2008 International Symposium on Computer Science and Computational Technology

Feature Selection Based on Ant Colony Optimization and Rough Set Theory

Ming He
College of Computer Science
Beijing University of Technology
Beijing, China
heming@bjut.edu.cn

***Abstract*—Ant colony optimization (ACO) algorithms have been applied successfully to combinatorial optimization problems. Rough set theory offers a viable approach for feature selection from data sets. In this paper, the basic concepts of rough set theory and ant colony optimization are introduced, and the role of the basic constructs of rough set approach in feature selection, namely attribute reduction is studied. Base above research, a rough set and ACO based algorithm for feature selection problems is proposed. Finally, the presented algorithm was tested on UCI data sets and performed effectively.**

Keywords-rough set; ant colony optimization; feature selection; core

I. INTRODUCTION

Nowadays, accuracy, speed and interpretation are three main hot topics in current machine learning community. As a pre-processing step, feature selection is one of the techniques which can improve the accuracy, speed and interpretation at the same time. In pattern recognition field, feature selection can eliminate the redundant features to improve the classification rate. In data mining field, feature selection can remove many irrelevant features and improve the speed and accuracy of data mining. Feature selection can select the most relevant features to help understand the problems from chemistry, medicine and biology fields. In general, feature selection is a key technique to improve the accuracy, speed and interpretation of intelligent systems.

Rough set theory was introduced by Pawlak [1, 2]. It has been recognized as a powerful mathematical tool for data analysis and knowledge discovery from imprecise and ambiguous data, and has been successfully applied in a wide range of application domains such as machine learning, expert system, pattern recognition, etc. Reduct is the most important concept in rough set application to data mining. A reduct is the minimal attribute set preserving classification accuracy of all attribute of original dataset. Finding a reduct is similar to feature selection problem. Unfortunately, it has been shown that finding minimal reduct or all reducts are both NP-hard problems and there are no universal solutions. It's still an open problem in rough set theory.

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 [3, 4] and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

In this paper, we propose, a novel feature selection method by combining the global optimization ability of ant colony optimization (ACO) algorithm and the rough set theory. The proposed method is availably applied for choosing the principal features in classification problems.

II. PRELIMINARIES

In this section, we briefly introduce some preliminary results and definitions that are useful for later discussion.

A. Rough Set

In rough set theory, an information system is a 4-tuple $S=<U, A, V, f>$, where U is a is a finite set of objects, called the universe, A is a finite set of $V = \bigcup_{a \in A} V_a$ is a domain of attribute a, and f: $U \times A \to V$ is called an information function such that $f(x, a) \in V_a$ for $\forall a \in A, x \in U$. An information system S is also seen as a decision table assuming that $A = C \cup D$ and $C \cap D = \varnothing$, where C is a set of condition attributes and D is a set of decision attributes. Let $S=<U, A, V, f>$ be an information system, each $B \subseteq A$ generates an indiscernibility relation $IND(B)$ on U, which is defined as follows:

$$IND(B) = \{(x, y) \in U^2 : \forall a \in B, f(x, a) = f(y, a)\}.$$

$U/IND(B) = \{C_1, C_2, \ldots, C_K\}$ is a partition of U by B, every C_i is an equivalence class. For every $x \in U$, the equivalence class of x in relation $U/IND(B)$ is defined as follows:

$$[x]_{IND(B)} = \{y \in U : \forall a \in B, f(y, a) = f(x, a)\}.$$

Let $B \subseteq A, X \subseteq U$, the B-lower approximation of X (denoted by $B\_(X)$) and B-upper approximation of X (denoted by $B^-(X)$) are defined as follows respectively:

$$B\_(X) = \{y \mid y \in U \wedge [y]_{IND(B)} \subseteq X\},$$

$$B^-(X) = \{y \mid y \in U \wedge [y]_{IND(B)} \cap X \neq \varnothing\}.$$

978-0-7695-3498-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ISCSCT.2008.43

$B\_(X)$ is the set of all objects from U which can be certainly classified as elements of X employing the set of attributes B. $B^{-}(X)$ is the set of all objects from U which can be possibly classified as elements of X employing the set of attributes B.

