

Research Article

Cost-Sensitive Attribute Reduction in Decision-Theoretic Rough Set Models

Shujiao Liao,^{1,2} Qingxin Zhu,¹ and Fan Min³

¹ School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

² School of Mathematics and Statistics, Minnan Normal University, Zhangzhou 363000, China

³ School of Computer Science, Southwest Petroleum University, Chengdu 610500, China

Correspondence should be addressed to Fan Min; minfanphd@163.com

Received 23 October 2013; Revised 12 January 2014; Accepted 12 January 2014; Published 27 February 2014

Academic Editor: Matjaž Perc

Copyright © 2014 Shujiao Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the theory of decision-theoretic rough set and its applications have been studied, including the attribute reduction problem. However, most researchers only focus on decision cost instead of test cost. In this paper, we study the attribute reduction problem with both types of costs in decision-theoretic rough set models. A new definition of attribute reduct is given, and the attribute reduction is formulated as an optimization problem, which aims to minimize the total cost of classification. Then both backtracking and heuristic algorithms to the new problem are proposed. The algorithms are tested on four UCI (University of California, Irvine) datasets. Experimental results manifest the efficiency and the effectiveness of both algorithms. This study provides a new insight into the attribute reduction problem in decision-theoretic rough set models.

1. Introduction

We are involved in decision making all the time. Most of the decisions are based on a group of criteria. In this case, decision making is often aimed at finding a proper balance or tradeoff among multiple criteria. There are a series of methods for analyzing multicriteria decision making, such as game theory. Game theory is an effective mathematical method for formulating decision problems as competition between several entities [1]. These entities, or players, aspire to either achieve a dominant position over the other players or cooperate with each other in order to find a position that benefits all [2]. Researchers have accumulated a vast literature on game theory and its applications. For example, recent advances in the study of evolutionary games are reviewed in [3–5], and some strategies in the spatial ultimatum game are discussed in [6, 7], and so on. However, most of these studies do not consider attribute reduction, which can significantly reduce the computation complexity.

Different from the works mentioned above, in rough set theory, attribute reduction is an important concept. It supports the wide applications of rough sets. Moreover, classical

rough sets [8–10] and their extensions [11–15] can be used in conflict analysis [16], a field related to decision making and game theory. Decision-theoretic rough sets (DTRS) [12, 13] may be particularly relevant to decision making and benefit from some new insights provided by game theory. In the rough set theory, a concept is usually described by three classification regions: positive region, boundary region, and negative region. The three regions in DTRS are systematically calculated based on a set of loss functions according to Bayesian decision procedure. The loss functions can be interpreted based on practical notions of costs and risks. In DTRS models, an object is classified into a particular region because the cost of classifying it into the region is less than that of classifying it into other regions. The expected cost of classifying a set of objects is called decision cost.

Generally speaking, attribute reduction can be interpreted as a process of finding the minimal set of attributes that can preserve or improve one or several criteria. The minimal set of attributes is called an attribute reduct. Some researchers have investigated the attribute reduction problem in DTRS models. Most of them addressed the problem based on the preservation or extension of the positive region or

the nonnegative region [17–19]. However, for DTRS, the regions are nonmonotonic with respect to the set inclusion of attributes [18–20], so it is difficult to evaluate and interpret region-based attribute reduction. To tackle the problem, minimal-decision-cost attribute reduction was discussed in [20]. However, most existing studies of attribute reduction in DTRS only concern decision costs but not test costs.

Test cost is the time, money, or other resources one pays for obtaining a data item of an object. Most of the existing attribute reduction problems assume that the data are already stored in datasets and available without charge. However, data are often not free in reality. Recently, the topic of test costs has drawn our attention due to its broad applications. According to the data models constructed in [21], the issues of test-cost-sensitive attribute reduction have been studied based on classical rough sets [22, 23], neighborhood rough sets [24], covering rough sets [25, 26], and so forth. In these works, both backtracking and heuristic algorithms have been implemented through an open source software Coser [27]. Unfortunately, few works have addressed attribute reduction with test cost in the context of DTRS.

In this paper, we study the cost-sensitive attribute reduction problem for DTRS through considering the tradeoff between test costs and decision costs, which is remarkably related to decision making and game theory. Since the purpose of decisions making is to minimize the cost, the process of attribute reduction should help in minimizing the total cost, namely, the summation of test cost and decision cost. A decreasing average-total-cost attribute reduct is defined, which ensures that the total cost will be decreased or unchanged for decisions making by using the reduct. In view of this, a minimal average-total-cost reduct (MACR) in DTRS models is introduced. An optimization problem is constructed in order to minimize the average total cost. It is a generalization of the minimal-decision-cost attribute reduction problem discussed in [20].

Both backtracking and heuristic algorithms are proposed to deal with the new attribute reduction problem. The backtracking algorithm is designed to find an optimal reduct for small datasets. However, for large datasets, it is not easy to find a minimal cost attribute subset. Therefore, we propose a heuristic algorithm to deal with this problem. To study the performance of both algorithms, experiments are undertaken on four datasets from the UCI library [28] through the software Coser. Experimental results show that the efficiency of the backtracking algorithm is acceptable, especially when the loss functions are not much more than the test costs, while the heuristic algorithm is rather efficient, and it can generate a minimal total cost reduct in most cases. Even if the reduct is not optimal sometimes, it is still acceptable from a statistical perspective. Moreover, both algorithms perform well on classification accuracy with CART and RBF-kernel SVM classifiers. Meanwhile, the number of selected attributes is effectively reduced by the two algorithms.

