Cultural Algorithm Toolkit for Interactive Knowledge Discovery

Sujatha Srinivasan[1] and Sivakumar Ramakrishnan[2]

[1]Dept. of Computer Science, Cauvery College for Women, Trichirappalli, India
ashoksuja03@yahoo.co.in
[2]Dept. of Computer Science, AVVM Sri Pushpam College, Poondi, Tanjore, India

ABSTRACT

Cultural algorithms (CA) are inspired from the cultural evolutionary process in nature and use social intelligence to solve problems. Cultural algorithms are composed of a belief space which uses different knowledge sources, a population space and a protocol that enables exchange of knowledge between these sources. Knowledge created in the population space is accepted into the belief space while this collective knowledge from these sources is combined to influence the decisions of the individual agents in solving problems. Classification rules comes under descriptive knowledge discovery in data mining and are the most sought out by users since they represent highly comprehensible form of knowledge. The rules have certain properties which make them useful forms of actionable knowledge to users. The rules are evaluated using these properties represented as objective and subjective measures. Objective measures are problem oriented while subjective measures are more user oriented. Evolutionary systems allow the user to incorporate different rule metrics into the solution of a multi objective rule mining problem. However the algorithms found in the literature allow only certain attributes of the system to be controlled by the user. Research gap exists in providing a complete user controlled system to experiment with evolutionary multi objective classification rule mining. In the current study a Cultural Algorithm Toolkit for Classification Rule Mining (CAT-CRM) is proposed which allows the user to control three different set of parameters. CAT-CRM allows the user to control the evolutionary parameters, the rule parameters as well as agent parameters and hence can be used for experimenting with an evolutionary system, a rule mining system or an agent based social system. Results of experiments conducted to observe the effect of different crossover rates and mutation rates on classification accuracy on a bench mark data set is reported.

KEYWORDS

Multi objective optimization; Classification rules; Evolutionary algorithm; Social intelligence; Cultural algorithm; Data mining.

1. INTRODUCTION

Hybridization of techniques from various domains is an active area of research in computer science. Artificial intelligence, Swarm intelligence, Evolutionary algorithms, are called nature inspired computing (NIT) [1], while simulated annealing gets its inspiration from physical sciences. Derived from mathematics, Multi objective optimization is used in solving various optimization problems in engineering, computer science, and in particular data mining in finding optimal solutions. Whitaker [1] presents evidence suggesting that nature-inspired optimization techniques are now more frequently studied and utilized than mathematical optimization techniques and other meta-heuristics. Cultural algorithm (CA) is a class of evolutionary social system which was inspired by the evolution taking place in the society and which is used in solving optimization problems in various domains.

Classification rule mining is a class of problems where the knowledge mined is represented as "If-Then" rules and are most sought out since they are more comprehensible to the user. There are often objective and subjective measures to evaluate the rules. These measures are sometimes

DOI : 10.5121/ijdkp.2012.2505

called the properties of the rule. The classification rules have to satisfy some of these properties to be used as a good classifier. The metrics often used for evaluating the rules are support and confidence. However there are other properties like comprehensibility and interestingness of the rule that make the classifiers more actionable to the user. But the objectives used for evaluation of rules are sometimes conflicting. For example a user may wish to have rules which are both novel and are accurate. These two objectives are conflicting since an accurate rule may not be interesting to a user and vice versa. Thus the problem of discovering rules with specific properties should be faced as a multi-objective optimization problem where the maximization or minimization of each property is one single objective.

Evolutionary algorithms (EA) are nature inspired systems and works on the strategy of survival of the fittest. Evolutionary multi objective optimization (EMOO) systems have been proposed in the literature to solve rule mining as a multi objective optimization problem. EMOO systems allow differing trade-offs to be incorporated into a multi objective problem. However research gap exists in giving the user freedom to control most of the parameters of an evolutionary multi objective system especially for classification rule mining. Therefore a Cultural Algorithm Toolkit for Classification Rule Mining (CAT-CRM) is proposed in the current study for interactive knowledge discovery. CAT-CRM combines the strengths of Evolutionary Computing, Social computing, Data mining and Artificial Intelligence in a Cultural algorithm framework. Cultural algorithm (CA) is an evolutionary algorithm that was introduced by Reynolds in 1994 [2] inspired by the social learning occurring in the society. CA best represents social systems in which agents thrive to optimize their utilities using various types of knowledge sources (KS) known as belief space. CA consists of two levels of evolution: the microevolution in a population space and the macroevolution in the belief space. The experiences of individuals in the population space are used to generate problem solving knowledge that is to be stored in the belief space which then manipulates the knowledge and in turn guides the evolution of the population space by means of an influence function [3]. Cultural algorithms have been used for modeling the evolution of complex social systems and for solving various optimization problems. The problem and related work on interactive evolutionary multi objective systems for rule mining and a short review of cultural algorithms is discussed in Section 2. Section 3 describes the proposed CAT-CRM. The proposed CAT-CRM is illustrated by applying it to a bench mark classification problem. Section 4 discusses experiments and results on the benchmark Iris data set. Section 5 concludes with future work.

2. PROBLEM AND RELATED WORK

2.1 Problem

Given a data source, the problem is presenting the user with a system which allows the user to control the various parameters of the system including evolutionary parameters, rule parameters and agent parameters so that the user can experiment with the system to find the influence of these parameters in discovering rule sets with differing tradeoffs in the solution and thus allowing the user to choose the best.

2.2 Aims of the study

i. Incorporating user preferences to control various parameters of an EMOO system for interactive knowledge discovery.

ii. Adding knowledge to Evolutionary algorithms which are blind search methods to improve the performance of the system for finding better solutions.

2.3 Related work

Evolutionary computing has been used extensively in data mining. Evolutionary algorithms perform a global search and are convenient for parallelization [4]. They are robust search methods

that adapt to the environment and can discover interesting knowledge that will be missed by greedy algorithms [5]. Also they allow the user to interactively select interesting properties to be incorporated into the objective function providing the user with a variety of choices [6]. Thus Evolutionary algorithms are very suitable for multi-objective optimization since they allow various objectives to be simultaneously incorporated into the solution.

