International Journal of Approximate Reasoning 55 (2014) 167–179

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

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

Feature selection with test cost constraint

Fan Min [a,*], Qinghua Hu [b], William Zhu [a]

[a] Lab of Granular Computing, Zhangzhou Normal University, Zhangzhou 363000, China
[b] Tianjin University, Tianjin 300072, China

ARTICLE INFO

Article history:
Available online 11 April 2013

Keywords:
Feature selection
Cost-sensitive learning
Constraint satisfaction problem
Backtracking algorithm
Heuristic algorithm
Decision-theoretic rough sets

ABSTRACT

Feature selection is an important preprocessing step in machine learning and data mining. In real-world applications, costs, including money, time and other resources, are required to acquire the features. In some cases, there is a test cost constraint due to limited resources. We shall deliberately select an informative and cheap feature subset for classification. This paper proposes the feature selection with test cost constraint problem for this issue. The new problem has a simple form while described as a constraint satisfaction problem (CSP). Backtracking is a general algorithm for CSP, and it is efficient in solving the new problem on medium-sized data. As the backtracking algorithm is not scalable to large datasets, a heuristic algorithm is also developed. Experimental results show that the heuristic algorithm can find the optimal solution in most cases. We also redefine some existing feature selection problems in rough sets, especially in decision-theoretic rough sets, from the viewpoint of CSP. These new definitions provide insight to some new research directions.

1. Introduction

Many data mining approaches employ feature selection techniques to speed up learning and to improve model quality [19,27,81]. These techniques are especially important for datasets with tens or hundreds of thousands of features [14,33]. Attribute reduction [51] is a special type of feature selection problems studied by the rough set society. A reduct is a feature subset that is jointly sufficient and individually necessary to preserve certain information of the data [72]. For decision making, the most often addressed information is the positive region with respect to the decision class [51]. The objective of the classical reduct problem is to find a minimal reduct [56], since simpler representation often provides better generalization ability according to Occam's razor principle. Other feature selection problems aim at finding feature subsets with maximal margin [6], maximal stability [3], minimal space [40], maximum relevance-maximum significance [37], best overall quality of the potential set of rules [68], etc.

Most of these problems assume the data are already stored in datasets and available without charge. However, data are not free in real-world applications. There are test costs, such as money, time, or other resources [41,62] to obtain feature values of objects. For example, it takes both time and money to obtain medical data of a patient [78]. Under this context, one would like to select the cheapest reduct [60]. This consideration and the parallel test assumption have motivated the minimal test cost reduct (MTR) problem [41]. Recently, a number of algorithms have been developed to deal with this problem (see, e.g., [16,41,50]). Other related issues have also been identified in addressing numerical features [79], observational errors [47], and test costs relationships [17,42]. All these problems aim at searching the cheapest feature subset which preserves sufficient information for classification.

Nevertheless, the available resource is usually limited, and users have to sacrifice necessary information to keep the test cost under budget. We have introduced the feature selection with test cost constraint (FSTC) problem [46] to formulate this

[*] Corresponding author. Tel.: +86 133 7690 8359.
E-mail addresses: minfanphd@163.com (F. Min), huqinghua@hit.edu.cn (Q. Hu), williamfengzhu@gmail.com (W. Zhu).

0888-613X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2013.04.003

issue. The upper bound of the available resource serves as the constraint. The FSTC problem is more general than MTR [41]. In fact, these two problems coincide when the constraint is no less than the test cost of the optimal reduct. If the constraint is so tight that the sufficiency condition cannot be met, then one cannot obtain a reduct. This is why the new problem falls in *feature selection* instead of in *attribute reduction*.

In this paper, the FSTC problem is redefined from the viewpoint of the constraint satisfaction problem (CSP). Specifically, it is defined with four aspects, namely input, output, constraint, and optimization objective. The new definition is simpler and easier to comprehend than the one defined from the viewpoint of set family [46]. Furthermore, we redefine the classical reduct problem and the minimal reduct problem [56] from the CSP viewpoint. We show that most feature selection problems in rough sets, including those of decision-theoretic rough sets (DTRS) [34,54,67,71–73,76] and game-theoretic rough set model (GTRS) [1,2,18], can be viewed as extensions of the minimal reduct problem [56] from one or more of these four aspects. This viewpoint gives insight to meaningful research trends concerning feature selection in a broader sense.

There are some closely related works to this one concerning the viewpoint of the reduct problem. Yu et al. [77] defined the problem explicitly as a CSP with the form of variables, functions and constraints. Jensen et al. [24] reformulated the problem in a propositional satisfiability (SAT) framework, and analyzed its relationship with CSP. Jia et al. [25,26] presented an optimization viewpoint of the problem on DTRS. Compared with them, this work is more systematic.

We develop a backtracking algorithm to the FSTC problem for small and medium-sized datasets. Backtracking algorithms are natural and effective approaches to CSPs for obtaining one or all optimal solutions. However, they are seldom employed to deal with feature selection problems in rough set theory (see, e.g., [5,47]), where discernibility matrix based approaches are more popular (see, e.g., [53,56,65,75]). One possible reason is that only a few people (see, e.g., [24,77]) have addressed attribute reduction problems explicitly from the viewpoint of CSP. As an exhaustive algorithm, the backtracking algorithm has a time complexity exponential with respect to the number of features.

We also develop a heuristic algorithm with polynomial time complexity for large datasets. We employ the addition–deletion approach [73] to design a heuristic function based on information gain often employed in similar problems [8,57,63,73]. It is similar to the one proposed in [41] to prefer low cost features through λ-weighting, where λ is a user specified parameter. The difference between the new algorithm and the one employed in [41] lies in the stopping criteria. To improve the performance of the algorithm, we employ the competition strategy [41]. With this strategy, different feature subsets are obtained through setting different λ values, then the best one is selected. This strategy can trade the quality of the result with the runtime. More importantly, with this strategy, the user is not involved in the setting of λ. Instead, a set of λ values which are valid for any dataset are specified by the algorithm.

Five open datasets are employed to study the performance of our algorithms. Experimental results show that the backtracking algorithm is efficient for medium-sized data. It takes less than 0.4 s to obtain an optimal feature subset for the mushroom dataset, which contains 22 features and 8124 objects. The backtracking algorithm is approximately 10 times faster than SESRA [46], which is based on another definition of the problem. The heuristic algorithm is stably more efficient than the backtracking one. With the help of the competition strategy, the heuristic algorithm can find the optimal solution in most cases.

The rest of the paper is organized as follows: Section 2 presents the problem definition. The classical reduct problem and the minimal test cost reduct problem are also redefined. Section 3 proposes both backtracking and heuristic algorithms. Experimental results on five UCI (University of California – Irvine) datasets are discussed in Section 4. Then Section 5 studies existing feature selection problems in the rough set society from the viewpoint of CSP. Some interesting new problems are also briefly discussed. Finally, Section 6 presents the concluding remarks and further research directions.

2. Problem definition

This section reviews three feature selection problems in rough sets. Two of them are under the classical rough sets [51], and the last one is concerned with test cost [41]. These problems are redefined as CSPs. Moreover, we propose a new problem called feature selection with test cost constraint.