Let $P,Q \subseteq A$, the positive region of classification U/IND (Q) with respect to the set of attributes P, or in short, P- positive region of Q, is defined as:

$$POS_P(Q) = \bigcup_{U/IND(P)} P\_(Q).$$

POS_P (Q) contains all objects in U that can be classified to one class of the classification U/IND (Q) by attributes P.

The dependency of Q on P is defined as:

$$\gamma_P(Q) = \frac{card(POS_P(Q))}{card(U)}.$$

An attribute a *is* said to be dispensable in P with respect to Q, if $\gamma_P(Q) = \gamma_{\{P-a\}}(Q)$; otherwise a is an indispensable attribute in P with respect to Q.

Let $S =< U, C \cup D, V, f >$ be a decision table, the set of attributes P ($P \subseteq C$) is a reduct of attributes C, which satisfies the following conditions:

$$(\gamma_P(D) = \gamma_C(D)) \wedge (\gamma_P(D) \neq \gamma_{P'}(D), \forall P' \subset P.$$

A reduct of condition attributes C is a subset that can discern decision classes with the same discriminating capability as C, and none of the attributes in the reduct can be eliminated without decreasing its discriminating capability.

The CORE is the set of attributes that are contained by all reducts, defined as: $CORE_D(C) = \bigcap RED_D(C)$, where $RED_D(C)$ is the D-reduct of C. In other words, the CORE is the set of attributes that cannot be removed without changing the positive region. This means all attributes present in the CORE are indispensable.

B. Ant Colony Optimization

Ant colony optimization (ACO) is a metaheuristic in which colonies of artificial ants cooperate in finding good solutions to discrete optimization problems. Each ant of the colony exploits the problem graph to search for optimal solutions. An artificial ant, unlike natural counterparts, has a memory in which it can store information about the path it follows. Every ant has a start state and one or more terminating conditions. The next move is selected by a probabilistic decision rule that is a function of locally available pheromone trails, heuristic values as well as the ant's memory. Ant can update the pheromone trail associated with the link it follows. Once it has built a solution, it can retrace the same path backward and update the pheromone trails. Ant system (AS) is the earliest example of this kind of algorithm. AS was first applied to solve the Traveling Salesman Problem (TSP) and it achieve encouraging results, yet not competitive with the state of the art on large problem instances. AS has been further modified and extended, and several variants have been designed. In recent years, a general framework for ant algorithms applied to combinatorial optimization has been proposed. This is called ACO metaheuristic. ACO algorithm is interplay of three procedures as described in [5].

- Construct ant solutions:

This procedure manages a colony of ants that concurrently and asynchronously visit adjacent states of the considered problem by moving through neighboring nodes of the solution space of the problem's construction graph.

The move probability distribution defines probabilities $p_{t\psi}^k$ to be equal to 0 for all moves which are infeasible(i.e., they are in the tabu list of ant k, that is a list containing all moves which are infeasible for ants k starting from state t), otherwise they are computed by means of the following formula, where α and β are user defined parameters($0 \le \alpha, \beta \le 1$):

$$p_{t\psi}^k = \begin{cases} \dfrac{\tau_{t\psi}^{\alpha} \eta_{t\psi}^{\beta}}{\sum_{t\zeta \notin tabu_k} (\tau_{t\psi}^{\alpha} + \eta_{t\psi}^{\beta})} & if\ t\psi \notin tabu_k \\ 0 & otherwise \end{cases} \quad (1)$$

In formula (1) $tabu_k$ is the tabu list of ant k, while parameters α and β specify the impact trail and attractiveness, respectively.

In ACS a new state transition rule called pseudo-random-proportional is introduced. With the pseudo-random rule the chosen state is the best with probability q_0 (exploitation) while a random state is chosen with probability 1- q_0 (exploration). Using the AS random-proportional rule the next state is chosen randomly with a probability distribution depending on η_{ij} and τ_{ij} . The ACS pseudo-random-proportional state transition rule provides a direct way to balance between exploration of new states and exploitation of a priori and accumulated knowledge. The best state is chosen with probability q_0 ($0 \leqslant q_0 \leqslant 1$) and with probability (1-q0) the next state is chosen randomly with a probability distribution based on η_{ij} and t τ_{ij} weighted by α and β.

$$s = \begin{cases} \arg\max_{ij \notin tabu_k} \{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}\} & if\ q \le q_0 \\ r & otherwise \end{cases} \quad (2)$$

where r is a random variable reference to formula (1).

