Rule Based Classification System for Medical Data Mining Using Fuzzy Ant Colony Optimization

Mostafa Fathi Ganji, Mohamad Saniee Abadeh

ABSTRACT

Pattern recognition systems have been extensively used in healthcare to mine patient data to discover a predictive model that makes reliable predictions. The objective of this paper is to utilize a hybridization of Ant Colony Optimization (ACO) and Fuzzy Logic to extract a set of fuzzy rules for classification of medical data. We will comment on some recent works and describe a new and efficient approach that leads us to significant results for medical classification problems, named Fc-AntMiner. Fc-AntMiner generates a set of fuzzy classification rules from labeled data by using an ACO-based learning algorithm. These rules are represented in linguistic forms that are easily interpreted and examined by users. The results reveal that Fc-AntMiner outperforms several famous methods in classification accuracy for medical classification.

Keywords

Fuzzy Classification, Ant Colony Optimization, Medical Data Mining, Pattern Recognition.

1. INTRODUCTION

Many diseases such as diabetes and heart disease show symptoms which some of these symptoms are appear in other diseases, i.e. many diseases have share symptoms [11]. Therefore, in addition to the evaluation of test results, physicians must pay attention to previous decisions which made for patients in the same conditions. In other words, the physician needs both knowledge and experience for proper decision making. Therefore, with regard to importance of problem (life application) and to ensure make fast, accurate and meaningful decisions, classification systems (or pattern recognition systems) could be used.

In the classification task, the goal is to assign each case (record, sample) to one class, out of a set of predefined classes, based on the values of some attributes (called predictor attributes) [1].

Fuzzy sets theory, introduced by Professor Zadeh [14], can improve such classification and decision support systems by defining overlapping class definitions [5]. The main advantages of representation by fuzzy rules are their partially overlapping properly and their powerful capabilities to handle uncertainty and vagueness [5]-[7]. The application of fuzzy if-then rules also improves the interpretability of the results and provides more insight into the classifier structure and decision making process [7]. Moreover, a case therefore, can belong to several classes with different degrees of membership. The fuzzy classification rules are represented in linguistic forms that are easily interpreted and examined by users [6].

Also, there has been an increasing interest in imitating living beings to develop powerful algorithms for hard optimization problems. Ant Colony Optimization (ACO), Introduced by Dorigo [2], is a heuristic algorithm with efficient local search for combinatorial problems [15]-[16]. ACO imitates the behavior of real ant colonies in the nature to search for food and to connect to one another by pheromone laid on paths traveled [15]. This algorithm has been developed as a probabilistic search algorithm for a large scale optimization problem that arises frequently in real applications [19].

Several methods have been proposed to produce fuzzy if-then rules. Ishibuchi *et al*. [8]-[9], introduced GA-based algorithms for selecting a small number of significant fuzzy if–then rules from a larger candidate rule base to construct a fuzzy classifier system with high classification accuracy. Roubos *et al.* [10] proposed a powerful fuzzy modeling scheme with complexity reduction. Parpineli *et al* [4] used the ACO for fuzzy rule learning for the first time and after that multiple works have been presented which have used the ACO to generate classification rules [3],[23],[24]. Saniee *et al.* [13], [25] used the GA-based algorithm to produce the fuzzy rules. They applied ACO algorithm to local search and improve the total classification accuracy.

Fc-AntMiner utilizes artificial ants in order to explore the search space and gradually make candidate fuzzy rules. We have tried to balance the cooperation and competition between the ants in a manner that the ants are encouraged to help the colony to find more accurate rules. After the extraction of fuzzy rules, the final classification system can be employed to classify medical data.

The proposed method has been tested using several medical data sets available at the University of California, Irvine web site [26]. These data sets are: Pima Indian Diabetes, Wisconsin Breast Cancer, Lung Cancer, BUPA Liver and Heart Disease. The results show that this algorithm can classify the medical data sets with considerable accuracy which is competitive or even better than the results achieved by earlier works.

The rest of the paper is as follow: Ant Colony Optimization is presented in section 2. Section 3 is devoted to proposed method. Experimental results are reported in Section 4 and Section 5 is conclusions.