2.1. Classical feature selection problems in rough sets

Data models are fundamental for feature selection. This paper only considers decision systems.

Definition 1 [69]**.** A *decision system* (DS) S is the 5-tuple:

$$S = (U, C, d, V = \{V_a | a \in C \cup \{d\}\}, I = \{I_a | a \in C \cup \{d\}\}), \tag{1}$$

where U is a finite set of objects called the universe, C is the set of features, d is the decision class, V_a is the set of values for each $a \in C \cup \{d\}$, and $I_a : U \to V_a$ is an information function for each $a \in C \cup \{d\}$.

Let the decision system $S = (U, C, d, V, I)$ be nominal, that is, all features in C are nominal. Any $\emptyset \neq B \subseteq C \cup \{d\}$ determines an indiscernibility relation $I(B)$ on U. A partition determined by B is denoted by $U/I(B)$, or U/B for brevity. Let $\underline{B}(X)$ denote the *B-lower approximation* of X. The positive region of $\{d\}$ with respect to $B \subseteq C$ is defined as $POS_B(\{d\}) = \bigcup_{X \in U/\{d\}} \underline{B}(X)$ [51,52].

Definition 2 [52]. Any $B \subseteq C$ is called a *decision relative reduct* (or *a reduct* for short) of S iff:

1. $POS_B(\{d\}) = POS_C(\{d\})$; and
2. $\forall a \in B$, $POS_{B-\{a\}}(\{d\}) \subset POS_C(\{d\})$.

Definition 2 indicates that a reduct is (1) jointly sufficient and (2) individually necessary for preserving a particular property (positive region in this context) of the decision system [31,51,72,80]. In other words, there are two constraints, named sufficiency and necessity, respectively. Consequently, the problem of obtaining one reduct can be defined in the CSP style as follows.

Problem 3. The attribute reduction problem.
Input: $S = (U, C, d, V, I)$;
Output: $B \subseteq C$;
Constraints: (1) $POS_B(\{d\}) = POS_C(\{d\})$;
(2) $\forall a \in B$, $POS_{B-\{a\}}(\{d\}) \subset POS_C(\{d\})$.

There may exist many reducts for a decision system. Let the set of all relative reducts of S be $Red(S)$. Any $R \in Red(S)$ is a minimal reduct if and only if $|R|$ is minimal. Minimal reducts are preferred because they provide the simplest representation of the knowledge. The problem of finding a minimal reduct is called the minimal reduct problem, as defined as follows.

Problem 4. The minimal reduct problem.
Input: $S = (U, C, d, V, I)$;
Output: $B \subseteq C$;
Constraint: $POS_B(\{d\}) = POS_C(\{d\})$;
Optimization objective: $\min |B|$.

Problem 4 has an optimization objective, which is typical in CSP. Note that that there is only one constraint, namely sufficiency. This does not indicate that the necessity constraint is not met. In fact, the necessity constraint can be derived from the optimization objective. One can easily prove this by contradiction. That is, if there are superfluous features, the size of the feature subset cannot be minimal. In other words, the problem definition is simplified while viewed as a CSP.

2.2. Feature selection minimizing test cost

Test cost is an important issue in many applications. We have built a hierarchy of six test-cost-sensitive decision systems [42]. Here we present a simple model which will be used in defining the new problem of this paper.

Definition 5 [42]. A *test-cost-independent decision system* (TCI-DS) S is the 6-tuple:

$$S = (U, C, d, \{V_a | a \in C \cup \{d\}\}, \{I_a | a \in C \cup \{d\}\}, c), \tag{2}$$

where $U, C, d, \{V_a\}$, and $\{I_a\}$ have the same meanings as in Definition 1, $c : C \to \mathbb{R}^+ \cup \{0\}$ is the test cost function. Test costs are independent of one another, that is, $c(B) = \sum_{a \in B} c(a)$ for any $B \subseteq C$.

The minimal test cost reduct (MTR) problem proposed in [41] can be redefined as follows.

Problem 6. The minimal reduct problem.
Input: $S = (U, C, d, V, I, c)$;
Output: $B \subseteq C$;
Constraint: $POS_B(\{d\}) = POS_C(\{d\})$;
Optimization objective: $\min c(B)$.

One can see there are two differences between Problem 6 and Problem 4. The first is the input, where the test cost is the external information. The second is the optimization objective, which is to minimize the test cost, instead of the number of features.

2.3. Feature selection with test cost constraint

Sometimes we are given limited resources to obtain the feature values. We proposed the issue of optimal sub-reduct in [46] to address this issue. Here we use the positive region instead of the conditional information entropy to define respective concepts.

Definition 7. Let $S = (U, C, d, V, I, c)$ be a TCI-DS and m the test cost upper bound. The set of all feature subsets subject to the constraint is

$$T(S, m) = \{B \subseteq C | c(B) \leq m\}. \tag{3}$$

In $T(S, m)$, the set of all feature subsets with the maximal positive region is

$$M_T(S, m) = \{B \in T(S, m) || POS_B(\{d\})| = \max\{|POS_{B'}(\{d\})||B' \in T(S, m)\}\}. \tag{4}$$

In $M_T(S, m)$, the set of all optimal sub-reducts is

$$P_{M_T}(S, m) = \{B \in M_T(S, m) | c(B) = \min\{c(B') | B' \in M_T(S, m)\}\}. \tag{5}$$

Any element in $P_{M_T}(S, m)$ is called an optimal sub-reduct with test cost constraint, or an optimal sub-reduct for brevity.

In Definition 7, Eq. (3) ensures the constraint is met; Eq. (4) ensures most informative feature subset is selected; and Eq. (5) ensures test cost is minimized. The problem of constructing $P_{M_T}(S, m)$ is called the *optimal sub-reducts with test cost constraint* (OSRT) problem [46]. Unfortunately, the definition is rather prolonged and hard to read. Next we follow the style of Problem 4 to present the following problem.

Problem 8. The feature selection with test cost constraint (FSTC) problem.
Input: $S = (U, C, d, V, I, c)$, the test cost upper bound m;
Output: $B \subseteq C$;
Constraint: $c(B) \leq m$;
Optimization objectives: (1) $\max |POS_B(\{d\})|$; and (2) $\min c(B)$.

Note that the two objectives are not equally important. They are the primary and the secondary objectives, respectively. In fact, Problem 8 is the same as the OSRT problem. However the problem definition is simpler and easier to comprehend. This phenomenon indicates that the form of CSP is more appropriate for this kind of problems.

By comparing Problems 6 and 8, we observe the following differences. First, the constraint is expressed by the test cost instead of the positive region. Second, the first objective of Problem 8 is to maximize the positive region. Third, the objective of Problem 6 becomes the secondary objective of Problem 8. This objective is considered after the primary one is achieved.

In fact, Problem 8 is more general than Problem 6. Let B' be a minimal test cost reduct subject to Problem 6. If $m \geq c(B')$, the constraint is met when the primary objective is achieved. In other words, the constraint is essentially redundant. The first objective will be replaced by $POS_B(\{d\}) = POS_C(\{d\})$, which serves as a constraint. The second objective is then the only objective. Consequently, Problem 8 coincides with Problem 6 in this case.

