Pattern Recognition Letters 31 (2010) 226–233

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A rough set approach to feature selection based on ant colony optimization

Yumin Chen *, Duoqian Miao, Ruizhi Wang

Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China

ARTICLE INFO

Article history:
Received 5 December 2008
Received in revised form 24 August 2009
Available online 25 October 2009

Communicated by A.M. Alimi

Keywords:
Rough sets
Ant colony optimization
Feature selection
Mutual information
Data mining

ABSTRACT

Rough set theory is one of the effective methods to feature selection, which can preserve the meaning of the features. The essence of rough set approach to feature selection is to find a subset of the original features. Since finding a minimal subset of the features is a NP-hard problem, it is necessary to investigate effective and efficient heuristic algorithms. Ant colony optimization (ACO) has been successfully applied to many difficult combinatorial problems like quadratic assignment, traveling salesman, scheduling, etc. It is particularly attractive for feature selection since there is no heuristic information that can guide search to the optimal minimal subset every time. However, ants can discover the best feature combinations as they traverse the graph. In this paper, we propose a new rough set approach to feature selection based on ACO, which adopts mutual information based feature significance as heuristic information. A novel feature selection algorithm is also given. Jensen and Shen proposed a ACO-based feature selection approach which starts from a random feature. Our approach starts from the feature core, which changes the complete graph to a smaller one. To verify the efficiency of our algorithm, experiments are carried out on some standard UCI datasets. The results demonstrate that our algorithm can provide efficient solution to find a minimal subset of the features.

1. Introduction

Feature selection can be viewed as one of the most fundamental problems in the field of machine learning. The main aim of feature selection is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features (Dash and Liu, 1997). In real world problems, feature selection is a must due to the abundance of noisy, irrelevant or misleading features (Jensen, 2005). By removing these factors, learning from data techniques can benefit greatly. As Liu pointed out in (Liu and Motoda, 1998), the motivation of feature selection in data mining and machine learning is to: reduce the dimensionality of feature space, improve the predictive accuracy of a classification algorithm, and improve the visualization and the comprehensibility of the induced concepts.

In recent years, a lot of feature selection methods have been proposed. There are two key issues in constructing a feature selection method: search strategies and evaluating measures. With respect to search strategies, complete (Somol et al., 2004), heuristic (Zhong and Dong, 2001), random (Raymer et al., 2000; Lai et al., 2006) strategies were proposed. And with respect to evaluating measures, these methods can be roughly divided into two classes: classifiers-specific (Kohavi, 1994; Guyon et al., 2002; Neumann et al., 2005; Gasca et al., 2006; Xie et al., 2006) and classifier independent (Kira and Rendell, 1992; Modrzejewski, 1993; Dash and Liu, 2003). The former employs a learning algorithm to evaluate the goodness of selected features based on the classification accuracies or contribution to the classification boundary, such as the so-called wrapper method (Kohavi, 1994) and weight based algorithms (Guyon et al., 2002; Xie et al., 2006). While the latter constructs a classifier independent measure to evaluate the significance of features, such as inter-class distance (Kira and Rendell, 1992) mutual information (Yao, 2003; Miao and Hou, 2004), dependence measure (Modrzejewski, 1993) and consistency measure (Dash and Liu, 2003).

Rough set theory (RST) was proposed by Pawlak (1982), which is a valid mathematic tool to handle imprecision, uncertainty and vagueness. As an effective method to feature selection, rough sets can preserve the meaning of the features. It has been widely applied in many fields such as machine learning (Swiniarski and Skowron, 2003), data mining (Duan et al., 2007), etc. (Mi et al., 2004). The essence of rough set approach to feature selection is to find a subset of the original features. Rough set theory provides a mathematical tool that can be used to find out all possible feature subsets. Unfortunately, the number of possible subsets is always very large when N is large because there are 2^N subsets for N features. Hence examining exhaustively all subsets of features for selecting the optimal one is NP-hard. Previous methods

* Corresponding author. Tel.: +86 21 69589867; fax: +86 21 69589359.
E-mail address: cym0620@163.com (Y. Chen).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.10.013

employed an incremental hill-climbing (greedy) algorithm to select feature (Hu, 1995; Deogun et al., 1998). However, this often led to a non-minimal feature combination. Therefore, many researchers have shifted to metaheuristic, such as genetic algorithm (GA) (Wrblewski, 1995; Zhai et al., 2002), tabu search (TS) (Hedar et al., 2006) and ant colony optimization (ACO) (Dorigo and Caro, 1999; Jensen and Shen, 2003; Jensen and Shen, 2004), etc.

ACO is a metaheuristic inspired by the behavior of real ants in their search for the shortest path to food sources. Metaheuristic optimization algorithm based on ACO was introduced in the early 1990s byDorigo and Caro (1999). ACO is a branch of newly developed form of artificial intelligence called Swarm Intelligence, which studies "the emergent collective intelligence of groups of simple agents" (Bonabeau et al., 1999). ACO algorithm is inspired of ant's social behavior. Ants have no sight and are capable of finding the shortest route between a food source and their nest by chemical materials called pheromone that they leave when moving. ACO algorithm was firstly used in solving traveling salesman problem (TSP) (Dorigo et al., 1996). Then has been successfully applied to a large number of difficult problems like the quadratic assignment problem (QAP) (Maniezzo and Colorni, 1999), routing in telecommunication networks, graph coloring problems, scheduling, feature selection, etc. ACO is particularly attractive for feature selection since there is no heuristic information that can guide search to the optimal minimal subset every time. On the other hand, if features are represented as a graph, ants can discover the best feature combinations as they traverse the graph.