The rest of the paper is organized as follows. In Section 2, we review the main ideas of DTRS. Section 3 gives a detailed explanation of the minimal-total-cost attribute reduction in DTRS models. An optimization problem is proposed. In Section 4, we present a backtracking algorithm and

a heuristic algorithm to address the optimization problem. Experimental settings and results are discussed in Section 5. Section 6 concludes and suggests further research trends.

2. Decision-Theoretic Rough Set Models

In this section, we review some basic notions of DTRS model [12, 13, 17], which presents a theoretical basis for our method.

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

$$S = (U, C, D, V = \{V_a \mid a \in C \cup D\}, I = \{I_a \mid a \in C \cup D\}), \quad (1)$$

where U is a finite nonempty set of objects called the universe, C is the set of conditional attributes, D is the set of decision attributes with only discrete values, V_a is the set of values for each $a \in C \cup D$, and $I_a : U \rightarrow V_a$ is an information function for each $a \in C \cup D$.

In a decision system, given a set of conditional attributes $A \subseteq C$, the equivalence class of an object x with respect to A , namely, $\{y \in U \mid I_a(x) = I_a(y), \forall a \in A\}$, is denoted by $[x]_A$ or $[x]$, if it is understood. In DTRS models, the set of states $\Omega = \{X, X^c\}$ indicates that an object is in a decision class X and not in X , respectively. The probabilities for these two complement states can be denoted by $P(X \mid [x]) = |X \cap [x]|/|[x]|$ and $P(X^c \mid [x]) = 1 - P(X \mid [x])$. With respect to the three regions: positive region $\text{POS}(X)$, boundary region $\text{BND}(X)$, and negative region $\text{NEG}(X)$, the set of actions regarding the state X is given by $\mathcal{A} = \{a_P, a_B, a_N\}$, where a_P, a_B, a_N represent the three actions of classifying an object x into the three regions, respectively. Let $\lambda_{PP}, \lambda_{BP}$, and λ_{NP} denote the cost incurred for taking actions a_P, a_B , and a_N , respectively, when an object belongs to X , and $\lambda_{PN}, \lambda_{BN}$, and λ_{NN} denote the cost incurred for taking the same actions when the object does not belong to X . The loss functions regarding the states X and X^c can be expressed as a 2×3 matrix given in Table 1.

Based on the loss functions, the expected costs of taking different actions for objects in $[x]$ can be expressed as

$$\begin{aligned} \mathfrak{R}(a_P \mid [x]) &= \lambda_{PP}P(X \mid [x]) + \lambda_{PN}P(X^c \mid [x]), \\ \mathfrak{R}(a_B \mid [x]) &= \lambda_{BP}P(X \mid [x]) + \lambda_{BN}P(X^c \mid [x]), \\ \mathfrak{R}(a_N \mid [x]) &= \lambda_{NP}P(X \mid [x]) + \lambda_{NN}P(X^c \mid [x]). \end{aligned} \quad (2)$$

The Bayesian decision procedure leads to the following minimal-risk decision rules:

- (P) If $\mathfrak{R}(a_P \mid [x]) \leq \mathfrak{R}(a_B \mid [x])$ and $\mathfrak{R}(a_P \mid [x]) \leq \mathfrak{R}(a_N \mid [x])$, decide $[x] \subseteq \text{POS}(X)$;
- (B) If $\mathfrak{R}(a_B \mid [x]) \leq \mathfrak{R}(a_P \mid [x])$ and $\mathfrak{R}(a_B \mid [x]) \leq \mathfrak{R}(a_N \mid [x])$, decide $[x] \subseteq \text{BND}(X)$;
- (N) If $\mathfrak{R}(a_N \mid [x]) \leq \mathfrak{R}(a_P \mid [x])$ and $\mathfrak{R}(a_N \mid [x]) \leq \mathfrak{R}(a_B \mid [x])$, decide $[x] \subseteq \text{NEG}(X)$.

Consider a special kind of loss functions with

$$\begin{aligned} \lambda_{PP} &\leq \lambda_{BP} < \lambda_{NP}, \\ \lambda_{NN} &\leq \lambda_{BN} < \lambda_{PN}. \end{aligned} \quad (3)$$

TABLE 1: The loss function matrix.

	a_P	a_B	a_N
X	λ_{PP}	λ_{BP}	λ_{NP}
X^c	λ_{PN}	λ_{BN}	λ_{NN}

That is, the cost of classifying an object x belonging to X into the positive region $\text{POS}(X)$ is less than or equal to the cost of classifying x into the boundary region $\text{BND}(X)$, and both of these costs are strictly less than the cost of classifying x into the negative region $\text{NEG}(X)$. The reverse order of costs is used for classifying an object that does not belong to X . The decision rules can be reexpressed as follows:

- (P) If $P(X | [x]) \geq \alpha$ and $P(X | [x]) \geq \gamma$, decide $x \in \text{POS}(X)$;
- (B) If $P(X | [x]) \leq \alpha$ and $P(X | [x]) \geq \beta$, decide $x \in \text{BND}(X)$;
- (N) If $P(X | [x]) \leq \beta$ and $P(X | [x]) \leq \gamma$, decide $x \in \text{NEG}(X)$,

where the parameters α , β , and γ are defined as

$$\begin{aligned}\alpha &= \frac{\lambda_{PN} - \lambda_{BN}}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}, \\ \beta &= \frac{\lambda_{BN} - \lambda_{NN}}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}, \\ \gamma &= \frac{\lambda_{PN} - \lambda_{NN}}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}.\end{aligned}\quad (4)$$