Participation of the user in the process of discovering knowledge is essential to improve the chance that discovered knowledge will be actually useful for the user [7]. Some systems allow the user to specify the metrics for optimization and/or the threshold values for rule selection while a very few systems allow the user to interact with the system during execution [8]. Iglesia et al. [6] [9], propose the use of multi-objective optimization evolutionary algorithms, to allow the user to interactively select a number of interest measures and deliver the best nuggets. Where in Iglesia et al. [9] propose to use Pareto-based MOEA to deliver nuggets that are in the Pareto optimal set according to some measures of interest which can be chosen by the user normally based on domain or expert knowledge. In Reynolds and Iglesia [10], the user selects a subset of the class of interest where the user is presented with a set of descriptions about the class. Presenting the user with a diverse set of rules is another area of data mining research which has been the basis of multi objective optimization in rule mining. The use of modified dominance relations have been used in [10] to increase the diversity of rules presented to the user and clustering techniques have been used in the presentation large sets of rules generated. The algorithm also considers misclassification cost and rule complexity as measures which are allowed to be controlled by the users. Whereas Reynolds and Iglesia, [11] allow the user to choose the mutation rate. In [12] the user is allowed to specify the goal attribute that is of interest to him which is used for mining highly predictive and comprehensible classification rules from large databases. Giusti et al. [13], allow the user to select a set of rules with specific properties in each generation to be used in subsequent generations. The multi objective algorithm proposed by Zhao [14] allows the decision maker to specify partial preferences on the conflicting objectives, such as false negative vs. false positive, sensitivity vs. specificity, and recall vs. precision to reduce the number of alternative solutions. This is one of a few systems which present the user with a graphical user interface. The user is allowed to choose a familiar visualization method including a ROC curve, sensitivity-specificity, precision-recall and false positive- false negative trade-offs to be visualized. The system also allows the user to visualize the progress of the evolution of solutions such that the decision maker can decide to stop the procedure when satisfactory solutions have been found or when the solutions on the front appear to have stabilized.

Apart from support, coverage and confidence of rules there are other measures that make the classifier appealing to the user such as surprisingness, interestingness, and comprehensibility of the rules. Moreover there are application specific metrics. For example sensitivity and specificity are rule metrics which are used in medical domain while precision and recall are measures used in information retrieval problems. MEPAR-miner (Multi-Expression Programming for Association Rule Mining) for rule induction is proposed in [4] which uses sensitivity and specificity of rules to define their fitness function. Reynolds et al., [15] describe the application of a multi-objective Greedy Randomized Search Procedure to rule selection, where previously generated simple rules are combined to give rule sets that minimize complexity and misclassification cost. A hybrid approach that combines a meta-heuristic and an exact operator is presented by Khabzaouil et al., [16] not only for finding non frequent rules but interesting ones also. The authors of [17], [18] propose Pitts-DNF-C, a multi-objective Pittsburgh-style Learning Classifier System that evolves a set of fuzzy rules for classification tasks. The system is explicitly designed to create consistent, complete, and compact rules for the user to comprehend. However evolutionary algorithms so far discussed are blind search methods and allow only partial preferences of the user to be incorporated into the system. Research gap exists in incorporating knowledge and user preferences into evolutionary systems to improve the performance and usability of the systems.

2.4 A short review of Cultural algorithms

Cultural algorithm is an evolutionary algorithm which is mostly applied in solving numerical function optimization problems and which has a set of five Knowledge sources for representing various primitive knowledge's and works on the strategy of survival of the fittest. The agents in the system affect the various Knowledge sources and the KS's in turn influence the agents thus directing them towards an optimal solution. Reynolds et al., [19], use cultural algorithm to solve numerical optimization problems to study the micro and macro evolution of the individuals and the system. The individuals are provided with five types of knowledge which are said to be primitive knowledge used by most living species including human beings. Cultural algorithm has been used in rule based systems. Sternberg and Reynolds [20] use an evolutionary learning approach based on cultural algorithms to learn about the behaviour of a commercial rule-based system for fraud detection. The learned knowledge in the belief space of the cultural algorithm is then used to re-engineer the fraud detection system. Lazar and Reynolds, [21] have used genetic algorithms and rough sets for knowledge discovery. Reynolds et al., [22] use decision trees to characterize location decisions made by early inhabitants at Monte Alban, a prehistoric urban centre, and have injected these rules into a socially motivated learning system based on cultural algorithms. They have then inferred an emerging social fabric whose networks provide support for certain theories about urban site formation. Reynolds and Mostafa, [23] propose a Cultural Algorithm Toolkit which allows users to easily configure and visualize the problem solving process of a Cultural Algorithm. The proposed system is applied in solving predator/prey problem in a cones world environment and engineering design.

However cultural algorithm toolkit for classification rule mining is hardly found in the literature. This paper makes a unique contribution by providing a Cultural Algorithm Toolkit for Classification Rule Mining (CAT-CRM) where the user can control three types of parameters namely the evolutionary parameters, the rule parameters and agent parameters. The system is designed considering rule mining as a multi-objective optimization problem and providing an evolutionary computation approach. The evolutionary parameters that can be controlled include the population size, the number of generations, crossover rate and mutation rate. The rule parameters that can be specified by the user are the rule metrics for optimization and a rule schema. The agent parameters include the number of agents of each type namely cautious, imitator and risk taker explained in later sections. Moreover by incorporating the various knowledge sources using a cultural algorithm framework, knowledge has been added to the otherwise blind evolutionary algorithm. This enables the system also to be used as a social computing system to study the micro and macro dynamics of a real world social system by analyzing the different knowledge sources once the algorithm terminates. The application of multi-objective evolutionary computation with Meta data for interactive rule induction using cultural algorithms (with cognitive agents) has not been explored yet and hence forms the inspiration for the current proposal.

3. CULTURAL ALGORITHM TOOLKIT FOR CLASSIFICATION RULE MINING (CAT-CRM)

Cultural Algorithm which derives from social structures, and which incorporates evolutionary systems and agents, and uses various knowledge sources for the evolution process better suits the need for solving multi objective optimization problem and has been used in different domains. CA has three major components: a population space, a belief space, and a protocol that describes how knowledge is exchanged between the first two components. The population space can support any population-based computational model, such as Genetic Algorithms, and Evolutionary Programming [22]. A Cultural Algorithm Toolkit for Classification Rule Mining (CAT-CRM) is proposed in the current study for studying the influence of various parameters on

an evolutionary multi objective classification rule mining system in presenting the user, solutions with differing trade-offs.