Since most common methods for RST-based feature selection often led to a non-minimal feature combination. In this paper we propose a novel feature selection algorithm based on rough sets and ACO, which adopts mutual information based feature significance as heuristic information for ACO. We also introduce the concept of feature core to the algorithm, by requiring that all ants must start from the core, when they begin their search through the feature space. Therefore those features near the core will be selected by the ants more quickly. The performance of our algorithm will be compared with that of RST-based algorithms and other metaheuristic-based algorithms.

This paper is organized as follows. In Sections 2 and 3, we introduce some preliminaries in rough set theory and ACO. In Section 4, we propose the approach to feature selection based on rough sets and ACO. And the pseudo-code of our algorithm is also given. Experimental results are given in Sections 5 and 6 concludes the paper.

2. Preliminary

2.1. Preliminary concepts of RST

This section recalls some essential definitions from RST that are used for feature selection. Detailed description and formal definitions of the theory can be found in (Pawlak, 1982).

The notion of information table has been studied by many authors as a simple knowledge representation method. Formally, an information table is a quadruple $\mathscr{I} = (U, A, V, f)$, where: U is a nonempty finite set of objects, A is a nonempty finite set of features, V is the union of feature domains such that $V = \bigcup_{a \in A} V_a$ for V_a denotes the value domain of feature a, any $a \in A$ determines a function $f_a : U \rightarrow V_a$,where V_a is the set of values of a.

With any $B \subseteq A$, there is an associated equivalence relation $IND(B)$:

$$IND(B) = \{(x, y) \in U \times U | \forall a \in B,\ f(x, a) = f(y, a)\}. \tag{1}$$

The partition of U, generated by $IND(B)$ is denoted $U/IND(B)$ and can be calculated as follows:

$$U/IND(B) = \otimes\{a \in B : U/IND(\{a\})\}, \tag{2}$$

where

$$R \otimes S = \{X \bigcap Y : \forall R \in X\ \forall S \in Y,\ X \bigcap Y \neq \varnothing\}. \tag{3}$$

Given an information table $\mathscr{I} = (U, A, V, f)$, for a subset $X \subseteq U$ and equivalence relation $IND(B)$, the $B - lower$ and $B - upper$ approximations of X are defined, respectively, as follows:

$$B_*(X) = \{x \in U : [x]_B \subseteq X\}, \tag{4}$$
$$B^*(X) = \{x \in U : [x]_B \bigcap X \neq \phi\}. \tag{5}$$

Let $P, Q \subset A$ be equivalence relations over U, then the positive,negative and boundary regions can be defined as

$$POS_P(Q) = \bigcup_{X \in U/IND(Q)} P_*(X), \tag{6}$$
$$NEG_P(Q) = U - \bigcup_{X \in U/IND(Q)} P^*(X), \tag{7}$$
$$BND_P(Q) = \bigcup_{X \in U/IND(Q)} P^*(X) - \bigcup_{X \in U/IND(Q)} P_*(X). \tag{8}$$

An important issue in data analysis is discovering dependencies between features. Dependency can be defined in the following way. For $P, Q \subset A$, P depends totally on Q, if and only if $IND(P) \subseteq IND(Q)$. That means that the partition generated by P is finer than the partition generated by Q. We say that Q depends on P in a degree $\mu_P(Q)$ $(0 \leqslant \mu_P(Q) \leqslant 1)$, if

$$\mu_P(Q) = |POS_P(Q)|/|U|. \tag{9}$$

If $\mu_P(Q) = 1$, Q depends totally on P, if $0 < \mu_P(Q) < 1$, Q depends partially on P, and if $\mu_P(Q) = 0$ then Q does not depend on P. Dependency degree μ can be used as heuristics in greedy algorithms to compute feature reduction.

More specially, $\mathscr{DT} = (U, C \bigcup D, V, f)$ is called a decision table if $A = C \bigcup D$ in an information table, where C is the set of condition features, D is the set of decision features. The degree of dependency between condition and decision features, $\mu_C(D)$, is called the quality of approximation of classification,induced by the set of decision features.

The goal of feature reduction is to remove redundant features so that the reduced set provides the same quality of classification as the original.A reduct is defined as a subset R of the conditional feature set C such that $\mu_R(D) = \mu_C(D)$. A given decision table may have many feature reducts, the set of all reducts is defined as

$$Red = \{R \subseteq C | \mu_R(D) = \mu_C(D) \forall B \subset R, \mu_B(D) \neq \mu_C(D)\} \tag{10}$$

In rough set feature reduction,a reduct with minimal cardinality is searched for.An attempt is made to locate a single element of the minimal reduct set $R_{min} \subseteq Red$

$$R_{min} = \{R \in Red | \forall R\prime \in Red,\ |R| \leqslant |R\prime|\}. \tag{11}$$

2.2. QUICKREDUCT feature selection

The problem of finding a feature reduct of a decision table has been the subject of much research. The QUICKREDUCT algorithm given in (Jensen and Shen, 2003), attempts to calculate a minimal reduct without exhaustively generating all possible subsets. It starts from an empty set and adds in turn, one at a time, those features that result in the greatest increase in dependency degree, until this produces its maximum possible value for the dataset.

