Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 754698, 13 pages
http://dx.doi.org/10.1155/2013/754698

Research Article

Cost-Sensitive Feature Selection of Numeric Data with Measurement Errors

Hong Zhao, Fan Min, and William Zhu

Laboratory of Granular Computing, Zhangzhou Normal University, Zhangzhou 363000, China

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

Received 24 December 2012; Accepted 22 March 2013

Academic Editor: Jung-Fa Tsai

Copyright © 2013 Hong Zhao 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.

Feature selection is an essential process in data mining applications since it reduces a model's complexity. However, feature selection with various types of costs is still a new research topic. In this paper, we study the cost-sensitive feature selection problem of numeric data with measurement errors. The major contributions of this paper are fourfold. First, a new data model is built to address test costs and misclassification costs as well as error boundaries. It is distinguished from the existing models mainly on the error boundaries. Second, a covering-based rough set model with normal distribution measurement errors is constructed. With this model, coverings are constructed from data rather than assigned by users. Third, a new cost-sensitive feature selection problem is defined on this model. It is more realistic than the existing feature selection problems. Fourth, both backtracking and heuristic algorithms are proposed to deal with the new problem. Experimental results show the efficiency of the pruning techniques for the backtracking algorithm and the effectiveness of the heuristic algorithm. This study is a step toward realistic applications of the cost-sensitive learning.

1. Introduction

Feature selection [1–3] is an essential process in data mining applications. The main aim of feature selection is to reduce the dimensionality of the feature space and to improve the predictive accuracy of a classification algorithm [4, 5]. In many domains, the misclassification costs [6–9] and the test costs [10, 11] must be considered in the feature selection process. Cost-sensitive feature selection [12–14] focuses on selecting a feature subset with a minimal total cost as well as preserving a particular property of the decision system [15, 16].

Test costs and misclassification costs are two most important types of cost in cost-sensitive learning [17]. The test cost is money, time, or other resources we pay for collecting a data item of an object [18, 19]. The misclassification cost is the penalty we receive while deciding that an object belongs to class J when its real class is K [6, 8]. Some works have considered only misclassification costs [20], or only test costs [21–23]. However, in many applications, it is important to consider both types of costs together.

Recently, the cost-sensitive feature selection problem for nominal datasets was proposed [17]. A backtracking algorithm has been presented to address this problem. However, this algorithm has been applied to only small datasets and addressed on only nominal data. In real applications, the data can be acquired from measurements with different errors. The measurement errors of the data have certain universality.

In this paper, we propose the cost-sensitive feature selection problem of numerical data with measurement errors and deal with it through considering the trade-off between test costs and misclassification costs. The major contributions of this paper are fourfold. First, based on normal distribution measurement errors, we build a new data model to address test costs and misclassification costs as well as error boundaries. It is distinguished from the existing models [17] mainly on the error boundaries. Second, we construct a computational model of the covering-based rough set with normal distribution measurement errors. In fact, normal distribution [24, 25] is found to be applicable over almost the whole of science and engineering measurement. With this model, coverings are constructed from data rather than assigned by users. Third, the cost-sensitive feature selection problem is defined on this new model of covering-based rough set. It is more realistic than the existing feature selection problems. Fourth, a backtracking algorithm is proposed to find an optimal feature subset for small datasets. However, for large dataset, finding a minimal cost feature subset is NP-hard. Consequently, we propose a heuristic algorithm to deal with this problem.

Six open datasets from the University of California-Irvine (UCI) library are employed to study the performance and effectiveness of our algorithms. Experiments are undertaken with open source software cost-sensitive rough sets (Coser) [26]. Experimental results show that the pruning techniques of the backtracking algorithm reduce searching operations by several orders of magnitudes. In addition, the heuristic algorithm can provide efficient solution to find an optimal feature subset in most cases. Even if the feature subset is not optimal, it is still acceptable from a statistical point of view.

The rest of the paper is organized as follows. Section 2 presents data models with test costs and misclassification costs as well as measurement errors. Section 3 describes the computational model, namely, covering-based rough set model with measurement errors. The feature selection with the minimal cost problem on the new model is also defined in this section. Then, Section 4 presents a backtracking algorithm and a heuristic algorithm to address this feature selection problem. In Section 5, we discuss the experimental settings and results. Finally, Section 6 concludes and suggests further research trends.

2. Data Models

Data models are presented in this section. First, we start from basic decision systems. Then, we introduce normal distribution errors to test and propose a decision system with measurement errors. Finally, we introduce a decision system based on measurement errors with test costs and misclassification costs.

2.1. Decision Systems. Decision systems are fundamental in data mining and machine learning. For completeness, a decision system is defined below.

Definition 1 (see [27]). A decision system (DS) is the 5-tuple:

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

where U is a universal set of objects, C is a nonempty set of conditional attributes, and d is the decision attribute. For each $a \in C \cup \{d\}$, $I_a : U \rightarrow V_a$. The set V_a is the value set of attribute a, and I_a is the information function for each attribute a.