3. Algorithm design

This section presents two algorithms. One is a backtracking algorithm, and the other is a heuristic algorithm. The backtracking algorithm always produces an optimal solution to the problem. The heuristic algorithm is more efficient to large datasets, however the feature subset obtained may not be optimal.

3.1. The backtracking algorithm

The backtracking algorithm is a natural solution to CSP. In the rough set society, people seldom employ this algorithm for attribute reduction. This is partly due to the form of problem definition as shown in Definition 2. The backtracking algorithm to the FSTC problem is illustrated in Algorithm 1. To invoke the algorithm, one should initialize the global variables m, let $B = \emptyset$, and use the following statement:
backtracking($\emptyset$, 0);
then at the end of the algorithm execution, an optimal feature subset will be stored in B.

In Algorithm 1, Lines 3 through 5 check the constraint. Feature subsets violating the constraint are simply discarded. Lines 6 through 8 indicate if the positive region of the current feature subset is the same as C, namely the sufficiency condition can be met, the FSTC problem coincides with the MTR problem. In this case we only need to address the MTR problem. Lines 9 through 11 are devoted to the optimization objective. $|POS_{B''}(\{d\})| > |POS_B(\{d\})|$ serves for the first objective. $c(B'') < c(B)$ serves for the second; it is checked only if $POS_{B''}(\{d\}) = POS_B(\{d\})$. In our implementation in Coser [48], the algorithm is implemented to avoid repeated computation of positive regions.

Algorithm 1. The backtracking algorithm to the FSTC problem

```
Input: Selected feature subset $B'$, feature index lower bound $l$
Output: Results are stored in the global variable $B$
Method: backtracking
 1: for ($i = l$; $i < |C|$; $i$++) do
 2:   $B'' = B' \cup \{a_i\}$;//One more feature
 3:   if ($c(B'') > m$) then
 4:     continue;//The constraint is violated
 5:   end if
 6:   if ($POS_{B''}(\{d\}) = POS_C(\{d\})$) then
 7:     throw new Exception("Coincides with the MTR problem");
 8:   end if
 9:   if ($(|POS_{B''}(\{d\})| > |POS_B(\{d\})| \vee (POS_{B''}(\{d\}) = POS_B(\{d\})) \wedge (c(B'') < c(B))$) then
10:     $B = B''$;//A better feature subset
11:   end if
12:   backtracking($B''$, $i + 1$);//Backtracking
13: end for
```

Table 1
A decision table for Example 9.

U	a_1	a_2	a_3	d
x_1	Y	Y	Y	A
x_2	N	Y	N	B
x_3	Y	N	N	B
x_4	N	N	Y	A
x_5	Y	Y	Y	B

Note that a feature is never removed from a subset. This is important to ensure the correctness of the algorithm. Line 2 shows that feature a_i is added. It may happens that $POS_{B''}(\{d\}) = POS_{B'' \cup \{a_i\}}(\{d\})$, i.e., a_i does not contribute to the positive region. However, a_i is not removed because it may be useful while combined with other features. We introduce the following example to explain the reason.

Example 9. Consider the decision system listed in Table 1. Let $c = [2, 3, 10]$ and $m = 6$. Because $c(a_3) = 10 > m$, a_3 is never selected. We have $POS_{\{a_1\}}(\{d\}) = POS_{\{a_2\}}(\{d\}) = \emptyset$. That is, neither a_1 nor a_2 contributes to the positive region alone. However, $POS_{\{a_1, a_2\}}(\{d\}) = \{x_2, x_3, x_4\}$, hence both a_1 and a_2 are useful. The optimal feature subset is $\{a_1, a_2\}$, which is the output of the algorithm.

In fact, B may contain some redundant features during the algorithm execution. It will eventually replaced by another feature subset with bigger positive region or smaller test cost in Line 10. Example 9 will be discussed further in Section 3.2.

The space complexity of Algorithm 1 is easy to analyze. The algorithm searches in a tree with depth $|C|$ in a depth-first manner. Whenever the backtracking method is invoked there is a need to obtain a new partition of the objects, which takes $O(|U| \times |C|)$ space. Hence the space complexity is

$$O(|C| \times |U| \times |C|) = O(|U| \times |C|^2). \tag{6}$$

Now we analyze the time complexity. The number of feature subsets is $2^{|C|}$. In the worst case all of them are checked. On the other hand, a feature subset is never checked twice. Therefore the number of backtracking steps, namely the number of time the backtracking method is invoked, is bounded by $2^{|C|}$. As indicated by Line 1, each time we need to compute a feature subset with one more feature. In this way, the computation involves splitting the dataset according to the current feature. Respective operation takes $O(|U| \times |V_{a_i}|)$ of time. Let $v_{max} = \max_{a \in C} |V_a|$. The time complexity is

$$O(|U| \times 2^{|C|} \times v_{max}). \tag{7}$$

Unfortunately, the average time complexity is hard to analyze. We will show by experimentation that it is significantly lower than the worst case.

The design of the algorithm is often closely related to the problem definition. Algorithm 1 can be easily obtained from Problem 8. Similarly, the SESRA algorithm [46] has three main steps, as indicated by Definition 7. This phenomenon shows further the influence of the problem viewpoint to the problem definition and the algorithm design.

3.2. The heuristic algorithm

The backtracking algorithm is not scalable. As indicated by Eq. (7), the runtime can be exponential with respect to the number of features in the worst case. Hence we need to design heuristic algorithms for large datasets. We adopt the well known addition–deletion approach [42,73] to design our algorithm, since the deletion approach is inefficient for large datasets [73].

The positive region seems to be a natural heuristic information, however, it may not work on some datasets. Let B be the currently selected feature subset. We would like to select $a_i \in C - B$ if it is informative (i.e., $|POS_{B\cup\{a_i\}} - POS_B|$ is big) and cheap (i.e., $c(a_i)$ is small). Unfortunately, we have counterexamples to this approach. Let us consider Example 9 again. At the very beginning $B = \emptyset$. Since $POS_{B\cup\{a_1\}} = \emptyset$, a_1 has no contribution to the positive region and therefore cannot be selected. For the same reason a_2 is not selected. a_3 cannot be selected due to the test cost constraint. Finally, this approach fails to construct the optimal feature subset $\{a_1, a_2\}$. Such cases happen in applications frequently. We have tested this approach on the datasets listed in 2. In the Voting and Tic-tac-toe datasets [4], no feature alone produces positive region, therefore the approach fails given any test cost setting.

A feasible heuristic information is the information gain [55,64]. Generally, a feature subset with less information entropy tends to produce bigger positive region. Therefore we employ information gain in this paper to design our algorithm. Let $H(Q|P)$ be the conditional information entropy of Q w.r.t. P [64]. Let further $B \subset C$ and $a_i \in C - B$, the information gain of a_i w.r.t. B is

$$f_e(B, a_i) = H(\{d\}|B) - H(\{d\}|B \cup \{a_i\}). \tag{8}$$

It is proven that $|POS_{B\cup\{a_i\}} - POS_B| > 0$ gives $H(\{d\}|B) - H(\{d\}|B \cup \{a_i\}) > 0$. But the reverse does not hold. In other words, information entropy is more sensitive to feature than positive region.