Mostafa Fathi Ganji is with Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Nasr Bridge, Jalal Ale Ahmad Highway, TEHRAN, IRAN (email: m.ganji@modares.ac.ir)

Mohamad Saniee Abadeh is with Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Nasr Bridge, Jalal Ale Ahmad Highway, TEHRAN, IRAN (email: saniee@modares.ac.ir)

2. ANT COLONY OPTIMIZATION (ACO)

Ant algorithms are based on the cooperative behavior of real ant colonies, which are able to find the shortest path from a food source to their nest [2]. While walking, real ants deposit a chemical substance called pheromone on the ground. Ants can smell pheromone and when choosing their way, they tend to choose, in a probabilistic way, paths marked by strong pheromone concentrations [19]. In the absence of pheromone, ants choose paths randomly. Pheromone is evaporated over time, therefore, in shorter paths pheromone evaporation is less in comparison with longer paths and causes more pheromone accumulation in the shorter routes [15]. This positive feedback effect means that because of more pheromone all the ants will eventually use the shortest path [16]. Although a single ant is capable of building a solution (i.e., a path), the optimal solution comes about solely as a result of the cooperative behavior of the ant colony (which is based on a simple form of indirect communication through the pheromone, called stigmergy). Although the first ACO algorithm, called Ant System, was applied to solve the TSP problem [7], a large number of applications to other problems were proposed after the introduction of ant system. Recently, the ACO metaheuristic was proposed as a common framework for existing applications [19]. Each ant builds a possible solution to the problem by moving through a finite sequence of neighbor states (nodes). Moves are selected by applying a stochastic local search directed by the ant internal state, problem-specific local information and the shared information about the pheromone [2].

3. PROPOSED METHOD

FC-ANTMINER operates in two main stages: Training Stage and Testing Stage. In training stage, at first, the ACO algorithm is applied to generate a set of fuzzy rules via training patterns. These fuzzy rules are displayed as the following form:
Rule Rj:

If x_1 *is* $A_{j,1}$ *and … and* x_n *is* $A_{j,n}$ *, then Class* C_j *with CF=*CF_j.

Where R_j is the label of the *jth* fuzzy if–then rule, $A_{j1},\ldots, A_{jn}$ are antecedent fuzzy sets on the unit interval [0,1] (each triple <attribute, operator, value> called a term), C_j is the consequent class (i.e., one of the given *c* classes), and CF_j is the grade of certainty of the fuzzy If–then rule R_j. The antecedents of each fuzzy rule are presented in the form of typical set of linguistic values as figure 1. The membership function of each linguistic value in figure 1 is specified by homogeneously partitioning the domain of each attribute into symmetric triangular fuzzy sets. We use such a simple specification in computer simulations to show the high performance of our fuzzy classification system, even if the membership function of each antecedent fuzzy set is not tailored. However, we can use any tailored membership functions in our fuzzy classifier system for a particular pattern classification problem.

ACO learns the rules associated to each class separately. Therefore, if we have c classes then we will have c rule sets, each one corresponding to its related class. All these rule sets make our final classification system. Figure 2 shows the training stage of Fc-AntMiner.

3.1 Fuzzy Rule Generation by ACO

The ACO algorithm applies the artificial ants to explore among the training patterns and gradually deriving fuzzy rules. The ants learns the rules related to each class separately, that is, for each

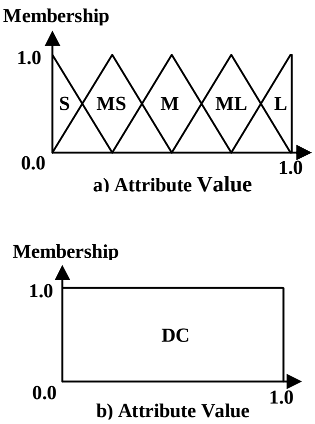

Figure 1. The used antecedent fuzzy sets in this paper. a) 1: Small, 2: medium small, 3: medium, 4: medium large, 5: large. b) 0: don't care.