Algorithm QUICKREDUCT

Input: a decision table $\mathscr{DT} = (U, C\bigcup D, V, f)$
Output: a feature reduct R
(1) Initial the feature reduct $R = \phi$ and a temporary variable $T = \phi$
(2) Do
(3) {
(4) $T = R$
(5) For every $x \in \{C - R\}$
(6) {
(7) If $\mu_{R\bigcup\{x\}}(D) > \mu_T(D)$, Then $T = R\bigcup\{x\}$
(8) }
(9) $R = T$
(10) } Until $\mu_R(D) == \mu_C(D)$
(11) Output R

According to the QUICKREDUCT algorithm, the dependency degree of each feature is calculated, and the best candidate chosen. However, it is not guaranteed to find a minimal feature set as its too greedy. Using the dependency degree to discriminate between candidates may lead the search down a non-minimal path.

Moreover, in some cases, QUICKREDUCT algorithm cannot find a feature reduct that satisfies the strict definition in Section 2.1, that is the feature subset discovered may contain irrelevant features. The classification accuracy may be degraded when designing a classifier using the feature subset with irrelevant features.

2.3. Mutual information based feature selection

In feature selection problems, the relevant features contain important information about output, whereas the irrelevant features contain little information regarding output. The task for feature selection is to find those input features that contain as much information about output as possible. For this purpose, Shannon's information theory (Shannon and Weaver, 1949) provides us a feasible way to measure the information of data set with entropy and mutual information (Miao and Hou, 2004; Miao and Wang, 1997).

Entropy can be used as an information measure of information table for feature selection. If initially only probabilistic knowledge about classes is given, then the uncertainty associated with the information table can be measured by entropy.

Definition 1. Let $\mathscr{I} = (U, A, V, f)$ be an information table. For any subset $B \subseteq A$ of features, let $U/IND(B) = \{X_1, X_2, \ldots, X_n\}$ denote the partition induced by equivalence relation $IND(B)$. The information entropy $H(B)$ of feature set B is defined as

$$H(B) = -\sum_{i=1}^{n} p(X_i)\log_2 p(X_i), \tag{12}$$

where $p(X_i) = |X_i|/|U|$, $1 \leqslant i \leqslant n$, $|X_i|$ is the cardinality of X_i.

Definition 2. Let $\mathscr{DT} = (U, C\bigcup D, V, f)$ be a decision table, $U/IND(C) = \{X_1, X_2, \ldots, X_n\}$ and $U/IND(D) = \{Y_1, Y_2, \ldots, Y_m\}$ denote the partitions induced by equivalence relations $IND(C)$ and $IND(D)$, respectively. The conditional entropy of D conditioned to C is defined as

$$H(D|C) = -\sum_{i=1}^{n} p(X_i)\sum_{j=1}^{m} p(Y_j|X_i)\log_2 p(Y_j|X_i), \tag{13}$$

where $p(X_i) = |X_i|/|U|$, $p(Y_j|X_i) = |X_i \bigcap Y_j|/|X_i|$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$.

Definition 3. Let $\mathscr{DT} = (U, C\bigcup D, V, f)$ be a decision table, $U/IND(C) = \{X_1, X_2, \ldots, X_n\}$ and $U/IND(D) = \{Y_1, Y_2, \ldots, Y_m\}$ denote the partitions induced by equivalence relations $IND(C)$ and $IND(D)$, respectively. The mutual information between C and D is defined as

$$I(C; D) = H(D) - H(D|C). \tag{14}$$

If the mutual information is large, the two feature sets are closely related. If the mutual information becomes zero, the two feature sets are independent.

One can consider mutual information related reducts to extract relevant feature sets with respect to the mutual information measure.

Definition 4. Let $\mathscr{DT} = (U, C\bigcup D, V, f)$ be a decision table. For any subset $B \subseteq C$ of features, if $I(B; D) = I(C; D)$ and for every $b \in B$, $I(B - \{b\}; D) < I(B; D)$, then B is called a feature reduct of C with respect to D in DT.

Definition 5. Let $\mathscr{DT} = (U, C\bigcup D, V, f)$ be a decision table. For every $a \in C$, if $I(C - \{a\}; D) < I(C; D)$, then a is a core feature of $\mathscr{DT}$.

The composition of all core features is called feature core. Feature core can be used as the starting point of reduction computation.

Definition 6. Let $\mathscr{DT} = (U, C\bigcup D, V, f)$ be a decision table. For any $B \subset C$ of features, and any feature $a \in C - B$, the significance of feature a with respect to B and D is defined as

$$sgn(a, B, D) = I(B\bigcup\{a\}; D) - I(B; D) \tag{15}$$

The significance of features can be used as heuristic information in greedy algorithms to compute a minimal feature reduct.

The elements of feature core are those features that cannot be eliminated.The algorithm for finding feature core is as follows:

Algorithm FEATURECORE
Input: a decision table $\mathscr{DT} = (U, C\bigcup D, V, f)$
Output: the feature *Core*
(1) Initial $Core = \phi$
(2) For every $a \in C$
(3) {
(4) If $I(C - \{a\}; D) < I(C; D)$, Then $Core = Core\bigcup\{a\}$
(5) }
(6) Output *Core*

In the worst case, the time complexity of Algorithm FEATURECORE is $O(mn^2)$, and its space complexity is $O(mn)$, where m and n are the cardinalities of C and U respectively. The feature selection algorithm starting from feature core based on mutual information (MIBR) is as follows:

Algorithm MIBR
Input: a decision table $\mathscr{DT} = (U, C\bigcup D, V, f)$
Output: a feature reduct R
(1) Initial the feature reduct $R = Core$ and a temporary variable $T = \phi$
(2) Do
(3) {
(4) $T = R$
(5) For every $a \in \{C - R\}$
(6) {
(7) If $I(R\bigcup\{a\}; D) > I(T; D)$, Then $T = R\bigcup\{a\}$
(8) }
(9) $R = T$
(10) } Until $I(R; D) == I(C; D)$
(11) Output R

According to the mutual information based algorithm, the features are sorted by the mutual information, and the best candidate chosen. However, it is also not guaranteed to find a minimal feature set as its too greedy.