When

$$(\lambda_{PN} - \lambda_{BN})(\lambda_{NP} - \lambda_{BP}) > (\lambda_{BP} - \lambda_{PP})(\lambda_{BN} - \lambda_{NN}), \quad (5)$$

we have $0 \leq \beta < \gamma < \alpha \leq 1$. After tie-breaking, the simplified rules are obtained as follows:

- (P1) If $P(X | [x]) \geq \alpha$, decide $x \in \text{POS}(X)$;
- (B1) If $\beta < P(X | [x]) < \alpha$, decide $x \in \text{BND}(X)$;
- (N1) If $P(X | [x]) \leq \beta$, decide $x \in \text{NEG}(X)$.

Let $\pi_D = \{D_1, D_2, \dots, D_m\}$ denote the partition of the universe U induced by D . Based on the thresholds (α, β) , one can divide the universe U into three regions of the decision partition π_D :

$$\begin{aligned}\text{POS}_A^{(\alpha, \beta)}(D) &= \{x \in U \mid P(D_{\max}([x]_A) \mid [x]_A) \geq \alpha\}, \\ \text{BND}_A^{(\alpha, \beta)}(D) &= \{x \in U \mid \beta < P(D_{\max}([x]_A) \mid [x]_A) < \alpha\}, \\ \text{NEG}_A^{(\alpha, \beta)}(D) &= \{x \in U \mid P(D_{\max}([x]_A) \mid [x]_A) \leq \beta\},\end{aligned}\quad (6)$$

where $D_{\max}([x]_A) = \arg \max_{D_i \in \pi_D} \{|[x]_A \cap D_i| / |[x]_A|\}$.

Let $p = p(D_{\max}([x]_A) \mid [x]_A)$; the Bayesian expected cost of positive rule, boundary rule, and negative rule can be expressed, respectively, as follows:

$$\begin{aligned}p \cdot \lambda_{PP} + (1 - p) \cdot \lambda_{PN}; \\ p \cdot \lambda_{BP} + (1 - p) \cdot \lambda_{BN}; \\ p \cdot \lambda_{NP} + (1 - p) \cdot \lambda_{NN}.\end{aligned}\quad (7)$$

3. Minimal-Total-Cost Attribute Reduction in Decision-Theoretic Rough Set Models

In this section, we focus on cost-sensitive attribute reduction based on test costs and decision costs in DTRS models. The objective of attribute reduction is to minimize the total cost through considering a tradeoff between test costs and decision costs. Minimizing the total cost is equal to minimizing the average total cost (ATC), so we study the minimal average-total-cost reduct problem.

Test cost is intrinsic to data. There are a number of test-cost-sensitive decision systems. A corresponding hierarchy consisting of six models was proposed in [21]. Here, we consider only the test-cost-independent decision system, which is the simplest though most widely used model.

Definition 2 (see [21]). A test-cost-independent decision system (TCI-DS) S is the 6-tuple:

$$S = (U, C, D, V, I, tc), \quad (8)$$

where U, C, D, V, I have the same meanings as in a DS and $tc : C \rightarrow \mathbb{R}^+$ is the test cost function. Test costs are independent of one another; that is, $tc(A) = \sum_{a \in A} tc(a)$ for any $A \subseteq C$.

By introducing test cost into DTRS models, we can obtain the following definition.

Definition 3. A test-cost-independent cost-sensitive decision system in DTRS models (DTRS-TCI-CDS) S is the 7-tuple:

$$S = (U, C, D, V, I, tc, (\lambda_{ij})_{2 \times 3}), \quad (9)$$

where U, C, D, V, I, tc have the same meanings as in Definition 2 and $(\lambda_{ij})_{2 \times 3}$ is the loss function matrix listed in Table 1, where $i \in \{P, B, N\}$ and $j \in \{P, N\}$.

An example of DTRS-TCI-CDS is given in Tables 2, 3, and 4. From Table 2 to Table 4, there are a decision system where $U = \{x_1, x_2, \dots, x_9\}$ and $C = \{a_1, a_2, \dots, a_6\}$, a corresponding test cost vector, and a corresponding loss function matrix, respectively.

TABLE 2: An example decision system.

	a_1	a_2	a_3	a_4	a_5	a_6	D
x_1	1	1	1	1	1	1	d_1
x_2	1	0	1	0	1	1	d_1
x_3	0	1	1	1	0	0	d_2
x_4	1	1	1	0	0	1	d_2
x_5	0	0	1	1	0	1	d_2
x_6	1	0	1	0	1	1	d_3
x_7	0	0	0	1	1	0	d_3
x_8	1	0	1	0	1	1	d_3
x_9	0	0	1	1	0	1	d_3

TABLE 3: An example test cost vector.

	a_1	a_2	a_3	a_4	a_5	a_6
$tc(a_i)$	\$92	\$9	\$96	\$81	\$87	\$54

TABLE 4: An example loss function matrix.