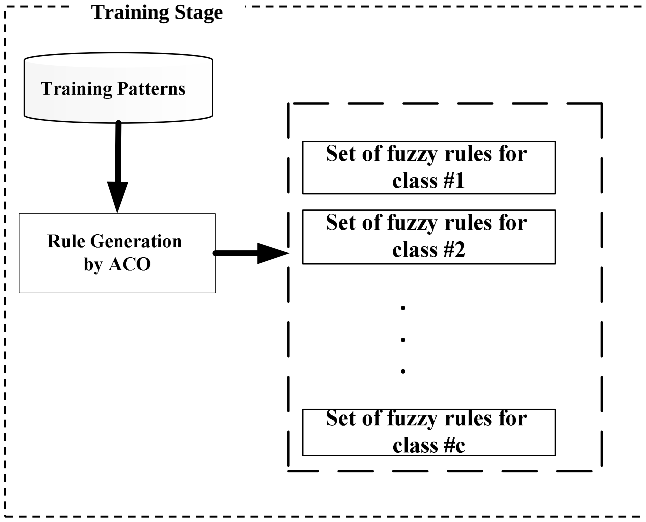

Figure 2. The ACO algorithm is applied to generate a set of fuzzy rules related to each class separately, via training samples.

class such as k a function is called to learn the corresponding fuzzy rules. Figure 3 shows the stages of fuzzy rule generation by ACO. At first, the output rule set is empty and to learn the fuzzy rules associated with each class, Rule-Learner function (figure 4) is called. This function learns the fuzzy rules corresponding to each class and returns them to main learning algorithm. All of the learned rules for each class could be used as our final classification system.

In Rule-Learner function (figure 4), the list of discovered rules is empty and the training set consists of all the training cases. In outer loop of Rule-Learn, the pheromone is initialized in a way that all cells in the pheromone table are initialized according to equation (1) [4]:

$$\tau_{i,j}(t=0)=\frac{1}{\sum_{i=1}^{a} b_i} \qquad (1)$$

Then, the first ant (ant_0) constructs rule R_j randomly by adding one term (each triple <attribute, operator, value> called a term) at a time and in the next iterations (t≥1) the ants modify rule R_j. The maximum terms that each ant can modify in each iteration (t≥1) is

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

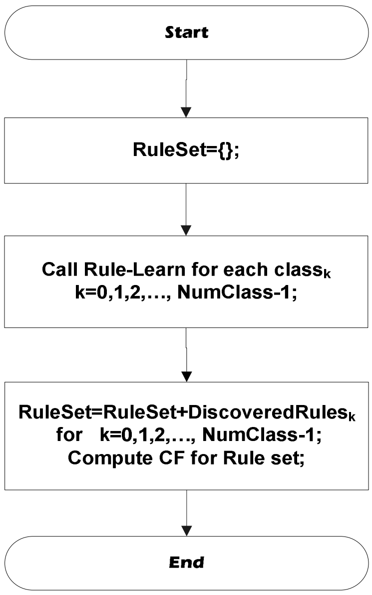

Figure 3. The stages of fuzzy rule learning by ACO

1- J=0;
2- TrainingPatterns= {all training samples};
3- DiscoverdRules$_k$= {};
4- Repeat
    4.1- t=0, Set Pheromone Table;
    4.2-Update Heuristic matrix (H_k);
    4.3- Ant$_t$ makes Rule R_j;
    4.4- While (t<Max_ants)
        4.4.1- Ant$_t$ modifies R_j;
        4.4.2- Computing the quality of R_j;
        4.4.3- Updating the pheromone;
        4.4.4- t++;
    4.5-The ant that has the best modification augments its rule to DiscoverdRules$_k$;
    4.6- Remove the cases correctly covered by the selected rule from the TrainingPatterns;
    4.7- J++;
5- Until (Stopping Conditions are satisfied)
6- Return DiscoverdRules$_k$;
7- End.

Figure 4. Rule_Learner Function.

determined by a parameter named Max_change. Max_change can be the number of attributes at most.

Each ant chooses term$_{i,j}$ to modify (or add to current rule in the first iteration) with following probability:

$$\mathbf{P_{i,j}} = \frac{\tau_{i,j}(t).\eta_{i,j}}{\sum_i^a \sum_j^{b_i} \tau_{i,j}(t).\eta_{i,j}}, \forall i \in I \tag{2}$$

Where, $\eta_{i,j}$ is a problem-dependent heuristic value for term$_{ij}$. The function that defines the problem-dependent heuristic value will be discussed in section (3.1.1)

$\tau_{i,j}$ is the amount of pheromone currently available(at time t) on the path between attribute i and value j.

a and b_i are the total numbers of values in the domain of attribute$_i$ and is the total number of attributes respectively. I is the set of attributes that are not yet used by the ant

The number of ants that modify the rule R_j in inner loop of FC-AntMiner is determined by user-defined parameter, named Max_ants.

