan, pp.86–97.

[26] Jyoti Vashishtha, Dharminder Kumar, Saroj Ratnoo, Kapila Kundu: "Mining Comprehensible and Interesting Rules: A Genetic Algorithm Approach", International Journal of Computer Applications (0975 – 8887), Volume 31– No.1, October 2011

[27] Goyal Garima and Jyoti Vashishtha, "A Study of Interestingness Measures for Knowledge Discovery in Databases—A Genetic Approach", L.C. Jain et al. (eds.), Computational Intelligence in Data Mining - Volume 2, Smart Innovation, Systems and Technologies 32, DOI 10.1007/978-81-322-2208-8_8, Springer India 2015.

[28] Saroj, Nishant Prabhat, "A Genetic-Fuzzy Algorithm to Discover Fuzzy Classification Rules for Mixed Attributes Datasets", International Journal of Computer Applications (0975 – 8887) Volume 34– No.5, November 2011.

[29] Carmona, P., Castro, J., Zurita, J.: FRIwE: Fuzzy rule identification with exceptions. IEEE Trans. Fuzzy Syst. 12(1), 140–151 (2004)

[30] Carmona, P., Castro, J.: An Ant Colony Optimization plug-in to enhance the interpretability of fuzzy rule bases with exceptions. In: Analysis and Design of Intelligent Systems Using Soft Computing Techniques. Advances in Soft Computing, vol. 41, pp. 436–444. Springer, Heidelberg (2007)

[31] Pablo Carmona and Juan Luis Castro : "An Improved ACO Based Plug-in to Enhance the Interpretability of Fuzzy Rule Bases with Exceptions" M. Dorigo et al. (Eds.): ANTS 2008, LNCS 5217, pp. 13–24, 2008. Springer-Verlag Berlin Heidelberg 2008

[32] James Smaldon and Alex A. Freitas "Improving the Interpretability of Classification Rules in Sparse Bioinformatics Datasets" Genetic and Evolutionary Computation Conference (GECCO-2007), pp.75-80, ACM Press, New York.

[33] Gisele L. Pappa1, Anthony J. Baines and Alex A. Freitas: "Predicting Post-Synaptic Activity in Proteins with Data Mining" Vol. 00 no. 0 2005, pages 1–5 doi:10.1093/bioinformatics/bti283.