3. Ant colony optimization

In the real world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep traveling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. Thus, when one ant finds a good (i.e. short) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads all the ants following a single path.

Inspired by the behavior of real ants, Dorigo and Caro (1999) proposed an artificial colony of ants algorithm, which was called the ant colony optimization (ACO) metaheuristic, to solve hard combinatorial optimization problems. The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph representing the problem to solve. The ACO was originally applied to solve the classical traveling salesman problem (Dorigo et al., 1996), where it was shown to be an effective tool in finding good solutions. They have an advantage over simulated annealing and genetic algorithm approaches when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. The ACO has also been successfully applied to other optimization problems including telecommunications networks, data mining, vehicle routing, etc.

ACO is a metaheuristic in which a colony 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. 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. 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. ACO algorithm is interplay of three procedures as described in (Dorigo and Caro, 1999): (1) Construct ant solutions; (2) update pheromones; and (3) Daemon actions.

It is worth mentioning that ACO makes probabilistic decision in terms of the artificial pheromone trails and the local heuristic information. This allows ACO to explore larger number of solutions than greedy heuristics. Another characteristic of the ACO algorithm is the pheromone trail evaporation, which is a process that leads to decreasing the pheromone trail intensity over time. pheromone evaporation helps in avoiding rapid convergence of the algorithm towards a sub-optimal region.

In the next section, we shall present our method to feature selection, which is based on rough sets and ACO, and explain how it is used for searching the feature space and selecting a minimal subset of features effectively.

4. Feature selection based on rough sets and ant colony optimization

Following the standard ACO algorithmic scheme for combinatorial optimization problems, Jensen and Shen propose a method for feature selection based on rough sets and ACO (JSACO) (Jensen and Shen, 2003). The basic procedure of JSACO is as follows: given a colony of k artificial ants to search through the feature space, these k ants perform a number of iterations. During every iteration t, each ant starts from a random feature, then selects the best route and the pheromone is updated. The algorithm stops iterating when a termination condition is met.

4.1. Problem representation

We can reformulate the problem of feature selection into an ACO-suitable problem. ACO requires a problem to be represented as a complete graph, where nodes represent features, and the edges between them denote the choices of the next features. The search for the optimal feature subset is then a traversal through the graph where a minimal number of nodes are visited and the traversal stopping criterions are satisfied. Jensen and Shen's method starts from a random feature, as shown in Fig. 1. Since in rough set theory, there exists a feature core of C. We propose a rough set approach to feature selection based on ACO starting from the feature core (RSFSACO). As shown in Fig. 2, we can transform that complete graph into a smaller one by deleting all features in the core from it.

In the left part of Fig. 2, if we assume that the ant is currently at node a, then it has a choice of which feature should be added next to its path (dotted lines). It chooses feature b next based on the transition rule, then c and then d. Upon arrival at d, the current subset $\{a, b, c, d\}$ is determined to satisfy the stopping criterions. And in the right part of Fig. 2, if the feature core is $\{a, b\}$, then we can change the complete graph shown in the left to a smaller one shown in the right. In this case, the ant start from the core $\{a, b\}$, and may choose feature c next, then d. Therefore the feature reduct can be found more quickly, if we introduce the core into the search.

4.2. Heuristic information

For most current methods used in ACO to solve combinatorial optimization problems, such as traveling salesman problem, the heuristic information is calculated before the construction of solutions. However, in RSFSACO the heuristic information is dynamically calculated during the construction process of solutions. The significance of features is adopted as heuristic information for RSFSACO. The significance of features is defined by information entropy and mutual information. Given a colony of ants and a decision table $\mathscr{DT} = (U, C \bigcup D, V, f)$, suppose $Core(C) = \{a, b\}$ denote the core of C with respect to D in DT. All ants start from $Core(C)$, and the given ant is currently at node p. Then the ant should find the next node. For any feature $r \in C - \{Core \bigcup p\}$, the heuristic information of feature r with respect to p is defined as follows:

$$\eta(r, p) = sgn(r, \{Core \bigcup p\}, D) \tag{16}$$

where sgn denotes the function of significance given in Definition 6.

In the above definition, if $\eta(r, p) < \varepsilon$, then let $\eta(r, p) = \varepsilon$, where $\varepsilon$is a small positive parameter.

4.3. Construction of feasible solutions

In RSFSACO, to construct a solution each ant should start from the feature core. Next the ant randomly selects a feature, then selects the second feature from those unselected features with a given probability. That probability is calculated by Dorigo et al. (1996):

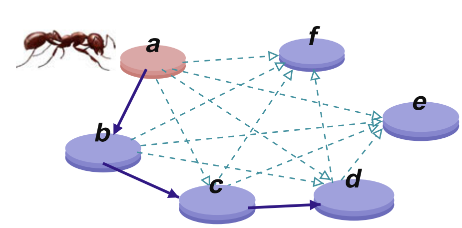

Fig. 1. An illustration of JSACO that an ant starting from a random feature.