While a rule modified by an ant, the quality function calculates the quality of modified rule. The quality of a rule such as R_j is computed according to equation (5) [13].

$$f_p = \frac{\sum_{k|c^k=c_i} w^k \mu_{R_i}(x^k)}{\sum_{k|c^k=c_i} w^k} \tag{3}$$

$$f_p = \frac{\sum_{k|c^k \neq c_i} w^k \mu_{R_i}(x^k)}{\sum_{k|c^k \neq c_i} w^k} \tag{4}$$

$$Q = w_p f_p - w_n f_n \tag{5}$$

Where

F_P: rate of positive training samples covered by the rule R_i.

F_N: rate of negative training samples covered by the rule R_i

W^k: a weight which reflects the frequency of instance x^k in the training set.

W_P: the weight of positive classification.

W_N: the weight of negative classification.

After each ant modifies the terms of a rule according to Max-change parameter, pheromone updating is carried out. We have defined a new simple method to update pheromone, in a way that whenever each ant modified the terms of rule R_j, quality of rule R_j is calculated. If the quality of rule R_j is increased then pheromone of this rule is increased according to value of quality that has been improved. Our experiments have shown that by this new update strategy, in each iteration, the pheromone helps the ants to improve the quality of rule effectively. Pheromone updating is carried out according to equation (7).

$$\Delta_Q = Q_i^{\text{After Modification}} - Q_i^{\text{Befor Modification}} \tag{6}$$

$$\tau_{i,j}(t+1) = \tau_{i,j}(t) + \tau_{i,j}(t).(\Delta_Q.C) \tag{7}$$

Where

Δ_Q shows difference between the quality of the rule R_i after and before modification. C is a parameter to regulate influence of improved quality to increase the pheromone (in our experiment C is 0.5).

It is necessary to decrease the pheromone of terms that have not participated in the construction of rules. For this purpose, pheromone evaporation is simulated. To simulate the pheromone evaporation in real ant colony, the amount of pheromone associated with each term$_{ij}$ that does not occur in the constructed rule must be decreased.The pheromone of unused terms is decreased by dividing the amount of the value of each τ_{ij} by the summation of all τ_{ij} [4].

Stopping condition in outer loop of Rule-Learner (figure 4) function refers to any condition that user has defined to terminate the loop. In our experiments, when minimum uncovered instances are remained or fairly all ants travel in the same path, the learning process is finished.

For each class independently, the above operations will be done iteratively and finally a set of rules would be discovered. These rules could be used as our classifier.

3.1.1 Heuristic Information

The ants modify the terms of the rules according to heuristic information and amount of pheromone. Fc-AntMiner has used a set of two-dimensional matrixes, named H, as heuristic information so that H= $\{H_1, H_2, \ldots, H_{numClass}\}$. For each class such as k we have matrix H_k which rows and columns indicate the attributes and fuzzy values respectively. H_k shows the distribution of the values of data set in class k. Each time a rule is discovered and correctly covered patterns are removed, set H is updated and it won't be changed in the inner loop of Rule-Learn function. Therefore, the computation overhead to calculate the heuristic information is significantly reduced. Also these matrixes help the ants to choose more relevant terms and make strong rules. Each member of set H such as H_k is represented as follow:

$$H_k = \begin{array}{c c} & \begin{array}{cccccc} DC & S & MS & M & ML & L \end{array} \\ \begin{array}{c} Att_1 \\ Att_2 \\ \vdots \\ \\ Att_{n-1} \end{array} & \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} & h_{1,6} \\ h_{2,1} & & & & & \\ \vdots & & & \ddots & & \vdots \\ h_{n-1,1} & & & & & \\ h_{n,1} & h_{n,2} & h_{n,3} & h_{n,4} & h_{n,5} & h_{n,6} \end{bmatrix} \end{array}$$

Fig.5.Matrix H_k shows the distribution of data set values (class k)

An entry of matrix H_k such as $h_{i,j}$ (j>1) shows the number of uncovered patterns that labelled with class k and value of attribute$_i$ is equal to jth fuzzy value, j=1,2,...,6 (DC is *1th* fuzzy value, S is *2th* fuzzy value, …).