To select the current best feature, both information gain [64] and test cost are taken into consideration. We use the same approach as that in [41] to select the current best test. And the λ-weighted function is defined as

$$f(B, a_i, c) = f_e(B, a_i)c_i^{\lambda}, \tag{9}$$

where λ is a non-positive number. With the introduction of λ, cheaper features are preferred. If $\lambda = 0, f(B, a_i, c) = f_e(B, a_i)$, and the heuristic information coincides with the information gain.

Algorithm 2. The λ-weighted heuristic algorithm

```
Input: $S = (U, C, D, V, I, c)$, $m$
Output: $B \subseteq C$
Method: $\lambda$-weighted-fstc
$B = \emptyset$; //initialize the output
$CA = C$; //unprocessed features
$c_l = m$; //available test cost
//Compute a feature subset with the least information entropy
while ($CA \neq \emptyset$) do
  For any $a \in CA$ satisfying $C(a) \leq c_l$, compute $f(B, a, c)$;
  //Addition
  Select $a' \in CA$ with the maximal $f(B, a', c)$;
  $B = B \cup \{a'\}$; $CA = CA - \{a'\}$; $c_l = c_l - c(a')$;
  //Deletion, remove redundant features from the viewpoint of information entropy
  for (each $a \in B$) do
    if ($H(\{d\}|B - \{a\}) = H(\{d\}|B)$) then
      $B = B - \{a'\}$; //$a'$ is redundant
      $c_l = c_l + c(a')$; //restore the constraint
    end if
  end for
  //Remove features not satisfying the constraint to speed up
  for (each $a \in CA$) do
    if ($c_a > c_l$) then
      $CA = CA - \{a\}$;
    end if
  end for
end while
//Remove redundant features from the viewpoint of positive region
for (each $a \in B$) do
  if ($POS_{B-\{a'\}}(\{d\}) = POS_B(\{d\})$) then
    $B = B - \{a'\}$; //$a'$ is redundant
  end if
end for
return $B$;
```

Table 2
Dataset information.

Name	Domain	$\|C\|$	$\|U\|$	d
Zoo	Zoology	16	101	Type
Voting	Society	16	435	Vote
Tic-tac-toe	Game	9	958	Class
Mushroom	Botany	22	8124	Class
Connect-4	Game	42	67,557	Class

Our algorithm is listed in Algorithm 2. The algorithm first constructs a feature subset meeting the constraint and with minimal information entropy in Lines 4 through 19. Lines 14 through 18 are not necessary, however they help speeding up the algorithm. Then redundant features are removed from the viewpoint of the positive region in Lines 20 through 24.

One may find that the algorithm is successful on Example 9. If we remove x_5 from the dataset, this algorithm also fails. To make the matter worse, the ID3 decision tree encounters the same problem. This might be a drawback of heuristic algorithms compared with exhaustive ones. Fortunately, this extreme case seldom happens in applications. On many UCI datasets we tested, Algorithm 2 never fails to construct a feature subset.

The space complexity of Algorithm 2 is decided by the size of the decision system. It is

$$O(|U| \times |C|). \tag{10}$$

Now we analyze the time complexity. In the worst case, the **while** loop indicated by Line 4 would execute $|C|$ times, and each time all remaining features are checked in Line 5. Line 5 is executed at most $\sum_{i=0}^{|C|-1}(|C| - i) = O(|C|^2)$ times. Since $f(B, a, c)$ is based on the positive region, similar to the analysis in Section 3.1, the time complexity is

$$O(|U| \times |C|^2 \times v_{max}). \tag{11}$$

In applications, it is hard for the user or even the expert to set a rational λ. To make the matter worse, the best λ does not always produce the best result. We can adopt the competition strategy working as follows. First, it specifies a set of λ values, then it obtains corresponding feature subsets using Algorithm 2, finally it chooses the feature subset with the maximal positive region and the minimal test cost. Since feature subsets produced by different λ values compete against each other with only one winner, this strategy is called the *competition strategy* [41].

Formally, let B_λ be the feature subset constructed by Algorithm 2 using the exponential λ. With Λ the set of user-specified λ values,

$$POS_\Lambda = \max_{\lambda \in \Lambda} POS_{B_\lambda}(\{d\}) \tag{12}$$

is the maximal positive region that can be obtained with the competition strategy. This process requires the algorithm to be run $|\Lambda|$ times and the time complexity would be $O(|\Lambda| \times |U| \times |C|^2 \times v_{max})$ instead. It is acceptable for relatively small $|\Lambda|$. We will show that setting Λ is easy in Section 4.3.

4. Experiments

The main purpose of our experiments is to answer the following questions.

1. Is the backtracking algorithm efficient?
2. Is the heuristic algorithm effective?
3. Is there an optimal setting of λ for any dataset?
4. Is the extra computation time consumed by the competition strategy worthwhile?

4.1. Datasets

We deliberately select five datasets from the UCI Repository of Machine Learning Databases [4]. Their basic information is listed in Table 2, where $|C|$ is the number of features, $|U|$ is the number of instances, and d is the name of the decision.

There are a number of notes to make. While counting the number of features, the decision is not included. Missing values (e.g., those appearing in the Voting dataset) are treated as one particular value. That is, ? is equal to itself, and unequal to any other value. The “animal name" feature is not useful in the Zoo dataset, and we simply remove it.

Most datasets from the UCI library [4] do not provide test cost information. For statistical purposes, we need to produce them. Different test cost distributions correspond to different applications. Three distributions, namely uniform distribution, normal distribution, and Pareto distribution have been discussed in [41]. For simplicity, this paper only employs the uniform distribution to generate random test cost in [1 . . . 100]. According to Definition 5, two TCI-DS are different once their test cost settings are different. In this sense, we can produce as many TCI-DS as needed from a given DS.

Table 3
Backtracking steps on four datasets (with 100 test cost settings).

Dataset	$\|C\|$	$2^{\|C\|}$	$\|B\|$			Backtracking steps		
			Min	Max	Aver.	Min	Max	Aver.
Zoo	16	65,536	4	6	4.74	132	4089	1112
Voting	16	65,536	7	9	8.23	8139	46,421	24,354
Tic-tac-toe	9	512	6	7	6.70	271	439	386
Mushroom	22	4,194,304	3	6	4.31	26	4899	725

Table 4
Runtime (ms) on four datasets (mean values for 100 test cost settings).