- Update pheromones:

It is the process by which pheromone trails are modified. The trail value can either increase, as ants deposit pheromone on the components or connections they use, or decrease, due to pheromone evaporation. Net increase/decrease in pheromone value at a given location on trail is determined by difference of deposition and evaporation.

After each iteration t of the algorithm, i.e., when all ants have completed a solution, trails are updated by means of formula 2:

$$\tau_{t\psi}(t) = \rho \tau_{t\psi}(t-1) + \Delta \tau_{t\psi} \quad (3)$$

where $\Delta\tau_{t\psi}$ denotes the sum of the contributions of all ants that used move $t\psi$ to construct their solution, ρ ($0 \le \rho \le 1$), is a user-defined parameter called evaporation coefficient, and $\Delta\tau_{t\varphi}$ represents the sum of the contributions of all ants that used move $t\psi$ to construct their solution. The ants' contributions are proportional to the quality of the solutions achieved, i.e., the better solution is, and the higher will be the trail contributions added to the moves it used.

For instance, in the case of the TSP, moves correspond to arcs of the graph, thus state t could correspond to a path ending in node i, the state ψ to the same path but with the arc(i, j) added at the end and the move would be the traversal of arc(i, j). The quality of the solution of ant k would be the length L_k of the tour found by the ant and formula (2) would become $\tau_{ij} = \rho\tau_{ij}(t-1) + \Delta\tau_{ij}$, with

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{4}$$

where m is the number of ants and $\Delta\tau_{ij}^{k}$ is the amount of trail laid on edge(i, j) by ant k, which can be computed as:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if ant } k \text{ user arc}(ij) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where, Q is a constant parameter.

Ant Colony System (ACS) a simplified version of Ant-Q which maintained approximately the same level of performance, measured by algorithm complexity and by computational results. In ACS, the final evaporation phase is substituted by a local updating of the pheromone applied during the construction phase. Each time an ant moves from the current city to the next the pheromone associated to the edge is modified in the following way:

$$\tau_{ij}(t) = \rho \cdot \tau_{ij}(t-1) + (1-\rho)\tau_0 \tag{6}$$

where $0 \le \rho \le 1$ is a parameter and τ_0 is the initial pheromone value. τ_0 is defined as $\tau_0 = 1/n \cdot L_{nn}$, where L_{nn} is the tour length produced by the execution of one ACS iteration without the pheromone component. A similar approach was proposed with the Max-Min-AS [6] that explicitly introduces lower and upper bounds to the value of the pheromone trials.

- Daemon actions:

This procedure is used to implement centralized actions which cannot be performed by single ants.

III. ROUGH SET AND ACO BASED FEATURE SELECTION

A. Principles

Feature selection is a process of finding a subset of features, from the original set of features forming patterns in a given data set, optimal according to the given goal of processing and criterion. An optimal feature selection is a process of finding a subset $A_{opt}=\{a_{1,\text{opt}}, a_{2,apt},\ldots,a_{m,opt}\}$ of A, which guarantees accomplishment of a processing goal by minimizing a defined feature selection criterion $J_{feature}(A_{feature})$. A solution of an optimal feature selection does not need to be unique. Rough set approach to feature selection can be based on the minimal description length principle and tuning methods of parameters of the approximation spaces to obtain high quality classifiers based on selected features. We have mentioned before an example of such parameter with possible values in the power set of the feature set, i.e., related to feature selection. Other parameters can be used e.g., to measure the closeness of concepts.

In this paper, we developed two main steps to solve feature selection problems. In the first step, considering the core may be thought of as the set of necessary attributes, we use core attributes as the starting point. In the second step, the ACO is used to traverses the search space by adding other attributes in turn.

B. Algorithm Framework

The key algorithm based on rough set theory and ACO has the following structure:

Algorithm: Rough set and ACO Based Algorithm for feature selection.

Input: The decision table S= (U, C, D).