First column ($h_{i,1}$ i=1,2,..,n) includes the don't care (DC) probability of features. Saniee *et al* [14] used a constant value to determine DC probability of the features value which was calculated with trial and error. FC-ANTMINER to determine the don’t care value of each attribute such as attribute$_i$ uses the uniformity measurement of domain values of attribute$_i$. The lesser value of DC shows the more uniformity distribution of the attribute values are and vice versa. So, if DC value of attribute$_i$ is equal to 1 and 0, it means that the domain values of attribute$_i$ have completely uniform distribution and none-uniform distribution respectively. For each attribute, DC value is measured in terms of the entropy. Therefore, the first column of matrixes in set H is updated by the following equations:

$$E_{i,1} = -\sum_{j=2}^{6} P(h_{i,j}|sumh_i).\log P(h_{i,j}|sumh_i), i = 1,2,\ldots,n \quad (8)$$

$$h_{i,1}=1-E_{i,1} \quad (9)$$

Where

$sumh_i$ is the summation of none-DC values of attribute$_i$ and p ($h_{i,j}|sumh_i$) is the empirical probability of observing the $h_{i,j}$.

It is essential to normalize the entries of matrixes set H to facilitate its use in Equation (5). The following normalization function has been applied to normalize the matrixes entry:

$$\eta_{i,j} = \frac{h_{i,j}}{Max(h_{i,i}),\ \forall\, i = 1,2,3,\ldots,n} \quad (10)$$

Where $Max(h_{i,j})$ is maximum value in column $_j$ and $\eta_{i,j}$ is heuristic information (this value is used in equation (5)).

3.2 Fuzzy Inference

Let us assume that our pattern classification problem is a *c*-class problem in the *n*-dimensional pattern space with continuous attributes. We also assume that *M* real vectors $x_p= (x_{p1}, x_{p2}, \ldots, x_{pn}), p=1,2,\ldots,m$ are given as training patterns from the *c* classes ($c<<M$).

When ACO-learning algorithm, corresponding to each class, generated a set of fuzzy rules using M patterns, a fuzzy inference engine is needed to classify test patterns (figure 6). For this purpose, certainty grade must be computed. The following steps are applied to calculate the certainty grade of each fuzzy if-then rule: [6]

Step 1: Calculate the compatibility of each training pattern $x_p= (x_{p1}, x_{p2}, \ldots, x_{pn})$ with the fuzzy if–then rule R_j by the following product operation:

$$\mu_j(x_p) = \mu_{j1}(x_{p1}) \times \ldots\ \mu_{jn}(x_{pm}), p = 1,2,3,\ldots,m \quad (11)$$

Where $\mu_{ji}(x_{pi})$ is the membership function of ith attribute of pth pattern and M denotes the total number of patterns.

Step 2: For each class, calculate the relative sum of compatibility grades of the training patterns with the fuzzy if–then rule R_j:

$$\beta_{Classh}(R_j) = \sum_{x_p \epsilon Classh} \frac{\mu_j(x_p)}{N_{Classh}}, \quad h = 1,2,\ldots,c \quad (12)$$

Where $\beta_{Classh}(R_j)$ is the sum of the compatibility grades of the training patterns in *Classh* with the fuzzy if–then rule R_j and N_{Classh} is the number of training patterns which their corresponding class is i.

Step 3: The grade of certainty CF_j is determined as follows [13]:

$$CF_j = \frac{\beta_{Class\,\hat{h}}(R_j) - \bar{\beta}}{\sum_{h=1}^{c} \beta_{Class\,h}(R_j)} \quad (13)$$

Where

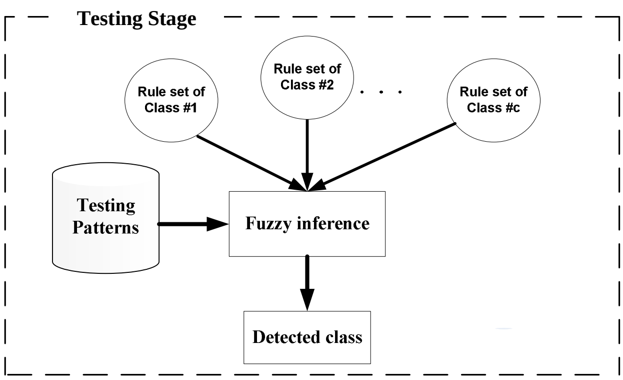

Figure 6. the testing Stage

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\bar{\beta} = \frac{\sum_{h\neq \widehat{h_j}} \beta_{Class\ h}\,(R_j)}{(c-1)} \qquad (14)$$

Now, we can specify the certainty grade for any combination of antecedent fuzzy sets. The task of our fuzzy classifier system is to generate combinations of antecedent fuzzy sets for generating a rule set S with high classification ability. When a rule set S is given, an input pattern $x_p = (x_{p1}, x_{p2}, \dots, x_{pn})$ is classified by a single winner rule R_j in S [6], which is determined as follows [6]:

$$\mu_j(x_p).CF_j = max\{\mu_j(x_p).CF_j | R_j\} \qquad (15)$$

That is, the winner rule has the maximum product of the compatibility and the certainty grade CF_j.