Dataset	SESRA	SESRA*	Backtracking	Heuristic
Zoo	50	48	7	2
Voting	5334	2498	485	18
Tic-tac-toe	167	39	28	26
Mushroom	3661	857	367	180

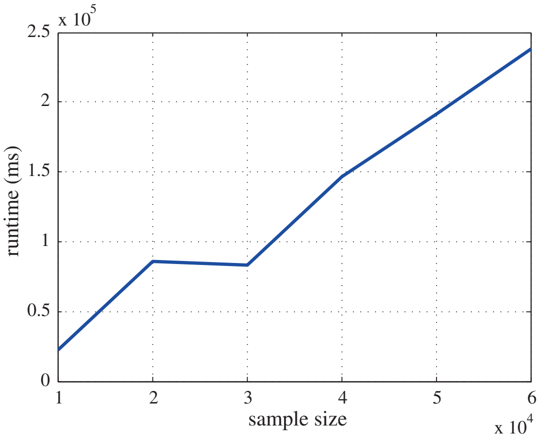

Fig. 1. The runtime of the heuristic algorithm on the Connect-4 dataset.

4.2. Efficiency of the algorithms

We need to know the efficiency of the backtracking algorithm from three viewpoints. The first is the average time complexity. We need to know whether or not the number of backtracking steps is exponential with respect to the number of features. The second is the time taken for small or medium-sized data. In fact, diagnosis data for one particular disease in a hospital may contain only a few thousands of instances. For those datasets, an optimal solution is always required. The third is the runtime compared with other exhaustive approaches. The backtracking algorithm is compared with SESRA and SESRA* proposed in [46]. SESRA is based on Definition 7, and SESRA* is an enhanced version.

Table 3 shows the number of backtracking steps, namely how many times the backtracking method is invoked. Let BS denote this number. $2^{|C|}$ is the size of the backtracking tree, hence it is also the upper bound of BS. For the Voting dataset, $|C| = 16$ and sometimes $|B| = 9$. Therefore the maximal BS can be 46,421, which is close to $2^{|C|}$ = 65,536. This indicates that sometimes BS can be exponential with respect to $|C|$. In contrast, For the Mushroom dataset, $|C| = 22$ and sometimes $|B| = 6$. The maximal BS is only 4899, which is significantly smaller than $2^{|C|}$ = 4,194,304. In one word, BS is relevant to not only $|C|$, but also $|B|$.

Table 4 compares the performance of the backtracking algorithm with SESRA and SESRA* [46] in terms of the runtime. The backtracking algorithm only takes 367 ms and 485 ms on the Mushroom and Voting datasets, respectively. In other words, it is appropriate for many real applications. Moreover, the backtracking algorithm stably outperforms SESRA and SESRA*. Only about 1/10 time is taken on the Tic-tac-toe and Mushroom datasets compared with SESRA. These results show further the advantage of the CSP viewpoint.

For convenience, the runtime of the heuristic algorithm is also listed in Table 4. The heuristic algorithm is always more efficient than exhaustive algorithms. The efficiency difference becomes significant when the runtime of exhaustive algorithms is long. Moreover, the efficiency depends more on the dataset size instead of $|B|$. To sum up, the heuristic algorithm can deal with larger datasets compared with exhaustive algorithms.

For the Connect-4 dataset, only the heuristic algorithm is tested. This is because that all our exhaustive algorithms fail on this dataset. We randomly sample the dataset to produce subtables with different number of objects. Then the heuristic algorithm is run on each subtable. The runtime is shown in Fig. 1. Here we observe that the runtime of the algorithm is proportional to the number of objects. In other words, the heuristic algorithm is scalable.

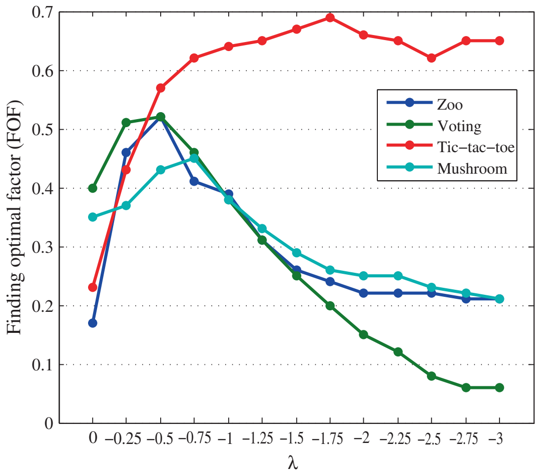

Fig. 2. The probability of finding the optimal feature subset for given λ.

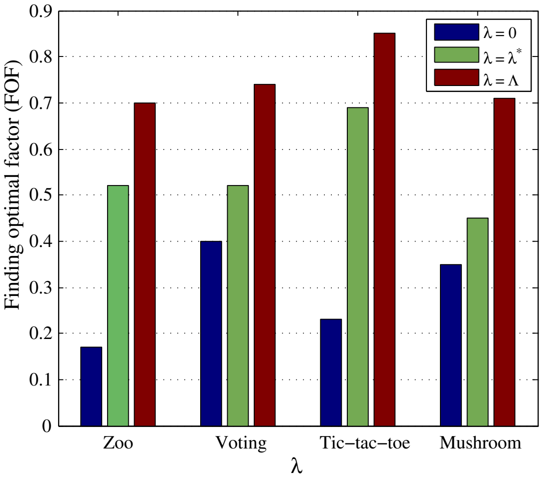

Fig. 3. The probability of finding the optimal feature subset.

4.3. *Effectiveness of the heuristic algorithm*

We compare the performance of the three approaches mentioned in Section 3.2. All three are based on Algorithm 2. The first approach, called the non-weighting approach, is implemented by setting $\lambda = 0$. The second approach, called the best λ approach, chooses the best λ value in $\Lambda = \{0, -0.25, -0.5, \dots, -3\}$. The third approach is the competition strategy based Λ as discussed in Section 3.2.

We now look at the influence of the λ setting. Fig. 2 shows the probability of finding the optimal feature subset for given λ. Although -0.75 seems a reasonable value, there does not exist an optimal setting of λ for all datasets. In other words, λ is hard to specify.

General results are depicted in Fig. 3, from which we observe the following. First, the approach without taking into considering the test cost performs poorly. In most cases it cannot find the optimal feature subset. Second, if we specify λ appropriately, namely $\lambda = \lambda^*$, the results are more acceptable. It is more likely to find the optimal feature subset. However, as discussed earlier, we often have no idea how to specify it. Third, the performance of the competition strategy is much better than the other two. In more than 70% cases it produces the optimal feature subset. Moreover, the user does not have to know the optimal setting of λ. In one word, the extra computation resource consumed by the competition strategy is worthwhile.

5. The CSP viewpoint to feature selection

Problems 3, 4, 6 and 8 provide the CSP viewpoint to feature selection. Most existing feature selection problems in rough sets can be viewed extensions of Problem 4 in one or more of the following aspects: input, output, constraint, and optimization objective. We analyze them from each aspect as follows.

First, there are some extensions concerning the input data model. Since the data model is essential, these extensions often require extensions of the Pawlak rough set.

