Classification Rule Discovery with Ant Colony Optimization and Improved Quick Reduct Algorithm

P. Jaganathan[1*], K.Thangavel[2] A. Pethalakshmi[3], M. Karnan[4]

[1] Department of Computer Applications, PSNA College of Engineering & Technology, Dindigul-624 622, Tamil Nadu, India. E-mail: jaganathodc@gmail.com

[2] Department of Computer Science, Periyar University, Salem, Tamil Nadu-636011, India. E-mail: drktvelu@yahoo.com.

[3]Department of Computer Science, M.V.M Government Arts College (W), Dindigul-624 001, Tamil Nadu,. India.E-mail: pethalakshmi@yahoo.com

[4]Deprtment of Computer science and Applications, Gandhigram Rural Institute- Deemed University, Tamil Nadu, India. E- Mail: karnanme@yahoo.com

Abstract - Ant colony optimization (ACO) algorithms have been applied successfully to combinatorial optimization problems. More recently, Parpinelli et al have applied ACO to data mining classification problems, where they introduced a classification algorithm called Ant Miner. In this paper, we present a system that combines both the proposed Improved Quickreduct algorithm for data preprocessing and ant miner. The proposed system was tested on standard data set and its performance is better than the original Ant Miner algorithm.

Index Terms- Ant Colony Optimization(ACO), Quick Reduct, Improved Quick Reduct Algorithm, Classification.

1 INTRODUCTION

1.1 DATA MINING

Data mining refers to extracting knowledge from large amounts of data. Data mining is often treated as synonym for another popularly used term, Knowledge Discovery in Databases (KDD)[5]. Data cleaning, Data integration, Data selection, Data transformation, data mining, Pattern evaluation and Knowledge presentation are the important steps of KDD. Data Mining is one of the steps in KDD and is defined as the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns of interest in data.

Classification is a supervised learning and is one of the most studied data mining technique. The main goal is to predict the class $C_i = f(x_1,\ldots,x_n)$, where $x_1,\ldots x_n$ are input attributes. There is one distinguished attribute called as dependant attribute. The input to the classification algorithm is a data set of training records with several attributes.

1.2 ROUGH SET BASED FEATURE REDUCTION

In 1982, Pawlak introduced the theory of Rough sets [14, 15]. A Rough Set is a mathematical tool to deal with Uncertainty and vagueness of an information system. An information system can be presented as a table with rows analogous to objects and columns analogous to attributes. Each row of the table contains values of particular attributes representing information about an object.

Using the Rough Sets approach, one can deal with two major problems in the analysis of an information system: (i) Reducing unnecessary objects and attributes so as to get the minimum subset of attributes, ensuring a good approximation of classes and an acceptable quality of classification. (ii) Representing the information system as a decision table which shows dependencies between the minimum subset of attributes(called conditions) and particular class numbers(called decisions), without redundancy.

Let U be any finite universe of discourse. Let R be any equivalence relation defined on U. Clearly, the equivalence relation partitions U. Here, (U, R) which is the collection of all equivalence classes, is called the approximation space. Let W_1, W_2, W_3, ..., W_n be the elements of the approximation space (U, R). This collection is known as knowledge base. Then for any subset A of U, the lower and upper approximations are defined as follows:

$$\underline{R}A = \cup\{W_i / W_i \subseteq A\}$$

$$\overline{R}A = \cup\{W_i / W_i \cap A \neq \varnothing\}$$

(Advance online publication: 13 February 2007)

The ordered pair ($\underline{R}A$, $\overline{R}A$) is called a rough set. Once defined these approximations of A, the reference universe U is divided into three different regions: the positive region $POS_R(A)$, the negative region $NEG_R(A)$ and the boundary region $BND_R(A)$, defined as follows:

$$POS_R(A) = \underline{R}A$$
$$NEG_R(A) = U - \overline{R}A$$
$$BND_R(A) = \overline{R}A - \underline{R}A$$

Hence, it is trivial that if BND(A) = Φ, then A is exact. This approach provides a mathematical tool that can be used to find out all possible reduces.