4. EXERIMENT RESULTS

For evaluating performance of Fc-AntMiner, five data sets from UCI data repository [26] such as Pima Indian Diabetes (Pima), Wisconsin Breast Cancer (Wisconsin), Lung Cancer (Lung), BUPA Liver (BUPA) and Heart Disease (Heart) were used (table.1).

Table 1. Data Sets Description

Data Set	Instances	Attributes	classes
Lung	32	56	3
Pima	768	8	2
Wisconsin	699	10	2
BUPA	345	7	2
Heart	270	13	2

We have normalized the data sets, where each numerical value in the data set is normalized between 0.0 and 1.0. For this purpose, the below function is applied to normalize the data set.

$$Normalize(X) = \frac{(X - X_{min})}{(x_{max} - x_{min})} \qquad (16)$$

Table 2 shows parameters specification that we have used in our computer simulations for Fc-AntMiner.

Comparative performance of Fc-AntMiner is evaluated using ten-fold cross-validation test [1] which data set is divided into ten partitions, and Fc-AntMiner is run ten times, using a different partition as test set each time, with the other nine as training set.

The classification rate is being calculated according to equation (17).

$$Classification\ Rate = \frac{(TP+TN)}{(TP+TN+FN+FP)} \qquad (17)$$

Where

TP: true positives, the number of cases in our training set covered by the rule that have the class predicted by the rule.

FP: false positives, the number of cases covered by the rule that have a class different from the class predicted by the rule

FN: false negatives, the number of cases that are not covered by the rule but that have the class predicted by the rule.

TN: true negatives, the number of cases that are not covered by the rule and that do not have the class predicted by the rule.

Table 2. Parameter specification in computer simulation

Parameter	Description		Value
Max_ant	Maximum ants that can modify a rule in each iteration		30
W^k	the frequency of instance x^k in the training set		1
W_P	the positive weight classification		0.3
W_n	the negative weight classification		0.7
Max_change	Specifies the maximum term per rules that ants can modify. Since each dataset has different number of attributes, we have set this parameter for each data set separately.	Lung	12
		Pima	3
		Wisconsin	4
		BUPA	3
		Heart	8

Also, precision measures of how many of the correctly classified samples are positive samples and Recall measures the of how many of positive sample are correctly classified. Precision and Recall are computed by following equations:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (18)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (19)$$

Precision and Recall stand in opposition to one another [22]. As precision goes up, recall usually goes down (and vice versa). F-Measure is a trade-off between Precision and Recall. It is the harmonic-mean of Precision and Recall and takes account of both measures. It is computed according to equation (20).

$$\text{F} - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (20)$$

Tables 2-6 show the mean classification rate, Precision, Recall and F-Measure for the generated rules by Fc-AntMiner and several well-known methods.

Table 3. Wisconsin Breast Cancer data set.

Method	Classification Rate	Precision	Recall	F-Measure
SVM	0.967096	0.967	0.967	0.967
NaiveBayes	0.959943	0.962	0.96	0.96
C4.5	0.945637	0.946	0.946	0.946
NN	0.958512	0.959	0.959	0.959
KNN	0.951359	0.951	0.951	0.951
Decision Table	0.95422	0.954	0.954	0.954
BayesNet	0.959943	0.962	0.96	0.96
Fc-AntMiner	**0.97511**	**0.978**	**0.978**	**0.978**

Table 4. Pima Indian Diabetes data set.

Method	Classification Rate	Precision	Recall	F-Measure
SVM	0.77343	0.769	0.773	0.763
NaiveBayes	0.76302	0.759	0.763	0.76
C4.5	0.73828	0.735	0.738	0.736
NN	0.753906	0.75	0.754	0.751
KNN	0.70182	0.696	0.702	0.698
Decision Table	0.71224	0.706	0.712	0.708
BayesNet	0.74349	0.741	0.743	0.742
Fc-AntMiner	**0.83333**	**0.848**	**0.831**	**0.839**

Table 4. Lung cancer data set.