	a_P	a_B	a_N
X	480	2895	6095
X^c	7846	4238	373

For a given DTRS-TCI-CDS, $A \subseteq C$; the decision cost is composed of the three types of cost formulated in (7), so the decision cost can be expressed as

$$\begin{aligned}
dc(U, A) = & \sum_{p_i \geq \alpha} (p_i \cdot \lambda_{PP} + (1 - p_i) \cdot \lambda_{PN}) \\
& + \sum_{\beta < p_j < \alpha} (p_j \cdot \lambda_{BP} + (1 - p_j) \cdot \lambda_{BN}) \quad (10) \\
& + \sum_{p_k \leq \beta} (p_k \cdot \lambda_{NP} + (1 - p_k) \cdot \lambda_{NN}),
\end{aligned}$$

where $p_i = p(D_{\max}([x_i]_A) \mid [x_i]_A)$. According to (6), we can rewrite the decision cost formulation as

$$\begin{aligned}
dc(U, A) = & \sum_{x_i \in \text{POS}_A^{(\alpha, \beta)}(D)} (p_i \cdot \lambda_{PP} + (1 - p_i) \cdot \lambda_{PN}) \\
& + \sum_{x_j \in \text{BND}_A^{(\alpha, \beta)}(D)} (p_j \cdot \lambda_{BP} + (1 - p_j) \cdot \lambda_{BN}) \\
& + \sum_{x_k \in \text{NEG}_A^{(\alpha, \beta)}(D)} (p_k \cdot \lambda_{NP} + (1 - p_k) \cdot \lambda_{NN}). \quad (11)
\end{aligned}$$

Obviously, we can obtain the average decision cost as follows:

$$\overline{dc(U, A)} = \frac{dc(U, A)}{|U|}. \quad (12)$$

Because the test cost of any object is the same for the test set A , the average total cost (ATC) is given by

$$\text{ATC}(U, A) = tc(A) + \overline{dc}(U, A). \quad (13)$$

Similar to [20], we study the decreasing cost attribute reduction to avoid the interpretation difficulties in region preservation based definitions. The definition of decreasing average-total-cost attribute reduct is presented as follows.

Definition 4. In a DTRS-TCI-CDS, $S = (U, C, D, V, I, tc, (\lambda_{ij})_{2 \times 3})$; $R \subseteq C$ is a decreasing average-total-cost attribute reduct if and only if

- (1) $\text{ATC}_R \leq \text{ATC}_C$,
- (2) $\forall R' \subset R, \text{ATC}_{R'} > \text{ATC}_R$.

According to the definition, we choose the subsets of C which ensure that ATC will be decreased or unchanged for decisions making in the processing of attribute reduction.

In most situations, users want to obtain the smallest total cost in the classification procedure, so we propose an optimization problem with the objective of minimizing average total classification cost. The proper attribute set to make ATC minimal is called minimal average-total-cost reduct (MACR). The optimization problem is, namely, the MACR problem. We define them as follows.

Definition 5. In a DTRS-TCI-CDS, $S = (U, C, D, V, I, tc, (\lambda_{ij})_{2 \times 3})$; $R \subseteq C$ is a MACR if and only if

$$R = \underset{A \subseteq C}{\text{argmin}} \{ \text{ATC}_A \}. \quad (14)$$

Definition 6. The MACR problem:

- input: $S = (U, C, D, V, I, tc, (\lambda_{ij})_{2 \times 3})$;
output: $A \subseteq C$;
optimization objective: $\min |\text{ATC}(U, A)|$.

If we set $tc(a) = c$ for all $a \in C$ where c is a constant, the MACR problem is essentially the minimal-decision-cost attribute reduct problem [20], so the former is a generalization of the latter.

4. Algorithms

Since the MACR problem is a combinational problem and it is not easy to get the optimal solution in a linear time, we use heuristic approach to obtain the approximate optimal solution. However, to evaluate the performance of a heuristic algorithm in terms of the quality of the solution, we should find an optimal reduct first, so an exhaustive algorithm is also needed. In this section, we propose a backtracking algorithm and a δ -weighted heuristic algorithm to address the MACR problem.

4.1. The Backtracking Attribute Reduction Algorithm. The backtracking algorithm is illustrated in Algorithm 1. In order to invoke this backtracking algorithm, several global variables should be explicitly initialized as follows:

- (1) $R = \emptyset$ is a reduct with minimal average total cost;
- (2) $\text{cmc} = \overline{dc}(U, R)$ is currently minimal average total cost;
- (3) $l = 0$ is current level test index lower bound.

Input: $(U, C, D, \{V_a\}, \{I_a\}, tc, (\lambda_{ij})_{2 \times 3})$, select tests R ,
current level test index lower bound l

Output: R and cmc , they are global variables

Method: backtracking

- (1) **for** $(i = l; i < |C|; i++)$ **do**
- (2) $A = R \cup \{a_i\}$
- (3) **if** $(tc(A) \geq cmc)$ **then**
- (4) $\text{continue; //Pruning for too expensive test costs}$
- (5) **end if**
- (6) **if** $(ATC(U, A) < cmc)$ **then**
- (7) $cmc = ATC(U, A); //Update the minimal average total cost$
- (8) $R = A; //Update the minimal total cost reduct$
- (9) **end if**
- (10) $\text{backtracking}(A, i + 1);$
- (11) **end for**

ALGORITHM 1: A backtracking algorithm to the MACR problem.

The backtracking algorithm is denoted as backtracking (R, l) . A reduct with minimal ATC will be stored in R at the end of the algorithm execution. Generally, the search space of the attribute reduction algorithm is $2^{|C|}$. To reduce the search space, we employ one pruning technique shown in lines 3 through 5 in Algorithm 1. The attribute subset A will be discarded if the test cost of A is not less than the current minimal average total cost (cmc), in that the decision costs are nonnegative in real applications.