Output: The reduction of S.

Method:

1. Let $CORE = \varnothing$ and calculate $POS_C(D)$;
2. For $\forall a \in C$, calculate $POS_{(C-\{a\})}(D)$. If $POS_{(C-\{a\})} \neq POS_C(D)$, then $CORE = CORE \cup \{a\}$; Else C=C-$\{a\}$;
3. Execute iteratively step 2 until all attributes among C are calculated;
4. If $POS_{CORE}(D) = POS_C(D)$, algorithm stops and return $CORE$ as the result of feature selection; otherwise go to step 5;
5. The pheromone of each arc (i, j) is assigned to an constant, i.e., $\tau_{ij}(0) = c$;
6. Some ants (assumed the number of ants is m) are distributed to each core attribute node to conduct feature selection;
7. Each ants selects next feature node according to expression (2);
8. Calculate $POS_{CORE\cup a_i}(D), a_i \in C - CORE,$ if $POS_{CORE\cup a_i}(D) = POS_C(D)$ algorithm stops and return $FS = CORE \cup a_i$ as the result of feature selection; else go to step 9;
9. Update value of pheromone τ_{ij} for each path link and go to step 7.

IV. RESULTS OF EXPERIMENT

Our approach was implemented in C++ and experiments were conducted on a 2.00GHz Intel with 1G memory, and experiments used three data sets from UCI data set repository [7]. All datasets are discredited using ROSETTA utility [8]. In order to find whether our algorithm could find optimal reduct, we computer all reducts using bool reasoning method as described in [9] for reference. Note that when we talk about optimal reduct we refer to the shortest reduct. The leftmost column is dataset names. The 2nd, 3rd, 4th columns are instance numbers, attribute numbers and results of our algorithm for corresponding dataset. From Table 1, we can see that our algorithm found most optimal reducts successfully.

TABLE I. RESULTS OF FEATURE SELECTION

Dataset	Summary of Datasets		
	Instances	*Attributes*	*optimal*
Zoo	67	17	5
Wine	118	14	4
Vote	300	17	8
german	666	21	10
DNA	2000	181	22
Satimage	4435	37	15

V. CONCLUSIONS

In this paper, we have over-viewed some basic concepts of rough set theory and ACO. After giving a brief introduction, we have discussed the principle of a new hybrid method for feature selection. It employs core attributes in rough set as a as heuristics information and applies ACO as a global path planning algorithm. The experimental results show that the algorithm proposed in this paper was effective in attributes reduction of data sets. As further research direction, we will research the more efficient algorithm and develop a more efficient weighting mechanism for feature selection.

ACKNOWLEDGMENT

The author would like to thanks for UCI repository of machine learning databases and ROSETTA utility supporting this work. This work was funded by the Doctoral Science Foundation of Beijing University of Technology (52007011200701).

REFERENCES

[1] Z. Pawlak, Rough Sets, International Journal of Computer and Information Sciences, 1982, pp. 341-356.

[2] Z. Pawlak, Rough Sets, Theoretical Aspects of Reasoning About Data. Kluwer Academic Publisher, Dordrecht, 1991.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, The ant system: an autocatalytic optimizing process, 1991.

[4] A. Colorni, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, Proc. ECAL'91, European Conference on Artificial Life, Elsevier Publishing, Amsterdam, 1991.

[5] M. Dorigo, T Stutzle, Ant Colony Optimization. Harlow, England: Addison-Wesley, 1999.

[6] T. Stützle and H. Hoos, Improvements on the ant system: Introducing Max-Min Ant System, Proceedings of ICANNGA'97, Int. Conf. on Artificial Neural Networks and Genetic Algorithms, Springer Verlag, Vienna, 1997.

[7] C. L. Blake, C. J. Merz, UCI repository of machine learning databases, 1998. http://www. www.ics.uci.edu/pub/machine-learning-databases/.

[8] A. Ohm, J.Komorowski. Rosetta: A rough set toolkit for analysis of data , 2007. http://www.idi.ntnu.no/~aleks/rosetta/.

[9] S.K.Pal, A.Skowron, Rough Fuzzy Hybridization- A new trend in decisionmaking,Springer, 1999.