3.1 The Belief space

The belief space comprises of the five knowledge sources namely the Normative, Situational, Domain, Topographical and the History KS. For the rule optimization problem the belief space is modified to hold different types of knowledge or Meta data obtained during evolution which is used in successive generations for creating better individuals. Further an additional KS has been added to hold the rules. The agents in the CA have also been given social or cognitive traits which they use in decision making. The following section discusses the different knowledge sources.

3.1.1 Normative KS

Normative Knowledge Source (NKS) contains the attributes and the possible values that the attributes can take. This information is gathered from the training data set. The normative knowledge source is used to store the maximum and minimum values for numeric attributes. For each nominal or discrete attribute, a separate list is maintained that stores the possible values that the attributes can take. The normative KS is updated during train data set creation and used by the agents during mutation.

3.1.2 Situational KS

Situational knowledge source (SKS) consists of the best exemplar found along the evolutionary process. It represents a leader for the other individuals to follow. This way, agents use the example instead of a randomly chosen individual for the recombination. This KS can be updated by storing the best examples at the end of each generation. Agents use these examples for choosing individuals for reproduction. Also the user can specify schema with don't care conditions for certain attributes which can be used by agents for the search of similar/dissimilar individuals to interest the user. In the current study the SKS stores the schema specified by the user as a vector of attribute values.

3.1.3 Domain KS

Domain knowledge source (DKS) contains the vector of rule metric values for each rule along with a Rule Identifier (RuleId). Individuals produced by agents are evaluated at the end of each generation and the fitness vector calculated. DKS is updated with these fitness vectors. The fitness vectors in DKS are compared with each other using Pareto optimization strategy to choose elite individuals at the end of each generation. The elite individuals thus chosen are stored in the historical KS.

3.1.4 Topographical KS

Diversity maintenance strategy is a characteristic of Multi objective evolutionary systems for keeping the solutions uniformly distributed in the Pareto optimal set, instead of gathering solutions in a small region only. Restricted mating, where mating is permitted only when the distance between two parents is large enough, is one technique for maintaining diversity of rules [12]. In the proposed system Topographic knowledge is used to store the difference or distance between two rules for the purpose of maintaining diversity of rules. This KS is updated at the end of each generation. The topographical knowledge contains a pair of RuleId's and their dissimilarity measure. Since the attributes are discrete the attribute values in the corresponding positions in the individuals are compared and a value of 0 is assigned to the attributes with similar values and a value of 1 is assigned to dissimilar values. The number of 1's are counted and assigned as the dissimilarity measure for the pair of rules. Hence topographical KS can be used to create novel and interesting rules by choosing pairs of individuals with maximum dissimilarity measure.

3.1.5 History KS

History knowledge source (HKS) records in a list, the best individuals along with their RuleId's, and are updated at the end of each generation. Evolutionary algorithms are termed as memory less since they do not retain memory of previous generations. However attempts have been made to retain elite individuals of each generation as a separate elite population to render memory to the evolutionary algorithms. Cultural algorithm renders memory to the evolutionary strategy in a systematic way by using the different knowledge sources. History knowledge can be used to store elite individuals of each generation thus maintaining memory across generations.

3.1.6 The rule KS

The cultural algorithm is extended to contain the individuals produced during evolution using the Rule KS (RKS). The representation of the individuals in RKS is similar to that of the HKS. Each entry holds a RuleId and the attribute values as a vector. The RuleId is used as a pointer by the other KS's. RKS is added to CA in order to render memory by maintaining good individuals evolved across generations. New rules are added to RKS at the end of each generation while worst ones are removed.

3.1.7 Social Agents

The proposed CA is also extended by adding cognitive traits to the agents. In the original CA the agents are not distinguished but rather considered as having same properties and are used for exploring the solutions. But the proposed CA explicitly distinguishes agents with three traits namely imitator, cautious and risk taker. The agents use this trait in the selection of parents for reproduction using different knowledge sources. Imitators use the situational KS while cautious agents use historical KS for choosing parents for mating. Risk takers are explorers and use any of the different KS's at random. A random integer in the range 0 to size of the corresponding KS is generated and the individual in that particular location in the RKS or SKS or HKS or TKS are chosen and undergo crossover or mutation. If the KS chosen is TKS then the individuals with the maximum dissimilarity measure is chosen from TKS and reproduction operators are applied to the individuals. This enables creation of diverse set of individuals. Cautious agents use only the historical knowledge source while the imitators use the situational knowledge source to create individuals namely the example specified by the user. Reynolds et al., [19], state that agents that use situational and domain knowledge are exploiters while normative and topographical knowledge users are explorers and Historical knowledge users are good trend predictors. The agents can be allowed to change their traits by enabling them to change the type of knowledge source used.

3.2 Influence phase

The influence function decides which knowledge sources influence individuals. In the original CA, roulette wheel selection based on performance of knowledge sources in the previous generations has been used. In the proposed system selection is left to the agents. In the proposed CAT-CRM the agents use their social trait namely risk taker or imitator or cautious to choose parents for reproduction. Risk takers use knowledge from any of the four knowledge sources namely RKS, HKS, SKS or TKS at random while cautious agents use only the HKS. The imitators use SKS to create individuals using the example specified by the user. NKS which stores the possible attribute values is used by all the agents during the mutation operation. The topographical knowledge source enables creation of a diverse set of rules. DKS stores the values of the user specified metrics of the individuals as a fitness vector and thus is used for comparing individuals using Pareto comparison. The Rule KS is used to store the individuals (both dominators as well as non-dominators thus avoiding losing of good individuals of initial generations) created during evolution. Thus the KS's guide the agents in the evolution process. More social traits can be added to the agents for studying the effect of various traits and knowledge sources on the outcome of the system. The proposed system thus can also be used as a

social simulation system or virtual organization which can be used for studying the micro and macro dynamics of real world social systems.