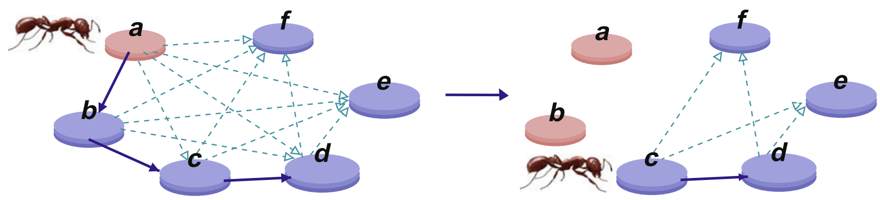

Fig. 2. An illustrative change of an ant going from the core.

$$p_{ij}^{k}(t)=\frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}(t)}{\sum_{l\in allowed_k}\tau_{il}^{\alpha}\eta_{il}^{\beta}(t)},\quad j\in allowed_k, \tag{17}$$

where k and t denote the number of ants and iterations, respectively, $allowed_k$ denotes the set of conditional features that have not yet been selected, $\tau(i,j)$ and $\eta(i,j)$ are the pheromone value and heuristic information of choosing feature j when at feature i. In addition, $\alpha > 0$ and $\beta > 0$ are two parameters which determine the relative importance of the pheromone trail and heuristic information.

If α is far larger than β, then ants will make decision mainly based on pheromone trails, and if β is far larger than α, ants will select those edges with higher heuristic information in a greedy manner. Based on preliminary experimental results, with larger values of β, it often takes ants longer time to reach a high quality solution, whereas ants find better solutions when their search behavior is mainly influenced by pheromone trails. α and β should be chosen in the range $0\sim 1$ and determined by experimentation.

A construction process is terminated by one of the following two conditions:

(1) $I(R;D) = I(C;D)$, where R is the current solution constructed by an ant.
(2) The cardinality of the current solution is larger than that of the temporary minimal feature reduct.

The first condition means that a better solution has been constructed and a reduct has been found. The first reduct is regarded as the minimal reduct temporarily. Next reducts compare with the minimal reduct. If the cardinality of the current solution is smaller than that of the minimal reduct, the current solution is regarded as the new minimal reduct. Otherwise, the current solution is discard. The second condition implies that no better solution will be constructed, thus it is unnecessary to continue the construction process.

4.4. Pheromone updating

After each ant has constructed a solution, the pheromone on each edge should be updated according to the following rule:

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau_{ij}(t), \tag{18}$$

where $\tau_{ij}(t)$ is the amount of pheromone on a given edge (i,j) at iteration t, $\tau_{ij}(t+1)$ is the amount of pheromone on a given edge (i,j) at next iteration, $\rho(0<\rho<1)$ is a decay constant used to simulate the evaporation of pheromone, and $\Delta\tau_{ij}(t)$ is the amount of pheromone deposited, typically given by

$$\Delta\tau_{ij}(t)=\begin{cases}\sum\frac{q}{|R(t)|} & \textit{if edge}(i,j)\textit{ has been traversed}\\ 0 & \textit{otherwise}\end{cases} \tag{19}$$

where parameter q is a given constant, $R(t)$ is the minimal feature reduct at iteration t. By this definition, all ants update the pheromone.

If an optimal subset has been found or the iteration reaches the maximum cycle, then the algorithm halts and outputs the minimal feature reduct encountered. If neither condition holds, then the pheromone is updated, a new set of ants are created and the process iterates once more.

4.5. The pseudo-code of RSFSACO

Algorithm RSFSACO.

Input: a decision table $\mathscr{DT}=(U,C\bigcup D,V,f)$ and parameters
Output: a minimal feature reduct *Rmin* and its cardinality *Lmin*
(1) Initial $R_{min}=C$, $L_{min}=|C|$, $iteration=0$
(2) Compute $I(C;D)$ by formula (12)–(14)
(3) Compute *Core* by algorithm FEATURECORE:
  For every $c\in C$, If $I(C-\{c\};D)<I(C;D)$, then $Core=\{\bigcup c\}$
(4) While ($iteration <= maxcycle$)
(5) {
(6)   For every $k\in Ant$
(7)   {
(8) $R_k = Core$, $L_k=|Core|$
(9)   Select a feature $a_k\in\{C-Core\}$ randomly
      $R_k=\{R_k\bigcup a_k\}$, $L_k=L_k+1$
(10)   Do: construct a solution
(11)   {
(12)   For every $s_k\in\{C-R_k\}$
(13)   {
(14)   Compute heuristic information $\eta(s_k,a_k)$ by formula (15) and (16)
(15)   }
(16)   Select next feature $b_k\in\{C-R_k\}$ by formula (17)
       $R_k=\{R_k\bigcup b_k\}$
       $L_k=L_k+1$
(17)   } Until ($I(R_k;D)==I(C;D)$ or $L_k>=L_{min}$)
(18)   If ($I(R_k;D)==I(C;D)$ and $L_k<L_{min}$)
     Then $R_{min}=R_k$, $L_{min}=L_k$
(19)   }
   :update pheromone
(20)   For every $x,y\in\{R_{min}-Core\}$
(21)   {
(22) $\tau_{x,y}=\rho\tau_{x,y}+q/L_{min}$
(23)   }
(24)   For every $u,v\in\{C-R_{min}\}$
(25)   {
(26) $\tau_{u,v}=\rho\tau_{u,v}$
(27)   }
(28) $iteration=iteration+1$
(29) }
(30) Output R_{min} and L_{min}

5. Experimental results

In this section, we shall demonstrate the performance of our algorithm RSFSACO given in Section 4. The algorithm is tested on a personal computer running windows XP with 2.0 GHZ processor and 1 GB memory. In our experiments, we set the parameters $\alpha = 1$, $\beta = 0.01$, $\rho = 0.9$, $q = 0.1$, $\varepsilon = 0.001$, and the initial pheromone was set to 0.5 with a small random perturbation added, the number of ants was half the number of features and the maximum number of cycles equals 100. These parameters are determined based on a small number of preliminary runs. The halting condition is reaching the maximum cycle or getting the same feature reduct under three consecutive iterations. As suggested by Jensen and Shen (2004), each dataset is test for 20 times.