Note that total costs may decrease with the addition of attributes, which means that ATC under an attribute set may be less than that under some of its subsets. That is different from the previous works which considered only test cost [25], in which test costs increase when more attributes are selected. The following example gives an intuitive understanding.

Example 7. Take the DTRS-TCI-CDS listed in Tables 2–4 for example. By computation, we find that ATC is 3974.8 when the selected attribute set $A = \{c_4\}$, while ATC is reduced to 3346.2 when $A = \{c_4, c_6\}$.

Therefore, no matter whether the currently selected attribute subset A satisfies $ATC(U, A) < cmc$ or not, A continues expanding to search a minimal ATC, which is shown in line 10 of Algorithm 1.

4.2. The δ -Weighted Heuristic Attribute Reduction Algorithm.

The δ -weighted heuristic attribute reduction algorithm is listed in Algorithm 2, in which the algorithm framework contains two main steps. Let A denote the set of currently selected attributes. First, we combine the current best attribute subset $S \subseteq (C - A)$ with A according to the heuristic attribute significance function $f(A, S, tc)$ until A becomes a superreduct. This step is essentially the attribute addition. Then, we delete the attribute a from A to guarantee A with the current minimal total cost.

Lines 4 through 13 contain the key code of the addition step. There are two main differences from those in existing works [22, 25, 29]. One is the heuristic attribute significance

function. We propose the δ -weighted attribute significance function as follows:

$$f(A, S, tc) = \left(\text{POS}_{A \cup S}^{(\alpha, \beta)}(D) - \text{POS}_A^{(\alpha, \beta)}(D) \right) \times \left(1 + \frac{1}{\sqrt{|S|} \cdot tc(a_{k_1}) \cdot tc(a_{k_2}) \cdots tc(a_{k_{|S|}})} \right), \quad (15)$$

where a_{k_i} is the attribute in S and $tc(a_{k_i})$ is the test cost of a_{k_i} .

The other difference is the computation steps. At first, the dimension of S is 1, which means that we test current left attributes one by one. However, since the positive region may shrink with the addition of attributes in DTRS models [19], for all $a_i \in CA$, $(\text{POS}_{A \cup \{a_i\}}^{(\alpha, \beta)}(D) - \text{POS}_A^{(\alpha, \beta)}(D))$ may not be more than 0 at the same time. In this case, we cannot choose a suitable attribute to make current $\text{POS}_A^{(\alpha, \beta)}(D)$ expand to reach $\text{POS}_{A \cup S}^{(\alpha, \beta)}(D) \supseteq \text{POS}_C^{(\alpha, \beta)}(D)$. To address this situation, we gradually increase the dimension of S , namely, consider multiattributes simultaneously, and compute the corresponding values of attribute significance function until at least one value is more than 0.

5. Experiments

In this section, the performance of our two algorithms is studied. We try to answer the following questions by experimentation.

- (1) Are both the backtracking algorithm and the heuristic algorithm efficient?
- (2) Is the heuristic algorithm effective for the MACR problem?
- (3) Are both algorithms appropriate for classification?

5.1. Data Generation. Experiments are undertaken on four datasets obtained from the UCI library. The basic information

```

Input:  $(U, C, D, \{V_a\}, \{I_a\}, tc, (\lambda_{ij})_{2 \times 3})$ 
Output: The reduct  $A$ 
Method:
(1)  $A = \emptyset$ ;
    //Addition
(2)  $CA = C$ ;
(3) while  $(POS_A^{(\alpha, \beta)}(D) \not\subseteq POS_C^{(\alpha, \beta)}(D))$  do
(4)    $n = 1$ ; //  $n$  controls the dimension of  $S$ 
(5)   for (each  $S \subseteq CA$ ) do
(6)     if  $(|S| = n)$  then
(7)       Compute  $f(A, S, tc)$ ;
(8)     end if
(9)   end for
(10)  Select  $S'$  with the maximal  $f(A, S', tc)$ ;
(11)  if  $(f(A, S', tc) \leq 0)$  then
(12)     $n++$ ;
(13)    Go to line 5;
(14)  else
(15)     $A = A \cup S'$ ;  $CA = CA - S'$ ;
(16)  end if
(17) end while
    //Deletion
(18) while  $(ATC(U, A) > ATC(U, A - \{a\}))$  do
(19)   for (each  $a \in A$ ) do
(20)     Compute  $ATC(U, A - \{a\})$ ;
(21)   end for
(22)   Select  $a'$  with the minimal  $ATC(U, A - \{a'\})$ ;
(23)    $A = A - \{a'\}$ ;
(24) end while
(25) return  $A$ ;

```

ALGORITHM 2: An addition-deletion cost-sensitive attribute reduction algorithm.

TABLE 5: Dataset information.

Name	Domain	$ C $	$ U $	$D = \{d\}$
Tic-tac-toe	Game	9	958	Class
Voting	Society	16	435	Vote
Zoo	Zoology	16	101	Type
Mushroom	Botany	22	8124	Class

of the datasets is listed in Table 5, where $|C|$ is the number of condition attributes, $|U|$ is the number of instances, and D is the name of the decision. Note that missing values in these datasets (e.g., in voting dataset) are treated as a particular value. That is, ? is equal to itself and unequal to any other value.