3.3 Acceptance phase

The acceptance function determines which individuals and their behaviors can impact the belief space knowledge [19]. Based on selected parameters such as performance, for example, a percentage of the best performers (e.g., top 10%), can be accepted [19]. But since the problem is one of classification rule mining, a threshold value for the rule metrics specified by the user can be used to accept individuals for next generation. Since the current implementation is one of multi objective optimization, the algorithm produces a set of solutions and the dominators are chosen and stored in HKS using Pareto optimality, while other KS's are updated as explained earlier. The process of agent's selection, reproduction, evaluation and updating of belief space forms a generation. At the end of a generation (iteration), the agents return the individuals created by them along with a fitness vector of rule metric values. The knowledge sources are updated with this new knowledge at the end of each generation and thus evolve along with the agents. The new values in these KS's then influence the population space. Thus the macro evolution takes place.

3.4 Evolutionary strategy

Genetic algorithm (GA) is by far the most used evolutionary strategy which is also used in the current study. The various attributes of the GA used are discussed below.

3.4.1 Chromosome representation

The chosen data records are converted into fixed size chromosomes and represented as a vector of attribute values. The system uses high level encoding where the attribute values are used as they appear in the data source. This reduces the cost of encoding and decoding individuals for creating rules for large data sets. The relational operators are not included in the genotype as found in most algorithms in the literature. Therefore they are not involved in the reproduction which further minimizes the length of the chromosome and thus the time taken for encoding and/or decoding. This representation also avoids use of different types of reproduction operators for different parts of the chromosome.

In the current study the class attribute is also included in the chromosome during the training phase. During the test phase classes are assigned to individuals as follows: If more than 75% of the values in the antecedent part are equal in the rule created and the test data instance then that class is assigned. If more than one rule covers the test instance then the maximum occurring class label that covers the rule is assigned otherwise maximum occurring class in the data set is assigned.

3.4.2 Population initialization

Evolutionary systems work on a population of individuals. Population initialization is an important aspect that decides the overall performance of the algorithm. The initial population is created using various procedures. One procedure is seeding where data instances chosen from training data randomly as in [12] or with the help of the users as in [13], are used as initial seeds to fill the population space. Casillas et al., [17] state that the initialization procedure has to guarantee that the initial individuals cover all the input examples from the training data set. To ensure this, the authors of [6] use mutated forms of the default rule as initial solutions where the default rule is the rule in which all limits are maximally spaced and all labels are included. In the current study maximum and minimum chromosomes are used as seeds to create initial population. That is, the initialization procedure uses two initial chromosomes as seeds where one chromosome contains the minimum value of all the attributes and the other seed contains the maximum attribute values. These maximum and minimum seeds undergo reproduction and fill the population space.

3.4.3 Reproduction operators

The operators used for reproduction are selection, crossover and mutation.

Selection strategy

Unlike algorithms found in the literature, in the proposed system, agents use their social traits in choosing the individuals for reproduction. The agent with the social trait of risk taking chooses rules using any of the knowledge sources at random. The cautious agents choose individuals from historical KS consisting of the elite ones, while imitators use rule schema specified by the user from the situational KS. In this way, knowledge based selection is used rather than random selection. This kind of selection strategy aids in creating not only interesting knowledge but also a diverse set of solutions using the various KS's.

Crossover

One point crossover is used. Initially two individuals are chosen at random from the population. A crossover point which is a random integer whose value is less than the size of the chromosome is chosen at random and the contents of the chromosome after the crossover point are swapped. Crossover produces two children. For example Figure 1 shows parents chosen for cross over and the two children produced by cross over.

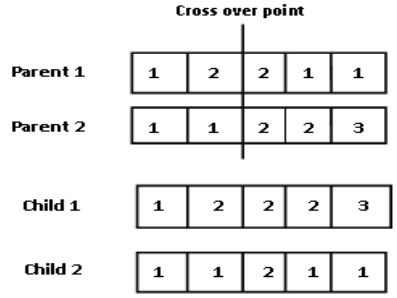

Figure 1 Individuals before and after crossover

Mutation

Mutation operates on individual values of attributes in the chromosome. A mutation point is chosen similar to that of the crossover point which is a random integer whose value is less than the chromosome size. The value of the attribute at that point is replaced by another value depending upon the type of the value. For nominal and/or discrete attributes the value to be replaced is chosen at random from a list of available values from NKS. If the value is continuous, a random value in a specified range of minimum and maximum values so far encountered is generated and used for reproduction. A list of values for discrete and nominal attributes and lower and upper bound for real valued attributes is stored in the normative knowledge source. Mutation is illustrated in Figure 2.

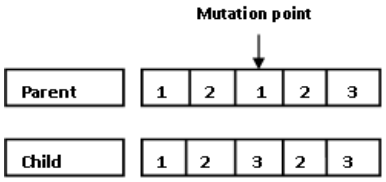

Figure 2 Parent and Child created by mutation

3.4.4 Parameters

The parameters that are to be considered and greatly influence the algorithm performance are the crossover rate and the mutation rate. Also the population size and the number of generations or the termination condition are parameters of importance.

Crossover rate
There are a variety of issues that have been discussed in the literature regarding the crossover rate. Experiments have been carried out using varying crossover rates ranging from 0% to 100% but hardly any optimum value has been reported. In the current study experiments are conducted by varying the crossover rates from 0% to 100% in steps of 20.

Mutation rate
Mutation rate is the rate at which mutation occurs in a generation. A low mutation rate of 1% and a high mutation rate of 20% can be found in the literature. However there are discussions about varying the mutation rate as the algorithm proceeds. Further experimentation is required for tuning this parameter since it depends upon various factors like the number and type of attributes in the data instances, the representation of the chromosomes in the solution space, and more. In the current study experiments are conducted by varying the mutation rates from 0% to 100% in steps of 20. High mutation rates are used in experiments since the chromosome representation uses high level encoding which enables mutation to act directly on individual attribute values.

Population size:
Population sizes ranging from a few dozens to hundreds have been reported. Population size can be varied depending upon the size of the data source. In the proposed study the population size is taken as 200.

Stopping criteria
Stopping criteria can be set to a certain number of generations, or it can be set by the user, where the user can stop the algorithm at a point where a satisfactory set of rules have been obtained. Another condition which can be used as stopping criteria is coverage of all the records in the train data set. The algorithm stops when all the records in the train data set have been covered by at least a single rule known as the sequential covering approach. In the current study number of generations specified by the user is used as the stopping criteria.