5.1. Comparison with RST-based methods

Our experiments are carried out on nine datasets which are from UCI datasets. In order to find whether our algorithm could find an optimal reduct, we compute all reducts using boolean reasoning methods as described in (Pal and Skowron, 1999) for reference. We compare the performance of our algorithm RSFSACO with traditional RST-based methods to feature select, which are RSQR (using QUICKREDUCT) and MIBR (Mutual Information Based Reduction). Note that when we talk about optimal reduct we refer to the minimal reduct.

The experimental results are summarized in Table 1. The leftmost column consists of dataset names. The 2nd and 3rd columns are instance numbers and feature numbers of corresponding dataset. The 4th column is the length of optimal reduct according to all reducts. The 5th and 6th columns are results of two RST-based algorithms. The rightmost column is result of our algorithm. These results are the lengths of feature reduct each algorithm finds. Since the RSFSACO may be get different result, each dataset is test for 20 times and the number in parentheses denotes the times of tests to achieve such a feature reduct. A minimal reduct could not be deter-

Table 1
Experimental results comparison with RST-based methods.

Dataset	Inst.	Feat.	MinRedu	RSQR	MIBR	RSFSACO
Audiology	200	70	–	20	19	$13^{(4)}14^{(16)}$
Breast-cancer	699	10	4	4	4	4
Chess-king	3196	37	29	30	30	29
Monk1	124	7	3	3	3	3
Monk3	122	7	4	4	4	4
Mushroom	8124	23	4	6	5	$4^{(13)}5^{(7)}$
Vote	435	17	9	12	11	9
Wine	178	14	5	6	5	$5^{(17)}6^{(3)}$
Zoo	101	17	5	6	5	5

Table 4
RSFSACO searching process on Audiology.

Iteration	Best solution	Feature subset length
1	1,2,4,5,6,10,11,15,35,40,47,57,58,59,60,64,66	17
2	1,2,4,5,6,10,11,15,35,40,47,57,59,60,64,66	16
3	1,2,5,6,10,15,18,35,40,47,57,59,60,64,66	15
4	1,2,4,5,6,10,11,15,35,40,47,57,60,64,66	15
5	1,2,4,5,6,10,14,15,35,40,47,60,64,66	14
6	1,2,4,5,6,10,11,14,15,47,60,64,66	13
7	1,2,4,5,6,10,11,14,15,47,60,64,66	13
8	1,2,4,5,6,10,11,14,15,47,60,64,66	13

Table 5
RSFSACO searching process on Mushroom.

Iteration	Best solution	Feature subset length
1	5,9,11,15,19,22	6
2	5,8,9,11,19,22	6
3	5,9,11,19,22	5
4	5,9,11,22	4
5	5,9,11,22	4
6	5,9,11,22	4

Table 6
RSFSACO searching process on vote.

Iteration	Best solution	Feature subset length
1	1,2,3,4,8,9,11,13,15,16	10
2	1,2,3,4,7,9,11,13,15,16	10
3	1,2,3,4,9,11,13,15,16	9
4	1,2,3,4,9,11,13,15,16	9
5	1,2,3,4,9,11,13,15,16	9

Table 2
Experimental results comparison with JSACO.

Dataset	Inst.	Feat.	Core	MinRedu	JSACO		RSFSACO	
					Redu	Average time (s)	Redu	Average time (s)
Audiology	200	70	3	–	$20^{(2)}21^{(18)}$	352.261	$13^{(4)}14^{(16)}$	298.332
Breast-cancer	699	10	1	4	4	0.468	4	0.439
Chess-king	3196	37	27	29	$29^{(4)}30^{(11)}31^{(5)}$	218.016	29	152.762
Monk1	124	7	3	3	3	0.012	3	< 0.01
Monk3	122	7	4	4	4	0.013	4	< 0.01
Mushroom	8124	23	0	4	$4^{(13)}5^{(7)}$	23.391	$4^{(13)}5^{(7)}$	51.399
Vote	435	17	7	9	$10^{(5)}11^{(7)}12^{(8)}$	1.798	9	0.731
Wine	178	14	0	5	$5^{(17)}6^{(3)}$	0.088	$5^{(17)}6^{(3)}$	0.091
Zoo	101	17	2	5	$5^{(18)}6^{(2)}$	0.089	5	0.061

Table 3
Experimental results comparison with other metaheuristics.

Dataset	Inst.	Feat.	Core	MinRedu	GenRSAR	TSAR	RSFSACO
Audiology	200	70	3	–	$20^{(3)}21^{(9)}22^{(8)}$	$20^{(8)}21^{(12)}$	$13^{(4)}14^{(16)}$
Mushroom	8124	23	0	4	$5^{(1)}6^{(5)}7^{(14)}$	$4^{(12)}5^{(8)}$	$4^{(13)}5^{(7)}$
Vote	435	17	7	9	$10^{(1)}11^{(8)}12^{(11)}$	$10^{(3)}11^{(9)}12^{(8)}$	9

Table 7
Classification results with different reducts.