A decision table may have more than one reduct. Anyone of them can be used to replace the original table. Finding all the reducts from a decision table is NP-Hard [11]. Fortunately, in many real applications it is usually not necessary to find all of them, one is sufficient. A natural question is which reduct is the best if there exist more than one reduct. The selection depends on the optimality criterion associated with the attributes. If it is possible to assign a cost function to attributes, then the selection can be naturally based on the combined minimum cost criteria. In the absence of an attribute cost function, the only source of information to select the reduct is the contents of the data table. For simplicity, we adopt the criteria that the best reduct is the one with the minimal number of attributes and that if there are two or more reducts with the same number of attributes, then the reduct with the least number of combinations of values of its attributes is selected.

1.3 LITERATURE REVIEW

Besides the introduction given here, the extensive literature of Rough sets theory can be referred to Orlowska [12], Peters et al. [16], Polkowski et al.[17] for recent comprehensive overviews of developments.

Hu et al.[6] developed two new algorithms to calculate core attributes and reducts for feature selection. These algorithms can be extensively applied to a wide range of real-life applications with very large data sets. Jensen et al.[7, 8, 9] developed the Quickreduct algorithm to compute a minimal reduct without exhaustively generating all possible subsets and also they developed Fuzzy-Rough attribute reduction with application to web categorization.

Zhong et al.[19] applies Rough Sets with Heuristics(RSH) and Rough Sets with Boolean Reasoning(RSBR) are used for attribute selection and discretization of real-valued attributes. Komorowski et al.[10] studies an application of rough sets to modeling prognostic power of cardiac tests. Carlin et al.[3] presents an application of rough sets to diagnosing suspected acute appendicitis.

The rest of the paper is organized as follows: Section 2 briefs about the data sets used and data preparation for this study. Section 3 describes the feature reduction algorithm using Improved Quickreduct Algorithm and its implementation. Section 4 describes the Ant Colony Algorithm in the context of classification. Results are discussed in Section 5 and the paper is concluded in Section 6.

2 DATA PREPARATION

The medical data sets viz., New-Thyroid, Pima, Wisconbreastcancer, Lung Cancer and Dermatology obtained from UCI machine learning repository [2]. Also we have collected and used real HIV data set are used for this study. The HIV database consists of information collected from the HIV Patients at Voluntary Counseling and Testing Centre (VCTC) of Government Hospital, Dindigul District, Tamilnadu, India, a well-known centre for diagnosis and treatment of HIV. The advantage of this data set is that it includes a sufficient number of records of different categories of people affected by HIV. The set of descriptors presents all the required information about patients. It contains the records of 500 patients. The record of every patient contains 49 attributes and this has been reduced to 22 attributes after consulting the Physician. The continuous attributes of all the data sets are discretized before applying it to the Ant Miner.

3. IMPROVED QUICKREDUCT ALGORITHM (IQR)

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. The problem of finding a reduct of an information system has been the subject of much research in [1, 18]. The most basic solution to locate such a subset is to simply generate all possible subsets and retrieve those with a maximum rough set dependency degree. Obviously, this is an expensive solution to the problem and is only practical for very simple datasets. Most of the time only one reduct is required as, typically, only one subset of features is used to reduce a dataset, so all the calculations involved in discovering the rest are pointless.

The Quickreduct algorithm [7,8,9] attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in the rough set dependency metric, until this produces its maximum possible value for the dataset. According to the Quick reduct algorithm, the dependency of each attribute is calculated, and the best candidate is choosen. This, however is is not guaranted to find a minimal subset as has been shown in [4]. Using the dependency function to discriminate between candidates may lead the search down a non-minimal path. It is impossible to predict which combinations of attributes will lead to an optimal reduct based on changes in dependency with the addition or deletion of single attributes. It does result in a close-to-minimal subset, though, which is still useful in greatly reducing dataset dimensionality. In [4], a potential solution to this problem has been proposed whereby the Quickreduct algorithm is altered, making it into an n-lookahead approach. However, even this cannot guarantee a reduct unless n is equal to the original number of attributes, but this reverts back to generate-and-test. It still suffers from the same problem as the original Quickreduct, i.e. it is impossible to mention at any stage whether the current path will be the shortest to a reduct.