3.5 Optimization strategy/Fitness evaluation

The optimization or multi objective optimization strategy forms the acceptance phase of the cultural algorithm. The ultimate objective of multi-objective algorithms is to guide the user's decision making, through the provision of a set of solutions that have differing trade-offs between the various objectives [24], and thus the user must be involved in the process of discovering rules . Therefore in the proposed system the user is allowed to control the system by specifying most of the attributes of the system including the rule metrics (objectives), the rule schema, and other parameters as discussed earlier. The user can choose any combination of metrics including coverage, support, confidence, interest, surprise, precision, recall/sensitivity, specificity and a difference measure that stores the difference between the rule and the user specified schema. Coverage and Confidence have been used in the current study. Pareto optimality and ranking composition methods are the frequently used optimization strategies. In the current study Pareto optimality has been used as the optimization strategy to select elite individuals.

Pareto optimality is an optimization strategy that uses comparison of the metrics represented as a vector. An individual "A" is said to be better than another individual "B" if "A" is better than "B" in all the metric values or equal to "B" in all but one metric and better at least in one metric value. This is enabled by the use of Domain KS which stores the rule metrics as fitness vectors. The entries in the DKS are compared with each other and the best performers in all the metrics are

returned as dominators. The dominators form the Pareto front found in the Historical KS at the end of the algorithm execution.

4. EXPERIMENTS AND RESULTS

4.1 Methodology

The CAT-CRM works as follows:

1. The data set specified by the user is loaded and training ($2/3^{rd}$ of data set) and test data ($1/3^{rd}$ of data set) sets are created.
2. The rule metrics, the rule schema and the evolutionary parameters of population size, cross over rate, mutation rate, number of generations and the agent parameters namely the number of agents of each type is obtained from the user. In the current study Coverage and Confidence are used as metrics to be used in optimization of individuals and are calculated as follows. For an individual R let A be the set of all data instances whose antecedent is same as R and C the set of all instances whose consequent equal to R. Let |S| be the cardinality of any set S. Then the coverage, support and confidence of R are defined as follows :
   a. $$\text{Coverage}(R) = \frac{|A \& C|}{a.\ |C|} \quad (1)$$
   b. $$\text{Confidence}(R) = \frac{|A \& C|}{a.\ |A|} \quad (2)$$
3. The data instances in train and test data sets are converted into chromosome.
4. The NKS is updated to hold the list of values of nominal and discrete attributes and minimum and maximum values of numerical attributes.
5. The Population is initialized using individuals where one seed contains the minimum value of all the attributes and the other the maximum value of the attributes. These maximum and minimum attribute chromosomes undergo cross over and mutation to create the initial population.
6. The initial population is pruned to choose rules which are consistent with the training data set and the others are removed from the population.
7. The initial population is evaluated to find the values for the user specified metrics and RKS, DKS and TKS are updated as defined in section 3.
8. The best rules which are dominators are chosen using the Pareto optimality strategy and are added to the HKS.
9. The agents use the knowledge in the KS's to create new generations of individuals. Agents use their social traits to choose rules from different KS's for reproduction.
10. The individuals produced by the agents are evaluated on the user specified metrics and the belief space is updated. The selection, reproduction, rule evaluation and updating of belief space form a generation.
11. When termination condition reaches, HKS contains the dominators in the user specified metric values. The algorithm evaluates the rules on test data, presents the rules and rule metrics to the user and stops.

4.2 Experiments

A variety of experiments have been conducted, for studying the influence of various parameters of the algorithm, for testing the algorithm performance on multiple objectives chosen by the user and to study the effect of number and type of agents in creating rules with different properties. In the current study experiments have been carried out for studying the influence of *crossover rate* and *mutation rate* in producing rules with high accuracy. The other parameters were kept constant and are summarized in Table 1.

Table 1 Parameters

Parameters	**Values**
Crossover rate	0% to 100% in steps of 10
Mutation rate	0% to 100% in steps of 10
Stopping criteria	No. of generations
Population size	200
Initialization process	Maximum, Minimum seeds
Optimization strategy	Pareto optimality
Metrics	Coverage, Confidence

Experiments were carried out by varying the crossover rate and mutation rate from 0% to 100% in steps of 20. The number of unique rules created by the algorithm (RKS), the number of rules in the HKS (elite individuals), the time taken by the algorithm and the predictive accuracy of the returned rules on test data were observed. The algorithm was run ten times for each parameter setting keeping other parameters constant as in Table 1.

4.3 Data set

Experiments have been conducted on various data sets from the UCI Machine Learning Repository [25]. In the current study results on the Iris data set, Mammography data set and Ljubljana data set are reported. The Iris data set is a classic data set for testing classification systems consisting of 150 data instances with 5 attributes. The first four attributes are continuous and represent the Sepal-length, Sepal-width, Petal-length and Petal-width of the Iris flower. The class attribute is the class of the flower and takes three values Iris setosa, Iris versicolor and Iris virginica represented by IS, IV and IVG respectively in the current algorithm implementation. There are fifty instances each of the three classes of iris flower. The continuous attributes were each divided into intervals using simple equal width binning and discretized by assigning numerical values to each interval. Table 2 provides the intervals and the discrete values assigned to the attribute values.

Table 2 Attributes, Intervals and discrete values used in CAT-CRM

Attribute	**Intervals**	**Values**
Sepal length	(-, 5.5), (5.6, 6.8), (6.9, -)	1, 2, 3
Sepal width	(-, 2.8), (2.9, 3.7), (3.8, -)	1, 2, 3
Petal length	(-, 3.0), (3.1, 5.0), (5.1, -)	1, 2, 3
Petal width	(-, 0.8), (0.9, 1.7), (1.8, -)	1, 2, 3
Class: Iris flower	Iris setosa, Iris versicolour, Iris virginica	IS, IV, IVG

4.4 Results

Table 3 gives the average over ten runs of the number of unique rules created by the algorithm as in RKS, the number of rules in HKS namely the dominators, the time taken and the predictive accuracy against the different crossover rates while Table 4 summarizes the algorithm performance for different mutation rates.