Since there are no intrinsic test costs and loss functions in the datasets mentioned above, we will create these data for experimentations. First, we generate test costs that are always represented by positive integers. Let a_i be a condition attribute; $tc(a_i)$ is set to a random number in $[1, 100]$ subject to the uniform distribution discussed in [22]. Then, we produce loss functions λ_{ij} ($i \in \{P, B, N\}, j \in \{P, N\}$), which are random nonnegative integers satisfying (3) and (5). Since the loss functions are often more than test costs in real life, we set the average of λ_{ij} to be in $[100, 5000]$. Of course, the assumptions of cost value could be easily changed if

necessary. To observe whether the algorithm efficiency is influenced by the ratio of loss functions to test costs, experiments shown below are undertaken with two groups of cost settings for each dataset listed in Table 5. Each group contains 100 different cost settings. Test costs in both groups are the same, but the loss functions are different. The average values of loss functions (ALF) in group 1 and group 2 are around 500 and 3000, respectively. Experiments are undertaken on a PC with Intel 2.20 GHz CPU and 4 GB memory.

5.2. Efficiencies of the Two Algorithms. We study the efficiencies of both algorithms using two metrics. One is the number of backtrack steps Algorithm 1 is invoked. Comparing it with the size of search space [25], the efficiency of the backtracking algorithm is investigated. The other is the run-time comparison between the two algorithms. The metric is used to study the efficiency of the heuristic algorithm. The search space size and the average number of backtracking steps for Algorithm 1 are depicted in Table 6, and the average run-time for both algorithms is shown in Table 7, where the unit of run-time is 1 ms.

From the results, we note the following.

- (1) In both groups, the number of backtrack steps is less than the search space size, which manifests the effectiveness of the pruning technique in Algorithm 1.

TABLE 6: The average number of backtrack steps for Algorithm 1.

Dataset	Search space size	Average backtrack steps	
		ALF \approx 500	ALF \approx 3000
Tic-tac-toe	2^9	130.68	406.34
Voting	2^{16}	774.18	4873.3
Zoo	2^{16}	547.05	5134.96
Mushroom	2^{22}	3758.50	307682.57

- (2) With the increasing of ALF, both the backtrack steps and the run-time of Algorithm 1 grow, which means that the efficiency of the backtracking algorithm is influenced by the ratio of loss functions to test costs. The reason is that, when the loss functions are much more than test costs, currently minimal ATC, namely, cmc in Algorithm 1, is also high compared to current test costs. In this case, the pruning technique shown in lines 3 to 4 of Algorithm 1 cannot make effect.
- (3) The run-time of Algorithm 2 is small compared with Algorithm 1, especially for the dataset Zoo. Therefore, the heuristic algorithm is very efficient. Moreover, the heuristic algorithm is stable in terms of run-time with the increasing of ALF.

In a word, the heuristic algorithm is good at the efficiency. Although the backtracking algorithm is not very efficient sometimes, it is still needed to evaluate the performance of a heuristic algorithm in terms of the quality of the solution.

5.3. *Effectiveness of the Two Algorithms.* In this part, we observe the effectiveness of both algorithms by using four metrics. First, two metrics defined in [22], namely, finding optimal factor (FOF) and average exceeding factor (AEF), are computed to measure the performance of the heuristic algorithm from the perspective of cost. In the computations, the results of the backtracking algorithm are used to evaluate the effectiveness of the heuristic algorithm. The results of the two metrics are shown in Figure 1.

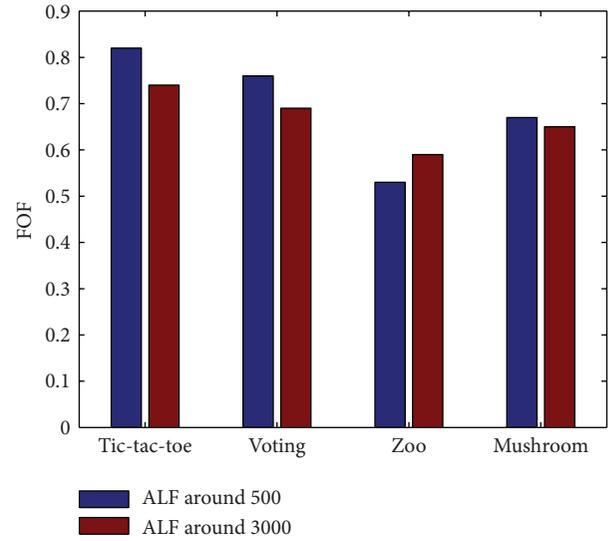
From the results, we note the following.

- (1) The values of FOF and AEF are not significantly different between $ALF \approx 500$ and $ALF \approx 3000$. Maybe we can conclude that the performance of the heuristic algorithm is little influenced by the ratio of loss functions to test costs.
- (2) All FOF are above 0.5, and all AEF are below 0.1. In other words, the results are acceptable.