The Quickreduct algorithm is improved herein and the pseudo code of the Improved Quickreduct Algorithm is given below:

Improved Quickreduct (C,D)
C, the set of all conditional features;
D, the set of decision features.
(a) $R \leftarrow \{\}$
(b) $\gamma_{best} = 0$, $\gamma_{prev} = 0$
(c) Do
(d) $T \leftarrow R$
(e) $\gamma_{prev} = \gamma_{best}$
(f) $\forall x \in C$
(g) if $\max(\gamma_{R\cup\{x\}}(D) > \gamma_{prev}$
Where $\gamma_R(D)=\text{card}(POS_R(D))$ / card(U)
$POS_R(D) = \underline{R}\,x$
(h) $T \leftarrow R\cup\{x\}$
(i) $\gamma_{best} = \gamma_T(D)$
(j) $R \leftarrow T$
(k) until $\gamma_{best} = \gamma_{prev}$
(l) return R

4 ANT COLONY OPTIMIZATION (ACO)

Ant Colony Optimization (ACO) [13] is a branch of newly developed swarm intelligence has been used for classification. Swarm intelligence is a field which studies "the emergent collective intelligence of groups of simple agents". In groups of insects, which live in colonies, such as ants and bees, an individual can only do simple tasks on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows. Most real ants are blind. However, each ant while it is walking, deposits a chemical substance on the ground called pheromone [8] of a newly developed form of artificial intelligence called swarm intelligence. Swarm intelligence is a field which studies "the emergent collective intelligence of groups of simple agents" [2]. In groups of insects, which live in colonies, such as ants and bees, an individual can only do simple tasks on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows. Most real ants are blind. However, each ant while it is walking, deposits a chemical substance on the ground called pheromone. Pheromone encourages the following ants to stay close to previous moves. The pheromone evaporates over time to allow search exploration. The goal of Ant-Miner is to extract classification rules from

4.1 ALGORITHM:A HIGH-LEVEL DESCRIPTION OF ANT-MINER

Training Set = {all training cases};
DiscoveredRuleList = [ ]; /* rule list is initialized with an empty list */
WHILE (Training Set > Max_uncovered_cases)
t = 1; /* ant index */
j = 1; /* convergence test index */
Initialize all trails with the same amount of pheromone;
REPEAT
Ant starts with an empty rule and incrementally constructs a classification rule Rt by adding one term at a time to the
current rule;
Prune rule Rt;
Update the pheromone of all trails by increasing pheromone in the trail followed by Antt proportional to the quality of Rt) and decreasing pheromone in the other trails (simulating pheromone evaporation);
IF (Rt is equal to Rt – 1) /* update convergence test */
THEN j = j + 1;
ELSE j = 1;
END IF
t = t + 1;
UNTIL (i " No_of_ants) OR (j " No_rules_converg)
Choose the best rule Rbest among all rules Rt constructed by all the ants;
Add rule R best to DiscoveredRuleList;
Training Set = Training Set - {set of cases correctly covered by Rbest};
END WHILE

4.2 PHEROMONE INITIALIZATION

All cells in the pheromone table are initialized equally as per the following equation:

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i} \tag{1}$$

where a is the total number of attributes, bi is the number of values in the domain of attribute i

Each rule in Ant-Miner contains a condition part as the antecedent and a predicted class as a consequent. The condition part is a conjunction of attribute-operator-value called terms. The operator used in all experiments is "=" since in Ant-Miner, all attributes are assumed to be categorical. Let us assume a rule condition such as $term_{ij} \approx A_i=V_{ij}$, where A_i is the i^{th} attribute and V_{ij} is the j^{th} value in the domain of A_i.