Table 3 Performance summary for crossover rates

Crossover rate%	Measures	RKS	HKS	Time (milliseconds)	Accuracy%
0	Avg	56.7	4.1	1515	91.6
	Stdev	6.33	2.64	83.93	2.07
	Min	49	1	1401	88
	Max	66	9	1702	94
20	Avg	64.2	3.5	1700.9	91.4
	Stdev	4.83	2.51	202.03	2.50
	Min	59	1	1367	86
	Max	74	8	2007	94
40	Avg	68.8	5.2	1798.5	93.2
	Stdev	7.73	3.94	276.36	4.02
	Min	51	1	1487	88
	Max	77	12	2252	100
60	Avg	81.8	5.2	1650.8	93.4
	Stdev	7.83	2.62	238.03	2.84
	Min	72	2	1358	90
	Max	94	11	2034	96
80	Avg	92.9	3.6	1633.1	**94.4**
	Stdev	7.08	1.9	258.08	**1.99**
	Min	84	1	1285	92
	Max	**109**	**6**	2139	**100**
100	Avg	92.2	4.1	**1425.4**	93.2
	Stdev	8.32	3.81	130.23	3.66
	Min	82	1	1139	90
	Max	106	11	1590	100

Best values marked in bold face

Since CAT-CRM performs multi-objective optimization, several rules are returned in each run. Fig 3 shows sample Pareto front for best crossover rate and Fig 4 for best mutation rates.

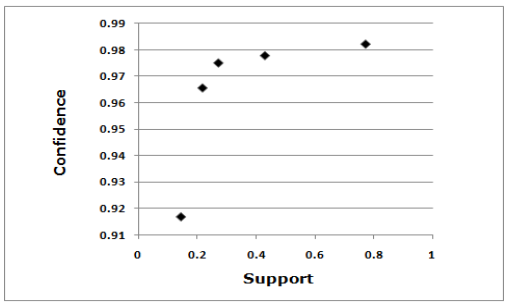

Figure 3 Sample Pareto front for best crossover rate

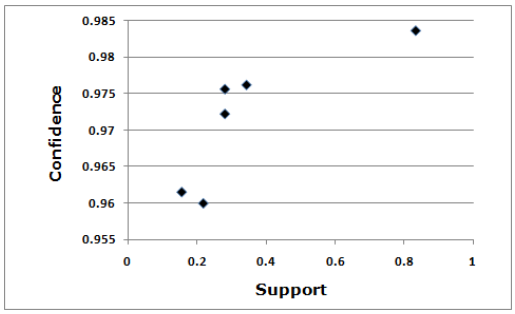

Figure 4 Sample Pareto front for best mutation rate

Fig 5 gives a comparative view of the distribution of accuracies for different crossover rates while Fig 6 for the different mutation rates using box plots.

Table 4 Performance summary for mutation rates

Mutation rate%	Measures	RKS	HKS	Time (milliseconds)	Accuracy%
0	Avg	60.30	3.10	1302.20	83.60
	Stdev	4.79	2.69	134.37	29.45
	Min	52	0	1145	0
	Max	66	8	1486	96
20	Avg	76.30	5.30	1513.50	**94.60**
	Stdev	5.54	2.63	220.43	1.90
	Min	71	2	1099	92
	Max	85	10	1880	98
40	Avg	84.60	3.80	1629.30	92.20
	Stdev	7.28	1.69	184.19	**1.48**
	Min	67	2	1413	90
	Max	92	7	1977	94
60	Avg	85.30	5.60	1628.50	93
	Stdev	6.70	4.35	306.90	3.43
	Min	74	1	1240	88
	Max	95	14	2127	98
80	Avg	104.30	3.30	1883.80	92.60
	Stdev	6.95	3.30	244.70	2.50
	Min	94	1	1470	88
	Max	119	**5**	2263	**98**
100	Avg	107.80	5.3	1864.50	92.20
	Stdev	9.94	3.77	184.92	3.82
	Min	90	1	1661	86
	Max	**122**	13	2291	98

Best values marked in bold face

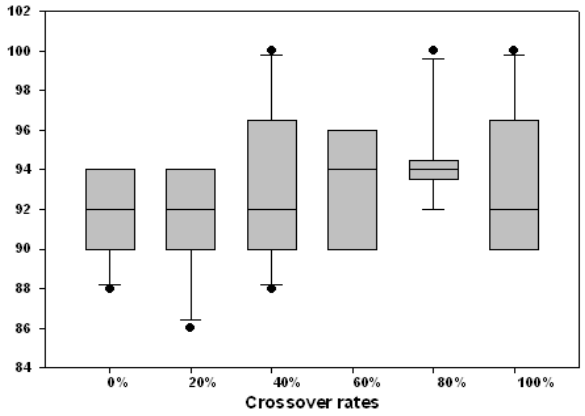

Figure 5 Spread of accuracies for different crossover rates

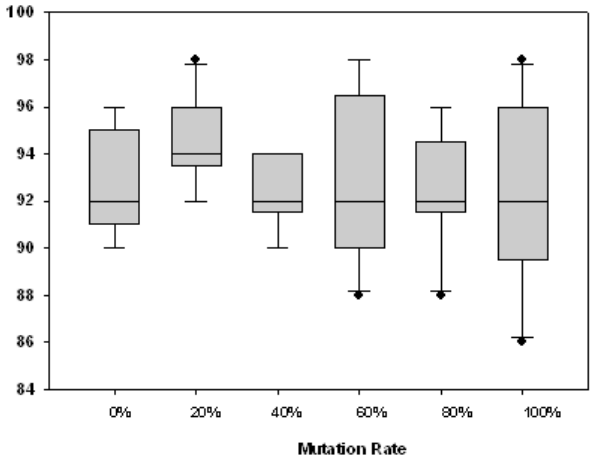

Figure 6 Spread of accuracies for different mutation rates

4.5 Discussion

Mining rules with desirable properties as specified by the user is considered as a multi objective optimization problem. The proposed Cultural Algorithm Toolkit uses a genetic algorithm as the evolutionary strategy and Pareto optimality as the optimization strategy in an extended Cultural Algorithm framework to solve the problem of classification rule mining to present the user with a good set of rules. The agents in the CAT-CRM use the knowledge sources to select and produce good rules using collective social intelligence which are updated at each generation and used in successive generations.