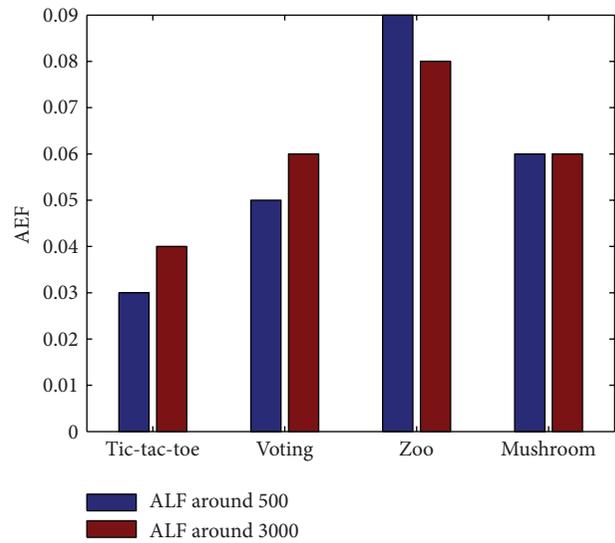
Then, we compare the classification performances of the original data and the reduced data obtained by our two algorithms based on 10-fold cross validation. CART and RBF-kernel SVM are used as learning algorithms, respectively. The results are depicted in Tables 8-9. We also present the comparison of the average numbers of selected attributes, which is shown in Table 10.

From the results, we observe the following.

- (1) The values of classification accuracy by our algorithms are a little lower than those by the raw data,



(a)



(b)

FIGURE 1: Assuming that ALF is around 500 and 3000, respectively, the effectiveness of the heuristic algorithm is measured by using two metrics. (a) Finding optimal factor (FOF) is the fraction of successful searches of an optimal reduct in experiments. The higher the FOF is, the better the heuristic algorithm is. It is shown that all FOF are above 0.5. (b) Average exceeding factor (AEF) is the average value of the fractions beyond the minimal-average-total costs. The lower the AEF, the better the algorithm. All AEF are below 0.1 in the figure.

but the numbers of selected attributes are effectively reduced, which is consistent with the essence of DTRS models. Different from the classical rough set, classification error is acceptable within a certain range according to the thresholds in DTRS models. Consequently, the reduction effectiveness is improved.

- (2) With the increasing of ALF, all numbers of selected attributes grow, and the classification performance of

TABLE 7: Average run-time comparison.

Dataset	ALF \approx 500		ALF \approx 3000	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
Tic-tac-toe	591.7	206.7	1376.14	177.16
Voting	975.93	203.77	3845.7	162
Zoo	252.77	1.73	2327.02	1.42
Mushroom	9541.24	807.91	64172.5	835.72

TABLE 8: Classification performance comparison with CART classifier.

Dataset	Raw data	ALF \approx 500		ALF \approx 3000	
		Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
Tic-tac-toe	93.72%	87.48%	87.18%	91.6%	89.4%
Voting	95.47%	90.34%	86.37%	94.07%	87.38%
Zoo	90.69%	82.85%	78.92%	89.65%	85.52%
Mushroom	99.96%	95.18%	90.51%	95.82%	90.29%

TABLE 9: Classification performance comparison with RBF-kernel SVM classifier.

Dataset	Raw data	ALF \approx 500		ALF \approx 3000	
		Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
Tic-tac-toe	87.59%	82.16%	81.8%	85.97%	84.28%
Voting	95.38%	87.35%	83.4%	92.13%	84.36%
Zoo	90.1%	83.73%	79.27%	89.52%	85.8%
Mushroom	100%	97.21%	90.38%	95.79%	91.88%

TABLE 10: The comparison of the average numbers of selected attributes.

Dataset	Raw data	ALF \approx 500		ALF \approx 3000	
		Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
Tic-tac-toe	9	1.31	1.34	5.59	4.11
Voting	16	1.29	1.06	2.6	1.6
Zoo	16	1.98	1.41	3.51	2.59
Mushroom	22	1.57	1.26	6.95	5.32

most datasets improves. This means that the tolerability of classification error is decreasing when the classification costs increase.

- (3) For all datasets, the classification performance of Algorithm 1 is a little better than that of Algorithm 2; meanwhile, the numbers of selected attributes are more in most cases.

6. Conclusions

In this paper, we address cost-sensitive attribute reduction problem in DTRS models. By considering the tradeoff of decision costs and test costs, minimal average-total-cost attribute reduct is defined, and the corresponding optimization problem is proposed. Both backtracking and heuristic algorithms are designed to deal with the optimization problem. Experimental results demonstrate the efficiency and the effectiveness of both algorithms. By combining test costs with the existing elements in DTRS models, such as the

loss functions and the probabilistic approaches, our model is practical in real applications.

The following research topics deserve further investigation.

- (1) The MACR problem could be addressed again based on more complicated test-cost-sensitive decision systems (DS), such as the simple common-test-cost DS and the complex common-test-cost DS [21]. The corresponding algorithms may also be more complicated.
- (2) Sometimes the costs one could afford are limited. We could consider the attribute reduction problem with test cost constraint or total cost constraint in DTRS models.
- (3) Recently, from the viewpoint of rough set theory, Yao [30, 31] has discussed three-way decisions, which may have many real-world applications. One could explore the cost-sensitive attribute reduction problem for three-way decisions with decision-theoretic rough sets.

In summary, this study suggests new research trends concerning decision-theoretic rough set theory, attribute reduction problem, and cost-sensitive learning applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is in part supported by the National Science Foundation of China under Grant nos. 61379089, 61379049, and 61170128 and the Education Department of Fujian Province under Grant no. JA12224.