4.3 RULE CONSTRUCTION

The rule is constructed by the ant incrementally by adding one term at a time. The term selection is based on the probability as given by the equation 2.

$$P_{ij(t)} = \frac{\tau_{ij}(t).\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{bi} \tau_{ij}(t).\eta_{ij}, \forall i \in I} \tag{2}$$

where η_{ij} is a problem-dependent heuristic value for term-ij, τ_{ij} is the amount of pheromone currently available (at time t) on the connection between attribute i and value I is the set of attributes that are not yet used by the ant in the domain of attribute i

4.4 HEURISTIC VALUE

In traditional ACO, a heuristic value is usually used in conjunction with the pheromone value to decide on the transitions to be made. In Ant-Miner, the heuristic value is taken to be an information theoretic measure for the quality of the term to be added to the rule. The quality here is measured in terms of the entropy for preferring this term to the others, and is given by the following equations(3) and (4):

$$\eta_{ij} = \frac{\log_2(k) - InfoT_{ij}}{\sum_{i}^{a}\sum_{j}^{bi} \log_2(k) - InfoT_{ij}} \tag{3}$$

$$InfoT_{ij} = -\sum_{w=1}^{k}\left[\frac{freqT_{ij}w}{|T_{ij}|}\right] * \log_2\left[\frac{freqT_{ijw}}{|T_{ij}|}\right] \tag{4}$$

where k is the number of classes, |Tij| is the total number of cases in partition Tij (partition containing the cases where attribute Ai has value Vij), $freqT_{ij}^{w}$ is the number of cases in partition Tij with class w, a is the total number of attributes.

The higher the value of $infoT_{ij}$, the less likely that the ant will choose $term_{ij}$ to add to its partial rule. Immediately after the ant completes the construction of a rule, pruning is undertaken to increase the comprehensibility and accuracy of the rule. After the pruning step, the rule may be assigned a different predicted class based on the majority class in the cases covered by the rule antecedent. The rule pruning procedure iteratively removes the term whose removal will cause a maximum increase in the quality of the rule. The quality of a rule is measured using the following equation (5)

$$Q = \left[\frac{TruePos}{TruePos + FalseNeg}\right] \times \left[\frac{TrueNeg}{FalsePos + TrueNeg}\right] \tag{5}$$

where TruePos is the number of cases covered by the rule and having the same class as that predicted by the rule, FalsePos is the number of cases covered by the rule and having a different class from that predicted by the rule, FalseNeg is the number of cases that are not covered by the rule, while having the class predicted by the rule, TrueNeg is the number of cases that are not covered by the rule which have a different class from the class predicted by the rule.

E. PHEROMONE UPDATE RULE

After each ant completes the construction of its rule, pheromone updating is carried out as per the following equation(6)

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t).Q, \forall term_{ij} \in therule \tag{6}$$

To simulate the phenomenon of pheromone evaporation in real ant colony systems, the amount of pheromone associated with each $term_{ij}$, which does not occur in the constructed rule must be decreased,. The reduction of pheromone of an unused term is performed by dividing the value of each τ_{ij} by the summation of all τ_{ij}.

5 RESULTS AND DISCUSSION

Six data sets from UCI data repository [2] such as New Thyroid, Pima Indian, Wisconsin Breast Cancer, Lung Cancer and Dermatology were used for this study (table.1). The attributes are reduced using Quick Reduct and Improved Quickreduct is in table. 2. We have evaluated comparative performance of the proposed method and Ant_Miner using ten-fold cross-validation. Each Database is divided into ten partitions, and each method is run ten times, using a different partition as test set each time, with the other nine as training set.