As can be observed from Table 3 maximum average accuracy was obtained with a crossover rate of 80%. The standard deviation is 1.99 which is also smaller compared to other crossover rates. A minimum accuracy of 92% and a maximum of 100% are achieved with this crossover rate. It is also interesting to observe that as the crossover rate increases the number of unique rules created by the algorithm as found in RKS also increases but starts decreasing after the rate of 80%. This observation may be due to the fact that the algorithm starts producing more repeated rules at a crossover rate of 100%. Moreover the accuracy starts decreasing after the rate of 80%. Also a compact rule set with 6 rules and 100% accuracy is obtained at this crossover rate which further confirms its optimality. Another interesting observation is that the average time taken is the least with 1425.4 milliseconds when the crossover rate is 100% which could not be explained and needs further exploration.

From Table 4 it can be observed that a high accuracy of 94.60 is obtained with a mutation rate of 20% with an average of 5.3 rules and an average time of 1513.5 milliseconds. But the standard deviation is higher with a value of 1.90 as compared to 1.48 at 40%. Again it can be noted that higher mutation rates produce more number of unique rules with a maximum of 122 rules with 100% mutation rate and a compact set of 5 rules with maximum accuracy of 98% is observed with 80% mutation rate. 98% accuracy is also observed when the mutation rates are 60, 80 and 100%. However high standard deviation suggests inconsistency in producing accurate rules at these rates. An interesting observation is that when the mutation rate is 0 the minimum number of rules in HKS is 0 leading to 0% accuracy. This further confirms the influence of mutation in producing better solution sets. As for the time taken, as the mutation rate increases, the time taken by the algorithm also increases with a maximum of 2291 milliseconds at 100% mutation.

From Fig 3 and Fig 4 it can be observed that the algorithm is able to choose best rules with high support and confidence. Fig 5 gives an overview of the distribution of accuracies for different crossover rates. The box plot for the crossover rate of 80% has a smaller and even distribution indicating the consistency of the algorithm in producing rules with good accuracy at this rate. The maximum of 100% accuracies is achieved at crossover rates of 40, 80 and 100%. However for 40% and 100% the distribution is not even indicating inconsistency in producing accurate rules.
Fig 6 gives an overview of the spread of accuracies for different mutation rates. The 0% accuracy was removed before plotting since it distorted the comparison of accuracies for different mutation rates. There is not much difference in distribution of accuracies for the different mutation rates as indicated by the box plots. However for 20% mutation rate the distribution is much even as compared to other mutation rates.

Thus it can be concluded that a crossover rate of 80% and a mutation rate of 20% produces good set of rules. But if it will be the case for other data sets with different sizes and dimensions is to be further explored. The results obtained in the current study are encouraging in producing good rules according to user specified metrics, i.e., multi objective optimization of rules. It further proves the influence of crossover rates and mutation rates in producing accurate rules.

4.6 Performance analysis

4.6.1 Complexity Analysis

The complexity of the algorithm is $O(n^2)$, where n is the number of individuals in the rule KS. The individuals in the RKS are compared with each other to find the difference while updating TKS. The topographical KS stores these values as Rule Id pairs along with their difference. Again the values in DKS are compared with each other to update HKS. This again depends upon the number of rules created during each generation and thus depends upon the number of rules in RKS. Thus the complexity of the algorithm is $O(n^2)$ where n is the number of individuals in the RKS. However addition of a procedure to ensure that the initial population consisted only of individuals that are consistent with the training data reduced the number of unnecessary rules in the initial population and thus the time taken for evaluation. The individuals in the initial population were used by agents for creating individuals in future generations which naturally would be better than keeping all the rules in the initial population. This in turn reduced the number of generations required for the algorithm to converge and in producing better rules and improving the classification accuracy on the test data.

4.6.2 Multi Objective optimization of rules

The user is allowed to select from a set of nine metrics including coverage, support, confidence, specificity, sensitivity, precision, recall, an interestingness, and a surprise measure. The similarity of the individuals with the user specified rule schema is added as the tenth metric to present the user with novel rules. The algorithm evaluates individuals based on these metrics specified by the user and presents the user with rules which are good in these measures. The results discussed show promising outcomes of CAT-CRM as a Toolkit for experimenting with multi objective optimization of rules.

4.7 Contributions of the system

The proposed CAT-CRM for multi objective optimization of rules contributes in various ways to classification rule mining, evolutionary multi objective optimization and an agent based social system, which are listed below.

i. In multi objective rule mining systems found in the literature there are two phases namely the rule induction phase and the rule pruning or rule optimization phase. But in the CAT-CRM proposed in this work individuals are created and immediately evaluated. Only

rules with good fitness values are retained thus reducing the overhead cost of processing low fit individuals.

ii. Most of the rule induction systems found in the literature are sub-group discovery systems meaning that the algorithm has to be run as many times as there are classes. This is avoided in the current CAT-CRM. Parallel rule induction is made possible to create rules simultaneously for all the classes.

iii. The proposed system is an Interactive Knowledge discovery system in that, the users can control the various parameters of the system and experiment with various attributes of the system. Moreover the system can be applied to any data set irrespective of the application domain. The user is provided with choices to choose almost all the parameters of the system. Thus the system can be used as a test suite to study rule mining system, evolutionary system and/or an agent based social computing system.

iv. Whitacre [1] states that nature-inspired meta-heuristics (NIM) are becoming increasingly utilized for optimization problems in academia and industry. Use of Meta heuristic for rule discovery using knowledge sources which are found in most animal species is enabled through CA for solving optimization problems. These knowledge sources direct the agents to discover a diverse set of novel and interesting rules.

The outcome of the system is Optimized rules mined by informed decisions and strategies used by agents with various cognitive traits which work cooperatively in an evolving environment modeled using cultural algorithm.

5. CONCLUSION

In the present study a cultural algorithm toolkit is proposed for interactive e knowledge discovery applied to the problem of multi objective optimization of classification rules. The tool provides a GUI through which the user can input values for different parameters. The proposed system allows the user to control rule parameters, evolutionary parameters as well as agent parameters so that the user can study the influence of various parameters on the outcome. Solutions with differing trade-offs is presented to the user from which the user can choose the best. Also the outcome of the system is a set of optimized rules which are tangible and thus can be used to evaluate the evolutionary and agent based components of the system in an efficient way. The CA enables incorporating knowledge in a systematic and principled manner into evolutionary algorithms which are blind search methods. Also incorporation of intelligent agents with cognitive traits has enabled integration of intelligent agent technology with data mining with the use of cultural algorithm, so that the system can also be used as a social system to study the dynamics of an organization or any real world social system.