Method	Classification Rate	Precision	Recall	F-Measure
SVM	0.481481	0.466	0.481	0.471
NaiveBayes	**0.703704**	0.704	0.704	0.704
C4.5	0.48148	0.342	0.481	0.400
NN	0.518519	0.528	0.519	0.522
KNN	0.481481	0.421	0.481	0.444
Decision Table	0.518519	0.494	0.519	0.502
BayesNet	0.592593	0.575	0.593	0.58
Fc-AntMiner	**0.62963**	**0.736**	**0.732**	**0.739**

Table 5. BUPA Liver data set.

Method	Classification Rate	Precision	Recall	F-Measure
SVM	0.582609	0.757	0.583	0.432
Naïve Bayes	0.565217	0.623	0.565	0.555
C4.5	0.686957	0.683	0.687	0.680
NN	**0.715942**	**0.714**	**0.716**	**0.711**
KNN	0.628986	0.63	0.629	0.629
Decision Table	0.576812	0.561	0.577	0.558
BayesNet	0.562319	0.535	0.562	0.522
Fc-AntMine	**0.678261**	**0.675**	**0.678**	**0.675**

Table 6. Heart Disease data set.

Method	Classification Rate	Precision	Recall	F-Measure
SVM	0.84074	0.841	0.841	0.840
NaiveBayes	0.83333	0.833	0.833	0.833
C4.5	0.76666	0.766	0.767	0.767
NN	0.78148	0.784	0.781	0.782
KNN	0.75185	0.753	0.752	0.752
Decision Table	**0.84814**	**0.848**	**0.848**	**0.848**
BayesNet	0.81111	0.811	0.811	0.811
Fc-AntMiner	**0.796296**	**0.818**	**0.817**	**0.817**

It can be seen Fc-AntMiner has the considerable results and its performance is competitive with famous methods such as SVM, NN.

The following table shows the mean number of effective features in each data set. Fc-AntMiner detected these attributes by using the new heuristic information which we have proposed

(Section 3.1.1). The other features are recognized as don't care features.

Table 7. Effective features.

Data set	Number of attributes	Number of effective Attributes
Lung	56	**18.354**
Pima	8	**3.240**
Wisconsin	10	**4.179**
BUPA	7	**3.196**
Heart	13	**7.2**

Also, table 8 shows the computation time to build each classifier.

Table 8. time taken to build classifier

Data Set	Non-Parallel	Parallel
Wisconsin	6.72	3.74
Pima	10.78	6.64
Bupa	11.45	7.93
Heart	5.73	3.01
Lung	4.69	2.69

5. CONCLUSION

This paper presents a mixture of Ant Colony Optimization and Fuzzy Logic to medical classification, named Fc-AntMiner. In training stage, we have proposed a cooperative ACO algorithm for fuzzy rule learning. Then in testing stage, a fuzzy inference engine is used to classify test patterns. The main new features of the presented algorithm are as follows:

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1- Introducing a new framework for learning the rules in such a way that the rules related to each class are learned independently. Therefore, the search space divided to multiple subsections (class) and the ant learn the rules associated to each class separately and fuzzy rules cab be learned in parallel way.

2- We have proposed a two dimensional matrix to recognize the DC attributes which uses an entropy function.

3- There are two important concepts in metaheuristic procedures that are: Competition and Cooperation. The previous versions of AntMiner paid more attention to Competition and this caused some of the rules was very strong while the other rules was nearly weak. In this paper we have paid attention to cooperation in order to produce a set of nearly strong rules. For this propose, we have encouraged the ants to have more cooperation in the body Fc-AntMiner function.

For our experiments we have used several data sets from UCI repository [26], which are: Pima Indian Diabetes, Wisconsin Breast Cancer, Lung Cancer, BUPA Liver and Heart Disease. The results show that Fc-AntMiner has considerable performance for medical classification and its results is competitive or even better than some well-known algorithms such as SVM, Neural Network, C4.5, KNN, Naïve Bayes.

REFERENCES

[1] R.S. Michalski, I. Bratko, M. Kubat, Machine Learning and Data Mining: Methods and Applications, Wiley, New York, 1998.

[2] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.

[3] David Martens, Manu De Backer, Raf Haesen, Jan Vanthienen, Monique Snoeck, and Bart Baesens, Classification With Ant Colony Optimization, *IEEE Trans* on Evolutionary Computaion, Vol. 11, pp.651-656, 2007.