The Quickreduct and Improved Quickreduct algorithm have been implemented using MATLAB for databases available in the UCI data repository [2] and the real HIV data set. The Comparative Analysis of Quickreduct and Improved Quickreduct algorithm is given below.(Table3.1, 3.2and 3.3)

Table 1: Data set Descrption

Data Set	Total No. Attributes	Categorical Attributes	Continuous Attributes	Classes
New-Thyroid	5	-	5	3
Pima	8	-	8	2
HIV	21	16	5	3
Wisconsin breastcancer	10	-	10	2
Lungcancer	56	-	56	3
Dermatology	34	-	34	6

Table 2: Reduced Data Sets

Data Sets	Instances	No. of attributes	Quick reduct	Improved Quick reduct
New-Thyroid	215	5	4	2
Pima	768	8	5	3
HIV	500	21	17	8
Wisconsin breast cancer	699	10	6	4
Lungcancer	32	56	8	5
Dermatology	366	34	17	6

The rule list produced by a training set is used to predict the class of each case in the test set. The accuracy rate is calculated according to equation (5) Every rule list includes a default rule, which has no condition and takes as its class the majority class in the set of training cases, so that we can apply the default rule if none of the rules in the list covers test case.

Table 3 shows accuracy rates for the rules produced by Ant_miner and the proposed system for ten runs on the two datasets. The mean accuracy rate and mean number of rule sets produced are reported in Table 4. It can be seen that Ant colony with Quick Reduct discovers somewhat more rules than Ant_Miner, but the mean accuracy of the rule sets discovered by the proposed system is higher than Ant_Miner.

Table 3.1: Test Set Accuracy Rate (%)

	New-Thyroid		Pima	
Run Number	Ant_ Miner	Ant_ Miner with improved Quick Reduct	Ant_ Miner	Ant_ Miner with Improved Quick Reduct
1	92.05	94.32	71.28	82.97
2	93.15	93.15	73.40	72.34
3	91.67	91.67	67.37	78.94
4	95.59	97.06	71.58	80.00
5	88.41	92.75	68.42	72.63
6	94.20	95.65	75.79	80.00
7	90.77	93.84	74.74	81.05
8	96.55	96.55	65.26	74.74
9	91.04	92.54	73.68	75.79
10	92.86	95.71	68.42	67.37

Table 3.2: Test Set Accuracy Rate (%)

	HIV		Wisconsin breast cancer	
Run Number	Ant_ Miner	Ant_ Miner with Improved Quick Reduct	Ant_ Miner	Ant_ Miner with Improved Quick Reduct
1	90.04	93.75	70.28	70.11
2	91.15	92.48	70.40	71.34
3	91.92	92.86	69.47	70.84
4	94.62	95.02	70.98	72.00
5	84.10	91.64	69.42	70.23
6	93.10	96.43	72.19	74.20
7	92.23	93.94	73.65	75.43
8	94.46	95.43	63.55	65.63
9	90.44	91.64	71.84	73. 84
10	94.45	94.71	69.54	68.96

Table 3.3: Test set accuracy rate (%)

	Lung cancer		Dermatology	
Run Number	Ant_ Miner	Ant_ Miner with Improved Quick Reduct	Ant_ Miner	Ant_ Miner with Improved Quick Reduct
1	91.05	90.32	74.28	82.37
2	93.15	91.15	72.40	72.44
3	91.63	90.67	66.37	77.84
4	95.49	93.66	72.58	79.00
5	87.44	87.15	67.42	70.66
6	94.22	93.66	76.79	81.60
7	90.97	89.84	73.74	82.55
8	95.55	94.55	66.26	72.64
9	92.04	91.54	74.68	76.80
10	94.86	93.71	67.44	69.45

Table 4: Mean accuracy rate and mean number of rule lists

Evaluation item	Accuracy %		No. of rules%	
Data Sets	Ant_ Miner	Ant_ Miner with Imp.Quick Red.	Ant_ Miner	Ant_ Miner with Imp.Quick Reduct
New-Thyroid	92.63	94.32	10.1	13.2
PIMA	70.99	76.58	12.2	11.3
HIV	91.65	93.79	14.26	13.1
Wisconsin breast cancer	70.13	71.26	16.28	12.25
Lungcancer	92.64	91.62	9.67	8.4
Dermatology	71.2	76.54	15.67	14.32

In practice, Ant_miner and the proposed system required almost identical running time.