1. Some conditional features are numeric. Numeric data are quite different from symbolic ones which are employed in Pawlak rough sets [51]. Coverings, instead of partitions, can be formed according to feature sets. Covering-based rough sets [66,84–86] deal with reduction of coverings. The neighborhood rough set model [19–22] generates neighborhood systems on such data.
2. The data are uncertain. The uncertainty of data may be caused by noise, observational error, etc. [7]. The error range based covering rough set model [47] was proposed to deal with observational error. Another well known data model might be interval-valued fuzzy sets [12], which has been studied through rough sets [11].
3. There are external information on features and feature subsets [74]. Some information are subjective and can be expressed by user preference. For example, features are ranked by the user, or even directly specified by an expert [44]. Other information are objective. For example, there is a weight or test cost for each feature [42,74]. There are a number of possible extensions to the weight computation of an feature subset. These are additive, average, maximal, minimal extensions [74]. In [42], six data models concerning test cost and relationships among features are defined. Test-cost-sensitive attribute reduction problems [17,47] can be defined according on these models.
4. There are external information on classification [28]. The most widely adopted information might be misclassification cost [62,83]. DTRS [29,71,73] consider loss functions concerning different classifications. These classifications correspond to positive, negative and boundary rules. There are cost for both misclassifications and correct classifications.
5. There are external information on both conditions and classifications. In applications such as clinic systems, both test costs and misclassification costs exist [62]. This issue is addressed in [47].

Second, there are some extensions concerning the output. People considered generalized reduct problems, such as attribute value reduction [52], discretization [49,61], symbolic value partition [43]. Since features are transformed or combined, these problems should be called *feature extraction* instead [15,23].

Third, there are some extensions concerning the constraint. Many of them are still expressed with the same form as Problem 4. However, the definitions of the positive region are different due to the change of the input data model. Others are expressed with different forms.

1. The computation of the positive region follows DTRS models [67,71,73]. In DTRS, parameters γ, β and δ are used to define positive regions. They are in turn computed based on a set of loss functions according to the Bayesian decision procedure. The major advantage is that parameters are not set by the user subjectively. Therefore the models have good semantics and wide applications.
2. The computation of the positive region follows the variable precision rough set model [87], or the Bayesian rough set model [58]. There is a user-specified parameter β to indicate the admissible classification error. Pawlak rough sets can be viewed a special case of variable precision rough sets where $\beta = 0$. This extension has inspired fruitful research works concerning probabilistic rough sets [13,35,32,70]. β-lower distribution and β-upper distribution [38] have been more closely studied.
3. The computation of the positive region follows the neighborhood rough set model [20–22] or the error range based covering rough set model [47]. In the neighborhood rough set model [20–22], positive regions also rely on a user specified parameter δ, which is the distance threshold. In the error range based covering rough set model [47], positive regions also rely on error ranges of data. Error ranges are determined by testing instruments and therefore they are objective.
4. The constraint is conditional information entropy [57,63,36]. It is expressed by $H(B|\{d\}) = H(C|\{d\})$ where $H(B|\{d\})$ denotes the conditional information entropy of B with respect to d. The conditional information entropy constraint is stricter than the positive region constraint. That is, the feature subset meeting the positive region constraint may not meet the conditional information entropy constraint. While the reverse does not hold. These two constraints are equivalent if and only if the decision system is consistent [39].

Fourth, there are some extensions concerning the optimization objective.

1. Minimize the cost. In test cost sensitive decision systems, the objective is to minimize the total test cost [41]. In misclassification cost sensitive decision systems, the objective is to minimize the average misclassification cost [31, 71,73], the risk [30,35], or the weighted accuracy and weighted error [82]. In decision systems with both test cost and misclassification cost, the objective is to minimize the total cost [45].

2. Minimize the feature space $\prod_{a\in B}|V_a|$. For the minimal reduct problem, features with more values are more likely to be selected. These features, however, have weaker generalization ability than features with fewer values. The new objective can help amend this drawback. When the domains of features have the same size, the new objective coincides with Problem 4 [40].
3. Maximize the stability. Dynamic reducts [3] are stable in the process of decision table sampling. Decision rules computed from dynamic reducts are more reliable. Parallel reducts [9] follow the same idea.
4. Maximize the margin. A margin is a geometric measure for evaluating the confidence of a classifier with respect to its decision [6,10]. Unlike other metric such as positive region or conditional information entropy, this measure is not monotonic. That is, it may increase or decrease when more features are selected.

Most problems mentioned above are no longer reduct problems. When the input is changed, the indiscernibility relation may not exist. One can only consider weaker relations such as the similarity relation [59]. When the constraint is changed, the positive region is not computed, or computed not in the Pawlak approach (see, e.g., [21,47]). Reducts subject to the conditional information entropy constraint may not be a Pawlak reduct. When the optimization objective is changed, the optimal reduct may not be minimal. Feature subset with the minimal total cost [47] may not be a reduct at all.

From these extensions, many meaningful new problems can be identified. A few of them are listed as follows.

1. Feature selection under DTRS with test cost. Note again the external information in DTRS cannot be expressed by a misclassification matrix. Test cost is also one kind of external information. By considering more external information, the problem is more interesting and challenging.
2. Feature selection with positive region constraint. To have an even simpler representation, we may require the positive region to be preserved to a certain degree. For example, the feature subset should have a positive region more than 95% of the original. Note that this problem is different from the variable precision rough set model [87] where the definition of positive region is changed. Their motivations are, however, quite similar in that they all deal with the overfitting issue.
3. Minimal test cost feature selection with positive region constraint. This problem differs from the last one in that the objective is to find a feature subset with least cost. It is a hybrid of the last problem and the MTR problem. It can be also viewed a dual problem of the FSTC problem.

Some of these problems are new combinations of existing extensions, some involve new extensions. We observe that the number of possible combinations is big, and many of them have certain application areas. In other words, much research issues are opened from the CSP viewpoint.

6. Conclusions and further works

This paper proposed a new feature selection problem concerning the test cost constraint. The new problem has a wide application area because the resource one can afford is often limited. Both backtracking and heuristic algorithms were designed for it. Experimental results showed the efficiency of the backtracking algorithm compared with existing ones, and the effectiveness the competition strategy based on the λ-weighted heuristic algorithm. It should be noted that with the competition strategy, we do not have to know the optimal setting of λ. Instead, we can specify a set of λ values which are valid for any dataset.

A more important contribution of the paper is the CSP viewpoint to feature selection in rough sets. From this viewpoint, most feature selection problems are natural generalizations of the minimal reduct problem. This viewpoint helps us to identify some other meaningful problems from the following aspects: input, output, constraint, and optimization objective. In summary, this paper has indicated important research trends concerning feature selection beyond rough sets.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61170128, the Natural Science Foundation of Fujian Province, China under Grant No. 2012J01294, and Fujian Province Foundation of Higher Education under Grant No. JK2012028.

References

[1] N. Azam, J.T. Yao, Multiple criteria decision analysis with game-theoretic rough sets, in: Proceedings of Rough Sets and Current Trends in Computing, LNCS, vol. 7414, Springer,Berlin/Heidelberg, 2012, pp. 399–408.
[2] N. Azam, J.T. Yao, Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets, International Journal of Approximate Reasoning, this issue.
[3] J.G. Bazan, A. Skowron, Dynamic reducts as a tool for extracting laws from decision tables, in: Proceedings of the 8th International Symposium on Methodologies for Intelligent Systems, Springer,Berlin/Heidelberg, 1994, pp. 346–355.
[4] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/∼mlearn/mlrepository.html>.