[4] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Trans* on Evolutionary Computaion vol.6, pp. 321–332, 2002

[5] Hisao Ishibuchi, Tomoharu Nakashima, and Tadahiko Murata, Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems, *IEEE Trans* on Systems, PART. B, VOL. 29, NO. 5, OCTOBER.1999.

[6] Ken Nozaki, Hisao Ishibuchi, and Hideo Tanaka, Adaptive Fuzzy Rule- Base classification systems, *IEEE Trans* on Fuzzy Systems, VOL. 4, NO. 3, AUGUST.1996.

[7] Yuhui Shi, Russell Eberhart and Yaobin Chen, Implementation of Evolutionary Fuzzy Systems, *IEEE Trans* on Fuzzy Systems, VOL. 7, NO. 2, APRIL.1999.M. Young, the Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[8] Hisao Ishibuchi, Takashi Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets and Systems, VOL.141 pp.59–88, 2004.

[9] Hisao Ishibuchi, Ken Nozaki, Naohisa Yamamoto, and Hideo Tanaka, Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE T*rans on Fuzzy Systems, VOL. 3, NO. 3, pp.260-271, AUGUST 1995.

[10] Hans Roubos, and Magne Setnes, Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction, *IEEE T*rans on Fuzzy Systems, VOL. 9, NO. 4, pp.516-524, AUGUST 2001.

[11] Kemal Polat, Salih Gu¨nes, Ahmet Arslan, A cascade learning system for classification of diabetes disease: Generalized Discriminant Analysis and Least Square Support Vector Machine, Expert Systems with Applications vol.34, pp.482–487, 2008.

[12] Hamid Mohamadi, Jafar Habibi, Mohammad Saniee Abadeh, Hamid Saadi, Data mining with a simulated annealing based fuzzy classification system, Pattern Recognition vol.41 pp.1824 – 1833, 2008.

[13] Mohammad Saniee Abadeh, Jafar Habibi, and Emad Soroush, Induction of fuzzy classification systems via evolutionary ACO-based Algorithms, International journal of simulation, systems, science, technology, VOL. 9, NO.3, 2008.

[14] L.A. Zadeh, Fuzzy sets, Information and Control, Vol.8, pp.338-353, 1965.

[15] Marco Dorigo, Christian Blum, Ant colony optimization theory: A survey, Theoretical Computer Science Vol.344, pp. 243 – 278, 2005.

[16] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Trans* on Systems, Man and Cybernetics, Vol.26, pp.1-13, 1996.

[17] Urszula Boryczka, Finding groups in data: Cluster analysis with ants, Applied Soft Computing, Vol. 9, pp.61–70, 2009.

[18] Esin Dogantekin, Akif Dogantekin, Derya Avci, Levent Avci, An intelligent diagnosis system for diabetes on Linear Discriminant Analysis and Adaptive Network Based Fuzzy Inference System: LDA-ANFIS, Digital Signal Processing, pp1–8, 2009.

[19] Christian Blum, Review Ant colony optimization: Introduction and recent trends, Physics of Life Reviews Vol.2, pp. 353–373, 2005.

[20] Brazdil, and Torgo, knowledge acquisition via knowledge integration, in current trend in knowledge accusation, IOS press, 1990.

[21] Carpenter, G. A., & Markuzon, N, ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. Neural Networks, Vol.11, pp.323–336, ,1998.

[22] Ian H. Witten, Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, vol.1, Morgan Kaufmann publications, pp. 363-483, 2005.

[23] B. Liu, H. A. Abbass, and B. McKay, Classification rule discovery with ant colony optimization, In Proc. *IEEE/WIC* Int. Conf. Intell. Agent Techno, 2003.

[24] B Liu, HA Abbass, B McKay, Density-based heuristic for rule discovery with ant-miner, The 6th Australia-Japan joint workshop on intelligent, 2002.

[25] Mohammad Saniee Abadeh, Jafar Habibi, Emad Soroush, Induction of Fuzzy Classification Systems Using Evolutionary ACO-Based Algorithms, Proceedings of the First Asia International Conference on Modelling & Simulation (AMS'07), *IEEE*, 2007.

[26] UCI Machine Learning Repository, http://www.archive.ics.uci.edu/ml/datasets.html (last accessed: March 2010).

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)