6 CONCLUSION

It is demonstrated that Ant-Miner with improved Quick Reduct produces a higher accuracy rate and fewer rules than the original Ant miner algorithm. In this paper, a new method called an Improved Quickreduct, based on a variant of Quickreduct is proposed. We compared the results of Ant miner and the Ant Miner with Improved Quickreduct. The performance of the ant miner is increased when it is used with Improved Quick reduct.

REFERENCES

[1] J.J. Alpigini, J.F. Peters, J. Skowronek and N. Zhong (Eds.), "Rough Sets and Current Trends in Computing", Third International Conference, RSCTC 2002, Malvern, PA, USA, October 14-16, 2002, Proceedings. Lecture Notes in Computer Science 2475, Springer, ISBN 3-540-44274-X. 2002.

[2] C.L.Blake, C.J. Merz, UCI Repository of machine learning databases, Irvine, University of California, 1998, http://www.ics.uci.edu/~mlearn/.

[3] U.Carlin and J.Komorowski and A.Ohrn, "Rough Set Analysis of Patients with Suspected Acute Appendicitis", Proc., IPMU, 1998.

[4] J. Chouchoulas, Halliwell and Q. Shen. On the Implementation of Rough Set Attribute Reduction, Proceedings of the 2002 UK Workshop on Computational Intelligence, 18-23, 2002.

[5] J.Han and M.Kamber, "Datamining: Concepts and Techniques", Morgan Kaufmann Publishers, 1992.

[6] X. Hu, T.Y. Lin, J.Jianchao, "A New Rough Sets Model Based on Database Systems", Fundamenta Informaticae , pp.1-18, 2004.

[7] R.Jensen, Qiang Shen, "Fuzzy-Rough Attribute Reduction with Application to Web Categorization", Fuzzy Sets and Systems, Vol.141, No.3, pp.469-485, 2004.

[8] R.Jensen, Qiang Shen, "Semantics Preserving Dimensionality Reduction: Rough and Fuzzy-Rough Based Approaches", IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.12, pp. 1457-1471, 2004.

[9] R.Jensen, "Combining Rough and Fuzzy Sets for Feature Selection", Ph.D Thesis, School of Informatics, University of Edinburgh, 2005.

[10] J. Komorowski and A.Ohrn, "Modelling Prognostic Power of Cardiac tests using rough sets", Artificial Intelligence in Medicine, 15, 167-191, 1999.

[11] T.Y. Lin, N. Cercone (Eds.), "Rough sets and DataMining: Analysis of Imprecise Data", Kluwer Academic Publishers, 1997.

[12]E.Orlowska, "Incomplete Information:Rough Set Analysis", Physica-Verlag, Heidelberg, 1998.

[13] R. S. Parpinelli, H. S. Lopesand A. A. Freitas, "Data Mining with an Ant Colony Optimization Algorithm", IEEE Transactions on Evolutionary Computing, Vol. 6, No. 4, 321-332, August 2002.

[14] Z. Pawlak, "Rough Sets", International Journal of Computer and Information Sciences, Vol.11, No.5, pp. 341-356, 1982.

[15] Z.Pawlak, "Rough Sets: Theoritical Aspects and Reasoning about Data", Kluwer Academic Publishers, Dordrecht, 1991.

[16J.F.Peters and A.Skowron(eds.), "Transactions on Rough Sets 1", Springer-Verlag, Berlin, 2004.

[17]L.Polkowski, S.Tsumoto and T.Y.Lin(eds.), "Rough Set Methods and applications:New Developments in Knowledge Discovery in Information Systems", Physica-Verlag, Heidelberg, 2000.

[18]R.W. Swiniarski, and A. Skowron, " Rough Set Methods in Feature Selection and Recognition", Pattern Recognition Letters, Vol. 24, No. 6, pp. 833-849, 2003.

[[19]N.Zhong and A. Skowron, "A Rough Set-Based Knowledge Discovery Process", Int. Journal of Applied Mathematics and Computer Science., 11(3), 603-619, 2001.

P.Jaganathan –Corresponding author is Currently pursuing research in The dept. of Computer Science &Applns, Gandhigram Rural Institute, Deemed Univ., Gandhigram, Tamil Nadu, India