Dataset	Inst.	Fea.	GenRSAR Number of rules	GenRSAR Classification accuracy	TSAR Number of rules	TSAR Classification accuracy	RSFSACO Number of rules	RSFSACO Classification accuracy
Audiology	200	70	126	96.45%	108	95.3%	89	98.91%
Mushroom	8124	23	67	100%	52	100%	19	100%
Vote	435	17	27	94.23%	25	93.12%	25	95.88%

mined with the all reduction algorithm running out of time, such as Audiology. A symbol '-' indicates the unknown minimal reduct.

From the results, we can see that RST-based methods produce the same reduct every time, while RSFSACO often finds different reducts and sometimes different lengths of reducts. It is obvious that in some situations hill-climbing methods can locate the optimal feature reduct. For example, RSQR finds the optimal reduct for dataset Breast-cancer, Monk1 and Monk3. MIBR finds the optimal reduct for dataset Breast-cancer, Monk1, Monk3, Wine and Zoo. But for other datasets, sub-optimal reducts are found, containing redundant features. The reducts found by RSQR often contain more redundant features than MIBR and RSFSACO, while the results found by MIBR often contain more redundant features than RSFSACO. According to the experimental results, we can see that RSFSACO outperforms the other traditional RST-based methods as respect to ability of finding optimal reduct.

5.2. Comparison with Jensen and Shen proposed ACO-based approach

Jensen and Shen proposed an ACO-based feature selection approach which starts from a random feature (Jensen and Shen, 2003), while our method starts from the feature core. Jensen and Shen's algorithm is called as JSACO. We have tested several datasets from UCI datasets. Table 2 shows the results of feature selection on those datasets. The leftmost column consists of dataset names. The 2nd and 3rd columns are the number of instances and the number of features in a dataset. The 4th and 5th columns are the length of feature core and minimal reduct. The 6th and 7th columns are the length of feature reduct found by Jesen and Shen's approach and its average run time. The 8th and 9th columns are the length of feature reduct found by our method and its average run time. Each dataset is tested for 20 times and the number in parentheses denotes the times of tests to achieve such a feature reduct.

In Monk1 and Monk3, the feature reduction is finished after the core is found, because the core is a reduct. In Mushroom and Wine, the core is empty. In other datasets, the core is not empty.

From the data present in Table 2, the following conclusion can be drawn: (a) For non-core datasets, both algorithms have the same ability to find optimal reducts, such as Mushroom and Wine. However, for core datasets, our algorithm outperforms JSACO with respect to the ability to find optimal reducts, such as Audiology, Breast-cancer, Chess-king and Zoo. (b) When the dataset has not a core, the JSACO outperforms our algorithm in run time. This is because our algorithm wastes some time in calculating core. When the dataset has a core, our algorithm outperforms JSACO in run time. This is because our algorithm changes the complete graph to a smaller one. The run time of ACO-based algorithm depends heavily on the construction of feasible solution as ants traverse the complete graph.

5.3. Comparison with other metaheuristics

We also compare RSFSACO with other metaheuristic algorithms, GenRSAR (GA-based) (Zhai et al., 2002) and TSAR (TS-based) (Hedar et al., 2006). These algorithms are carried out on three UCI datasets, Audidogy, Mushroom and Vote. The experimental results are summarized in Table 3.

From Table 3, we can see that RSFSACO provides the best results, the performance of TSAR is better than GenRSAR. Results of the searching progress of RSFSACO are listed in Table 4 for the Audiology, in Table 5 for the Mushroom and in Table 6 for the Vote. In Table 4, "Best solution" lists the best feature subset encountered at an iteration, in which each number denotes one feature of the dataset.

From the results, we can see that RSFSACO has the ability to quickly converge in locating the optimal solution. In general, it can find the optimal solution within tens of iterations. If exhaustive search is used to find the optimal reduct in the dataset Audiology, there will be tens of thousands of candidate subsets, which is impossible to execute. But with RSFSACO, at the sixth iteration the satisfactory solution is found.

We also use the LEM2 algorithm (Stefanowski, 1998) to extract rules from the data and the global strength for rule negotiation in classification. We apply ten-fold cross validation to estimate the classification. The number of decision rules and the classification accuracy with different feature reducts are showed in Table 7. Most of the reducts found by RSFSACO result in smaller rules and exhibit higher classification accuracy.

6. Conclusion

This paper discusses the shortcomings of conventional hill-climbing rough set approaches to feature selection. These techniques often fail to find optimal reducts, as no perfect heuristic can guarantee optimality. On the other hand, complete searches are not feasible for even medium sized datasets. So, ACO approaches provide a promising feature selection mechanism.

We proposed a novel feature selection technique based on rough sets and ACO. ACO has the ability to quickly converge. It has a strong search capability in the problem space and can efficiently find minimal reducts. An algorithm called RSFSACO was also given, which starts from the core and utilizes mutual information based feature significance as heuristic information to search through the feature space for optimal solutions. Our algorithm has the following characteristics: (a) It applies rough set reduction method and ACO technique to feature selection; (b) It constructs a solution starting from the feature core; and (c) its heuristic information is on the basis of mutual information based feature significance. Experimental results on real datasets demonstrate the effectiveness of our method to feature selection.

In the future work, more experimentation and further investigation into this technique may be required. We should extend to approximate entropy reduction. For large datasets, to speed up the computations of reduction, parallel algorithm may be employed.