[5] Y. Chen, D. Miao, R. Wang, K. Wu, A rough set approach to feature selection based on power set tree, Knowledge-Based System 24 (2011) 275–281.

[6] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.

[7] J. Dai, W. Wang, Q. Xua, H. Tian, Uncertainty measurement for interval-valued decision systems based on extended conditional entropy, Knowledge-Based Systems 27 (2012) 443–450.

[8] J. Dai, Q. Xu, Approximations and uncertainty measures in incomplete information systems, Information Sciences 198 (2012) 62–80.

[9] D. Deng, Parallel reduct and its properties, in: Proceedings of IEEE Granular Computing, 2009, pp. 121–125.

[10] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection – theory and algorithms, in: Proceedings of the 21st International Conference on Machine Learning, ICML, ACM, 2004, pp. 43–50.

[11] Z. Gong, B. Sun, D.G. Chen, Rough set theory for the interval-valued fuzzy information systems, Information Sciences 178 (8) (2008) 1968–1985.

[12] B. Gorzafczary, Interval-valued fuzzy controller based on verbal modal of object, Fuzzy Sets and Systems 28 (1988) 45–53.

[13] J.W. Grzymala-Busse, P.G. Clark, M. Kuehnhausen, Generalized probabilistic approximations of incomplete data, International Journal of Approximate Reasoning, this issue.

[14] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[15] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag, New York, Inc., 2006.

[16] H.P. He, F. Min, Accumulated cost based test-cost-sensitive attribute reduction, in: Proceedings of the 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI, vol. 6743, Springer,Berlin/Heidelberg, 2011, pp. 244–247.

[17] H.P. He, F. Min, W. Zhu, Attribute reduction in test-cost-sensitive decision systems with common-test-costs, in: Proceedings of the 3rd International Conference on Machine Learning and Computing, v1, IEEE, 2011, pp. 432–436.

[18] J.P. Herbert, J.T. Yao, Game-theoretic Rough sets, Fundamenta Informaticae 108 (2011) 267–286.

[19] Q.H. Hu, J.F. Liu, D.R. Yu, Mixed feature selection based on granulation and approximation, Knowledge-Based Systems 21 (2008) 294–304.

[20] Q.H. Hu, W. Pedrycz, D.R. Yu, J. Lang, Selecting discrete and continuous features based on neighborhood decision error minimization, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 40 (1) (2010) 37–50.

[21] Q.H. Hu, D.R. Yu, J.F. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (18) (2008) 3577–3594.

[22] Q.H. Hu, D.R. Yu, Z. Xie, Numerical attribute reduction based on neighborhood granulation and rough approximation, Journal of Software 19 (3) (2008) 640–649 (in Chinese).

[23] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2) (1979) 153–158.

[24] R. Jensen, Q. Shen, A. Tuson, Finding rough set reducts with SAT, in: Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, LNAI, vol. 3641, Springer,Berlin/Heidelberg, 2005, pp. 194–203.

[25] X.Y. Jia, W.W. Li, L. Shang, J.J. Chen, An optimization viewpoint of decision-theoretic Rough set model, in: Proceedings of Rough Sets and Knowledge Technology, LNAI, vol. 6954, Springer,Berlin/Heidelberg, 2011, pp. 457–465.

[26] X.Y. Jia, Z.M. Tang, W.H. Liao, L. Shang, On an optimization representation of decision-theoretic rough set model, International Journal of Approximate Reasoning. <http://dx.doi.org/10.1016/j.ijar.2013.02.010>.

[27] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI, John Wiley & Sons Ltd, 1992, pp. 129–134.

[28] D.L.T.R. Li, D.C. Liang, Incorporating logistic regression to decision-theoretic rough sets for classifications, International Journal of Approximate Reasoning. <http://dx.doi.org/10.1016/j.ijar.2013.02.013>.

[29] H.X. Li, Y.Y. Yao, X.Z. Zhou, B. Huang, A two-phase model for learning rules from incomplete data, Fundamenta Informaticae 94 (2009) 219–232.

[30] H.X. Li, X.Z. Zhou, Risk decision making based on decision-theoretic rough set: a three-way view decision model, International Journal of Computational Intelligence Systems 4 (1) (2011) 1–11.

[31] H.X. Li, X.Z. Zhou, J.B. Zhao, D. Liu, Attribute reduction in decision-theoretic rough set model: a further investigation, in: Proceedings of Rough Sets and Knowledge Technology, LNCS, vol. 6954, Springer,Berlin/Heidelberg, 2011, pp. 466–475.

[32] T.J. Li, X.P. Yang, An axiomatic characterization of probabilistic rough sets, International Journal of Approximate Reasoning. <http://dx.doi.org/10.1016/j.ijar.2013.02.012>.

[33] J.Y. Liang, F. Wang, C.Y. Dang, Y.H. Qian, An efficient rough feature selection algorithm with a multi-granulation view, International Journal of Approximate Reasoning 53 (2012) 912–926.

[34] D. Liu, T.R. Li, H.X. Li, A multiple-category classification approach with decision-theoretic Rough sets, Fundamenta Informaticae 155 (2–3) (2012) 173–188.

[35] D. Liu, T.R. Li, D. Ruan, Probabilistic model criteria with decision-theoretic Rough sets, Information Sciences 181 (2011) 3709–3722.

[36] Q.H. Liu, F. Li, F. Min, M. Ye, G.W. Yang, An efficient reduction algorithm based on new conditional information entropy, Control and Decision 20 (8) (2005) 878–882 (in Chinese).

[37] P. Maji, S. Paul, Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data, International Journal of Approximate Reasoning 52 (2011) 408–426.

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

[39] D.Q. Miao, Y. Zhao, Y.Y. Yao, H.X. Li, F.F. Xu, Relative reducts in consistent and inconsistent decision tables of the pawlak rough set model, Information Sciences 179 (24) (2009) 4140–4150.

[40] F. Min, X.H. Du, H. Qiu, Q.H. Liu, Minimal attribute space bias for attribute reduction, in: Proceedings of Rough Set and Knowledge Technology, LNCS, vol. 4481, Springer,Berlin/Heidelberg, 2007, pp. 379–386.

[41] F. Min, H.P. He, Y.H. Qian, W. Zhu, Test-cost-sensitive attribute reduction, Information Sciences 181 (2011) 4928–4942.

[42] F. Min, Q.H. Liu, A hierarchical model for test-cost-sensitive decision systems, Information Sciences 179 (2009) 2442–2452.

[43] F. Min, Q.H. Liu, C.L. Fang, Rough sets approach to symbolic value partition, International Journal of Approximate Reasoning 49 (2008) 689–700.

[44] F. Min, Q.H. Liu, H. Tan, L.T. Chen, The *M*-relative reduct problem, in: Proceedings of Rough Set and Knowledge Technology, LNAI, vol. 4062, Springer,Berlin/Heidelberg, 2006, pp. 170–175.