References

- [1] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, USA, 1944.
- [2] D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, Cambridge, Mass, USA, 1991.
- [3] M. Perc and A. Szolnoki, "Coevolutionary games—a mini review," *BioSystems*, vol. 99, no. 2, pp. 109–125, 2010.
- [4] M. Perc, J. G. Gómez-Gardeñes, A. Szolnoki, L. M. Floría, and Y. Moreno, "Evolutionary dynamics of group interactions on structured populations: a review," *Journal of the Royal Society Interface*, vol. 10, no. 80, Article ID 20120997, 17 pages, 2013.
- [5] M. Perc and P. Grigolini, "Collective behavior and evolutionary games—an introduction," *Chaos, Solitons & Fractals*, vol. 56, pp. 1–5, 2013.
- [6] A. Szolnoki, M. Perc, and G. Szabó, "Defense mechanisms of empathetic players in the spatial ultimatum game," *Physical Review Letters*, vol. 109, Article ID 078701, 4 pages, 2012.
- [7] A. Szolnoki, M. Perc, and G. Szabó, "Accuracy in strategy imitations promotes the evolution of fairness in the spatial ultimatum game," *Europhysics Letters*, vol. 100, no. 2, Article ID 28005, 6 pages, 2012.
- [8] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [9] Z. Pawlak and A. Skowron, "Rough sets: some extensions," *Information Sciences*, vol. 177, no. 1, pp. 28–40, 2007.
- [10] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, no. 1, pp. 3–27, 2007.
- [11] W. Ziarko, "Variable precision rough set model," *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39–59, 1993.
- [12] Y. Y. Yao, S. K. M. Wong, and P. Lingras, "A decision-theoretic rough set model," in *Methodologies for Intelligent Systems, 5 (Knoxville, TN, 1990)*, pp. 17–24, North-Holland, New York, NY, USA, 1990.
- [13] Y. Y. Yao and S. K. M. Wong, "A decision theoretic framework for approximating concepts," *International Journal of Man-Machine Studies*, vol. 37, no. 6, pp. 793–809, 1992.
- [14] S. Greco, B. Matarazzo, and R. Slowinski, "Rough set approach to decisions under risk," in *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC '00)*, vol. 2005 of *Lecture Notes in Computer Science*, pp. 160–169, 2001.
- [15] S. Greco, B. Matarazzo, and R. Slowinski, "Rough sets theory for multicriteria decision analysis," *European Journal of Operational Research*, vol. 129, no. 1, pp. 1–47, 2001.
- [16] A. Skowron, S. Ramanna, and J. F. Peters, "Conflict analysis and information systems: a rough set approach," *Proceedings of the 1st International Conference on Rough Sets and Knowledge Technology (RSKT '06)*, vol. 4062, pp. 233–240, 2006.
- [17] Y. Yao and Y. Zhao, "Attribute reduction in decision-theoretic rough set models," *Information Sciences*, vol. 178, no. 17, pp. 3356–3373, 2008.
- [18] Y. Zhao, S. K. M. Wong, and Y. Yao, "A note on attribute reduction in the decision-theoretic rough set model," in *Transactions on Rough Sets XIII*, vol. 6499 of *Lecture Notes in Computer Science*, pp. 260–275, 2011.
- [19] H. Li, X. Zhou, J. Zhao, and D. Liu, "Attribute reduction in decision-theoretic rough set model: a further investigation," in *Proceedings of 6th International Conference on Rough Sets and Knowledge Technology (RSKT '11)*, vol. 6954 of *Lecture Notes in Computer Science*, pp. 466–475, 2011.
- [20] X. Jia, W. Liao, Z. Tang, and L. Shang, "Minimum cost attribute reduction in decision-theoretic rough set models," *Information Sciences*, vol. 219, pp. 151–167, 2013.
- [21] F. Min and Q. Liu, "A hierarchical model for test-cost-sensitive decision systems," *Information Sciences*, vol. 179, no. 14, pp. 2442–2452, 2009.
- [22] F. Min, H. He, Y. Qian, and W. Zhu, "Test-cost-sensitive attribute reduction," *Information Sciences*, vol. 181, no. 22, pp. 4928–4942, 2011.
- [23] F. Min, Q. Hu, and W. Zhu, "Feature selection with test cost constraint," *International Journal of Approximate Reasoning*, vol. 55, no. 1, part 2, pp. 167–179, 2014.
- [24] S. Liao, J. Liu, F. Min, and W. Zhu, "Minimal-test-cost reduct problem on neighborhood decision systems," *Journal of Information and Computational Science*, vol. 9, no. 14, pp. 4083–4098, 2012.
- [25] F. Min and W. Zhu, "Attribute reduction of data with error ranges and test costs," *Information Sciences*, vol. 211, pp. 48–67, 2012.
- [26] H. Zhao, F. Min, and W. Zhu, "Test-cost-sensitive attribute reduction of data with normal distribution measurement errors," *Mathematical Problems in Engineering*, vol. 2013, Article ID 946070, 12 pages, 2013.
- [27] F. Min, W. Zhu, H. Zhao, and G. Pan, "Coser: cost-sensitive rough sets," 2011, <http://grc.fjz.edu.cn/~fmin/coser/>.
- [28] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998, <http://www.ics.uci.edu/~mllearn/mlrepository.html>.
- [29] H. Zhao, F. Min, and W. Zhu, "Cost-sensitive feature selection of numeric data with measurement errors," *Journal of Applied Mathematics*, vol. 2013, Article ID 754698, 13 pages, 2013.
- [30] Y. Yao, "Three-way decisions with probabilistic rough sets," *Information Sciences*, vol. 180, no. 3, pp. 341–353, 2010.
- [31] Y. Yao, "The superiority of three-way decisions in probabilistic rough set models," *Information Sciences*, vol. 181, no. 6, pp. 1080–1096, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