References

[1] Whitacre J M (2011) Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. Computing, 93:121–133, DOI: 10.1007/s00607-011-0154-z

[2] Reynolds, R G (1994) An introduction to cultural algorithms. In: Proceedings of the 3rd Annual Conference on Evolutionary Programming, World Scientific: River Edge, NJ, 131–139.

[3] Kendall Graham and Su Yan, (2007) Imperfect Evolutionary Systems. IEEE Transactions on Evolutionary Computation, 11(3): 294-307

[4] Baykasoglu A, Ozbakir L (2007) MEPAR-miner: multi-expression programming for classification rule mining. Eur J Oper Res 183:767–784.

[5] Freitas Alex A., (2007) A Review of Evolutionary Algorithms for Data Mining. Soft Computing for Knowledge Discovery and Data Mining, 79-111

[6] De la Iglesia B, Philpott M S, Bagnall A J, Rayward-Smith V J (2003) Data mining rules using multi-objective evolutionary algorithms. In: Proceedings of 2003 IEEE congress on evolutionary computation, 1552–1559.

[7] Freitas AA (2004) A critical review of multi-objective optimization in data mining: a position paper. SIGKDD Explor 6(2):77–86

[8] Srinivasan Sujatha and Sivakumar Ramakrishnan, Evolutionary multi objective optimization for rule mining: a review. Artificial Intelligence Review, 36(3): 205-248, DOI: 10.1007/s10462-011-9212-3.

[9] De la Iglesia B, Reynolds Alan, Rayward-Smith Vic J (2005) Developments on a multi-objective meta-heuristic (MOMH) algorithm for finding interesting sets of classification rules. In: Proceedings of third international conference on evolutionary multi-criterion optimization, EMO2005, LNCS 3410, Springer, Berlin, 826–840

[10] Reynolds A. P. and de la Iglesia, B., (2006) Rule induction using multi-objective meta-heuristic: Encouraging rule diversity. In: Proceedings of IJCNN 2006, 6375-6382.

[11] Reynolds A. P. and de la Iglesia B., (2009) A Multi-Objective GRASP for Partial Classification. Soft Computing, 13(3) :227-243.

[12] Dehuri S. and Mall R., (2006) Predictive and comprehensible rule discovery using a multi-objective genetic algorithm. Knowledge-Based Systems, 19:413–421.

[13] Giusti Rafael, Gustavo E A, Batista P A, Prati Ronaldo Cristiano, (2008) Evaluating Ranking Composition Methods for Multi-Objective Optimization of Knowledge Rules, In: Proceedings of Eighth International Conference on Hybrid Intelligent Systems, 537-542.

[14] Zhao H (2007) A multi-objective genetic programming approach to developing Pareto optimal decision trees. Decis Supp Syst 43:809–826

[15] Reynolds A P, Corne David W, De la Iglesia B (2009) A multi-objective grasp for rule selection. In: Proceedings of the 11th annual conference on genetic and evolutionary computation, GECCO'09, Montréal Québec, Canada, pp 643–650

[16] Khabzaoui M, Dhaenens C, Talbi EG, (2008) Combining evolutionary algorithms and exact approaches for multi-objective knowledge discovery. RAIRO Operations Research, 42:69–83.

[17] Casillas J, Orriols-Puig A, Bernad-o-Mansilla E (2008) Toward evolving consistent, complete, and compact fuzzy rule sets for classification problems. In: Proceedings of 3rd international workshop on genetic and evolving fuzzy systems, Witten-Bommerholz, Germany, 89–94

[18] Casillas J, Pedro Martinez AE, Benitez Alicia D (2009) Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems. Soft Comput 13:419–465.

[19] Reynolds R G, Bin Peng, and Mostafa Ali, (2007) The Role of Culture in the Emergence of Decision-Making Roles, An Example using Cultural Algorithms. Complexity, Wiley Periodicals, Inc., 13(3): 27-42.

[20] Sternberg M and Reynolds R G, (1997) Using cultural algorithms to support re-engineering of rule-based expert systems in dynamic environments: A case study in fraud detection. IEEE Trans. Evol. Comput., 1(4):225–243.

[21] Lazar Alina and Reynolds R.G., (2002) Heuristic, Heuristics and Optimization for Knowledge Discovery. Vol., 2 (ed. Ruhul A. Sarker, Hussein A. Abbass, and Charles S. Newton) by Idea Group Publishing, USA.

[22] Reynolds, R.G., Mostafa Ali, and Thaer Jayyousi, (2008) Mining the Social Fabric of Archaic Urban Centers with Cultural Algorithms. IEEE Computer, 64-72.

[23] Reynolds, R.G. and Mostafa Z. Ali, (2007), Exploring Knowledge and Population Swarms via an Agent-Based Cultural Algorithms Simulation Toolkit (CAT). IEEE congress on evolutionary computing (CEC 2007), pp.2711-2718

[24] Reynolds A. P. and de la Iglesia B., (2007) Rule Induction for Classification Using Multi-Objective Genetic Programming. In: proceedings of 4th Int'l. Conf. on Evolutionary Multi-Criterion Optimization. LNCS 4403, 516-530.

[25] Newman D., Hettich S., Blake C. and Merz C., (1998) UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, Univ. of California at Irvine, http://www.ics.%20uci.edu/~mlearn/MLRepository.html.

Authors

Sivakumar Ramakrishnan is Reader and Head of the Research Department of Computer Science in AVVM Sri Pushpam College, Tamil Nadu, India since 1987. His research interests include Data mining, Human Computer Interaction and Bio-informatics. He has published a number of papers in National and International Journals. He received his PhD in Computer Science from Barathidasan University, India in the year 2005.

Sujatha Srinivasan received her Master's degree in Mathematics in 1993 and Master's degree in Computer Applications in 2000. She received her Master of Philosophy in Computer Science in 2004. She is Assistant Professor in the PG and Research department of Computer Science in Cauvery College for women, Tamil Nadu, India for the past eleven years. She is a Research scholar in AVVM Sri Pushpam College, India. Her research interests include Simulation modeling, Data mining, Human Computer Interaction and Evolutionary computing. She has published papers in International Journals and presented papers in International Conferences.