[45] F. Min, W. Zhu, Minimal cost attribute reduction through backtracking, in: Proceedings of International Conference on Database Theory and Application, FGIT-DTA/BSBT, vol. 258, CCIS, Springer,Berlin/Heidelberg, 2011, pp. 100–107.

[46] F. Min, W. Zhu, Optimal sub-reducts with test cost constraint, in: Proceedings of Rough Set and Knowledge Technology, LNAI, vol. 6954, Springer,Berlin/Heidelberg, 2011, pp. 57–62.

[47] F. Min, W. Zhu, Attribute reduction of data with error ranges and test costs, Information Sciences 211 (2012) 48–67.

[48] F. Min, W. Zhu, H. Zhao, G.Y. Pan, J.B. Liu, Z.L. Xu, Coser: cost-senstive rough sets, Springer,Berlin/Heidelberg, 2012. <http://grc.fjzs.edu.cn/~fmin/coser/>.

[49] H.S. Nguyen, Discretization problem for rough sets methods, in: Rough Sets and Current Trends in Computing, LNCS, vol. 1424, Springer,Berlin/Heidelberg, 1998, pp. 545–552.

[50] G.Y. Pan, F. Min, W. Zhu, A genetic algorithm to the minimal test cost reduct problem, in: Proceedings of IEEE International Conference on Granular Computing, 2011, pp. 539–544.

[51] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (1982) 341–356.

[52] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Boston, 1991.

[53] J. Qian, D. Miao, Z. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, International Journal of Approximate Reasoning 52 (2) (2011) 212–230.

[54] Y.H. Qian, H. Zhang, Y.L. Sang, J.Y. Liang, Multigranulation decision-theoretic rough sets, International Journal of Approximate Reasoning. <http://dx.doi.org/10.1016/j.ijar.2013.03.004>.
[55] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[56] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: Intelligent Decision Support, 1992, pp. 331–362.
[57] D. Ślęzak, Approximate entropy reducts, Fundamenta Informaticae 53 (3–4) (2002) 365–390.
[58] D. Ślęzak, W. Ziarko, The investigation of the bayesian rough set model, International Journal of Approximate Reasoning 40 (2006) 81–91.
[59] R. Slowinski, D. Vanderpooten, A generalized definition of rough approximations based on similarity, IEEE Transactions on Knowledge and Data Engineering 12 (2) (2000) 331–336.
[60] R. Susmaga, Computation of minimal cost reducts, in: Z. Ras, A. Skowron (Eds.), Foundations of Intelligent Systems, LNCS, vol. 1609, Springer, Berlin/Heidelberg, 1999, pp. 448–456.
[61] D. Tian, X. jun Zeng, J. Keane, Core-generating approximate minimum entropy discretization for rough set feature selection in pattern classification, International Journal of Approximate Reasoning 52 (2011) 863–880.
[62] P.D. Turney, Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm, Journal of Artificial Intelligence Research 2 (1995) 369–409.
[63] G. Wang, Attribute core of decision table, in: Proceedings of Rough Sets and Current Trends in Computing, LNCS, vol. 2475, Springer,Berlin/Heidelberg, 2002, pp. 213–217.
[64] G. Wang, H. Yu, D. Yang, Decision table reduction based on conditional information entropy, Chinese Journal of Computers 2 (7) (2002) 759–766.
[65] J. Wang, J. Wang, Reduction algorithms based on discernibility matrix: the ordered attributes method, Journal of Computer Science and Technology 16 (6) (2001) 489–504.
[66] T. Yang, Q.G. Li, Reduction about approximation spaces of covering generalized rough sets, International Journal of Approximate Reasoning 51 (3) (2010) 335–345.
[67] X.P. Yang, J.T. Yao, Modelling multi-agent three-way decisions with decision-theoretic Rough sets, Fundamental Informaticae 115 (2–3) (2012) 157–171.
[68] J.T. Yao, M. Zhang, Feature selection with adjustable criteria, in: Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, LNAI, vol. 3641, Springer,Berlin/Heidelberg, 2005, pp. 204–213.
[69] Y.Y. Yao, A partition model of granular computing, Transactions on Rough Sets I 3100 (2004) 232–253.
[70] Y.Y. Yao, Probabilistic rough set approximations, International Journal of Approximate Reasoning 49 (2) (2008) 255–271.
[71] Y.Y. Yao, S. Wong, A decision theoretic framework for approximating concepts, International Journal of Man-machine Studies 37 (1992) 793–809.
[72] Y.Y. Yao, Y. Zhao, Attribute reduction in decision-theoretic rough set models, Information Sciences 178 (17) (2008) 3356–3373.
[73] Y.Y. Yao, Y. Zhao, J. Wang, On reduct construction algorithms, Transactions on Computational Science 2 (2008) 100–117.
[74] Y.Y. Yao, Y. Zhao, J. Wang, S.Q. Han, A model of user-oriented reduct construction for machine learning, Transactions on Rough Sets IV 5084 (2008) 332–351.
[75] D.Y. Ye, Z.J. Chen, A new discernibility matrix and the computation of a core, Acta Electronica Sinica 7 (2002) 1086–1088.
[76] H. Yu, Z.G. Liu, G.Y. Wang, An automatic method to determine the number of clusters using decision-theoretic rough set, International Journal of Approximate Reasoning, this issue.
[77] H. Yu, G.Y. Wang, F.K. Lan, Solving the attribute reduction problem with ant colony optimization, Transactions on Rough Sets XIII 6499 (2011) 240–259.
[78] S.C. Zhang, Cost-sensitive classification with respect to waiting cost, Knowledge-Based Systems 23 (5) (2010) 369–378.
[79] H. Zhao, F. Min, W. Zhu, Test-cost-sensitive attribute reduction based on neighborhood rough set, in: Proceedings of the 2011 IEEE International Conference on Granular Computing, 2011, pp. 802–806.
[80] Y. Zhao, F. Lou, S. Wong, Y.Y. Yao, A general definition of an attribute reduct, in: Proceedings of Rough Sets and Knowledge Technology, LNAI, vol. 4481, Springer,Berlin/Heidelberg, 2007, pp. 101–108.
[81] N. Zhong, J.Z. Dong, S. Ohsuga, Using rough sets with heuristics to feature selection, Journal of Intelligent Information Systems 16 (3) (2001) 199–214.
[82] B. Zhou, A comparison study of cost-sensitive classifier evaluations, in: Proceedings of the International Conference on Brain Informatics, LNCS, vol. 7670, Springer,Berlin/Heidelberg, 2012, pp. 360–371.
[83] Z. Zhou, X. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Transactions on Knowledge and Data Engineering 18 (1) (2006) 63–77.
[84] W. Zhu, Topological approaches to covering rough sets, Information Sciences 177 (6) (2007) 1499–1508.
[85] W. Zhu, F. Wang, Reduction and axiomization of covering generalized rough sets, Information Sciences 152 (1) (2003) 217–230.
[86] W. Zhu, F. Wang, On three types of covering rough sets, IEEE Transactions on Knowledge and Data Engineering 19 (8) (2007) 1131–1144.
[87] W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences 46 (1) (1993) 39–59.