Acknowledgements

The research is supported by the National Natural Science Foundation of China under Grant Nos: 60775036, 60475019, and the Research Fund for the Doctoral Program of Higher Education of China under Grant No: 20060247039.

References

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems. Oxford Univ. Press, New York.

Dash, M., Liu, H., 1997. Feature selection for classification. Intell. Data Anal. 1 (3), 131–156.

Dash, M., Liu, H., 2003. Consistency-based search in feature selection. Artif. Intell. 151, 155–176.

Deogun, J., Choubey, S., Raghavan, V., Sever, H., 1998. Feature selection and effective classifiers. J. ASIS 5, 403–414.

Dorigo, M., Caro, G.D., 1999. Ant colony optimization: A new meta-heuristic. In: Proc. Congress on Evolutionary Computing.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. The Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernet. Part B 26 (1), 29–41.

Duan, Q.G., Miao, D.Q., et al., 2007. Personalized Web retrieval based on rough-fuzzy method. J. Comput. Inform. Systems 3 (3), 1067–1074.

Gasca, E., Sanchez, J.S., Alonso, R., 2006. Eliminating redundancy and irrelevance using a new MLP-based feature selection method. Pattern Recognition 39 (2), 313–315.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine Learn. 46, 389–422.

Hedar, A., Wang, J., Fukushima, M., 2006. Tabu search for attribute reduction in rough set theory, Technical Report 2006-008, Department of Applied Mathematics and Physics, Kyoto University.

Hu, X., 1995. Knowledge discovery in databases: An attribute oriented rough set approach. Ph.D. Thesis, Regina University.

Jensen, R., 2005. Combining rough and fuzzy sets for feature selection, Ph.D. Thesis, Univ. Of Edinburgh.

Jensen, R., Shen, Q., 2003. Finding rough set reducts with ant colony optimization. In: Proceeding of 2003 UK Workshop Computational Intelligence, pp. 15–22.

Jensen, R., Shen, Q., 2004. Semantics-preserve dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE Trans. Knowledge Data Eng. 16, 1457–1471.

Kira, K., Rendell, L.A., 1992. The feature selection problem: Traditional methods and a new algorithm. In: Proc. AAAI-92, San Jose, CA, pp. 129–134.

Kohavi, R., 1994. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In: Proc. AAAI Fall Symposium on Relevance, pp. 109–113.

Lai, C., Reinders, M.J.T., Wessels, L., 2006. Random subspace method for multivariate feature selection. Pattern Recognition Lett. 27, 1067–1076.

Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer, Boston.

Maniezzo, V., Colorni, A., 1999. The ant system applied to the quadratic assignment problem. Knowledge Data Eng. 11 (5), 769–778.

Miao, D.Q., Hou, L., 2004. A comparison of rough set methods and representative inductive learning algorithms. Fundamenta Inform. 59 (2–3), 203–219.

Miao, D.Q., Wang, J., 1997. Information-based algorithm for reduction of knowledge. IEEE Internat. Conf. Intell. Process. Systems, Beijing, China, 1155–1158.

Mi, J.S., Wu, W.Z., Zhang, W.X., 2004. Approaches to knowledge reduction based on variable precision rough set model. Inform. Sci. 159 (3–4), 255–272.

Modrzejewski, M., 1993. Feature selection using rough sets theory. In: Proceedings of the European Conference on Machine Learning, Vienna, Austria, pp. 213–226.

Neumann, J., Schnorr, C., Steidl, G., 2005. Combined SVM-based feature selection and classification. Machine Learn. 61, 129–150.

Pal, S.K., Skowron, A., 1999. Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer.

Pawlak, Z., 1982. Rough sets. Internat. J. Comput. Inform. Sci. 11 (5), 341–356.

Raymer, M.L., Punch, W.E., Goodman, E.D., et al., 2000. Dimensionality reduction using genetic algorithms. IEEE Trans. Evol. Comput. 4 (2), 164–171.

Shannon, C.E., Weaver, W., 1949. The Mathematical Theory of Communication. University of Illinois Press, Urbana, Israel.

Somol, P., Pudil, P., Kittler, J., 2004. Fast branch & bound algorithms for optimal feature selection. IEEE Trans. Pattern Anal. Machine Intell. 26 (7), 900–912.

Stefanowski, J., 1998. On rough set based approaches to induction of decision rules. In: Skowron, A., Polkowski, L. (Eds.), Rough Sets in Knowledge Discovery, vol. 1. Physica Verlag, Heidelberg, pp. 500–529.

Swiniarski, R.W., Skowron, A., 2003. Rough set methods in feature selection and recognition. Pattern Recognition Lett. 24, 833–849.

Wrblewski, J., 1995. Finding minimal reducts using genetic algorithms. In: Proc. Second Annual Join Conf. on Information Sciences, Wrightsville Beach, NC, 28(1), pp. 186–189.

Xie, Z.X., Hu, Q.H., Yu, D.R., 2006. Improved feature selection algorithm based on SVM and correlation, ISNN1, pp. 1373–1380.

Yao, Y.Y., 2003. Information-theoretic measures for knowledge discovery and data mining. In: Karmeshu (Ed.), Entropy Measures, Maximum Entropy and Emerging Applications. Springer, Berlin, pp. 115–136.

Zhai, L.Y. et al., 2002. Feature extraction using rough set theory and genetic algorithms: An application for the simplification of product quality evaluation. Comput. Indust. Eng. 43, 661–676.

Zhong, N., Dong, J.Z., 2001. Using rough sets with heuristics for feature selection. J. Intell. Inform. Systems 16, 199–214.