In order to facilitate processing and comparison, the values of conditional attributes are normalized from their value into a range from 0 to 1. In fact, there are a number of normalization approaches. For simplicity, we employ the linear function $y = (x - \min)/(\max - \min)$, where x is the initial value, y is the normalized value, and max and min are the maximal and minimal values of the attribute domain, respectively.

Table 1: An example of numeric decision system (*Liver*).

Patient	Mcv	Alkphos	Sgpt	Sgot	Gammagt	Drinks	Selector
x_1	0.31	0.23	0.08	0.28	0.09	0.00	y
x_2	0.14	0.38	0.23	0.35	0.06	0.10	y
x_3	0.25	0.40	0.40	0.14	0.17	0.20	y
x_4	0.60	0.46	0.51	0.25	0.11	0.60	n
x_5	0.41	0.64	0.62	0.30	0.02	0.30	n
x_6	0.35	0.50	0.75	0.30	0.02	0.40	n
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{344}	0.68	0.39	0.15	0.23	0.03	0.80	n
x_{345}	0.87	0.66	0.35	0.52	0.21	1.00	n

Table 1 is a decision system of *Bupa liver disorder* (*Liver* for short), in which conditional attributes are normalized values. Here, C = {Mcv, Alkphos, Sgpt, Sgot, Gammagt, Drinks}, d = {Selector}, and $U = \{x_1, x_2, \ldots, x_{345}\}$.

Liver contains 7 attributes. The first 5 attributes are all blood tests which are thought to be sensitive to *liver* disorders that might arise from excessive alcohol consumption. The sixth attribute is the number of alcoholic drinks per day. Each line in *Liver* constitutes the record of a single male individual. The *Selector* attribute is used to split data into two sets.

2.2. A Decision System with Measurement Errors. In real applications, datasets often contain many continuous (or numerical) attributes. There are a number of measurement methods with different test costs to obtain a numerical data item. Generally, higher test cost is required to obtain data with smaller measurement error [28]. The measurement errors often satisfy normal distribution which is found to be applicable over almost the whole of science and engineering measurement. We include normal distribution measurement errors in our model to expand the application scope.

Definition 2 (see [28]). A decision system with measurement errors (MEDS) S is the 6-tuple:

$$S = (U, C, d, V, I, n), \tag{2}$$

where U, C, d, V, and I have the same meanings as in Definition 1, $n : C \rightarrow \mathbb{R}^+ \cup \{0\}$ is the maximal measurement error function, and $\pm n(a)$ is the error boundary of attribute a.

Given $x_i \in U$, the error boundary of attribute a is given by

$$n(a) = \frac{\Delta \Sigma_{i=1}^{m} a(x_i)}{m}, \tag{3}$$

where the regulator factor $\Delta \in [0, 1]$ can adjust the error boundary.

In applications, one can deal with the abnormal value of measurement error according to the Pauta criterion of measurement error theory, which is used to determine

Table 2: An example of neighborhood boundary vector.

a	Mcv	Alkphos	Sgpt	Sgot	Gammagt	Drinks
$n(a)$	0.069	0.087	0.086	0.036	0.026	0.017

Table 3: An example of test cost vector.

a	Mcv	Alkphos	Sgpt	Sgot	Gammagt	Drinks
$\mathrm{tc}(a)$	\$26	\$17	\$34	\$45	\$38	\$5

the abnormal values. That is, if the repeated measurement data satisfy $|x_i - \overline{x}| > 3\sigma$, $(i = 1, 2, \ldots, N)$, the x_i would be considered as an abnormal value and be rejected, where σ is the standard deviation, and $\overline{x}$ is the mean of all measurement values.

Recently, the concept of neighborhood (see, e.g., [29, 30]) has been applied to define different types of covering-based rough set [31–34]. A neighborhood based on static error range is defined [35]. Although showing similarities, it is essentially different from ours. The proposed neighborhood is considered as the distribution of the data error and the confidence interval. The neighborhood boundaries for different attributes of the same database are completely different. An example of neighborhood boundary vector is listed in Table 2.

2.3. A Decision System Based on Measurement Errors with Test Costs and Misclassification Costs. In many applications, the test cost must be taken into account [5]. Test cost is the money, time, or other resources that we pay for collecting a data item of an object [8, 9, 18, 19, 36]. In addition to the test costs, it is also necessary to consider misclassification costs. A decision cannot be made if the misclassification costs are unreasonable [5]. More recently, researchers have begun to consider both test costs and misclassification costs [8, 13, 17].

Now, we take into account both test and misclassification costs as well as normal distribution measurement errors. We have defined this decision system in [37] as follows.

Definition 3. A decision system based on measurement errors with test costs and misclassification costs (MEDS-TM) S is the 8-tuple:

$$S = (U, C, d, V, I, n, \mathrm{tc}, \mathrm{mc}), \tag{4}$$

where U, C, d, V, I, and n have the same meanings as Definition 2, $\mathrm{tc} : C \rightarrow \mathbb{R}^+ \cup \{0\}$ is the test cost function, and $\mathrm{mc} : k \times k \rightarrow \mathbb{R}^+ \cup \{0\}$ is the misclassification cost function, where $k = |I_d|$.

Here, we consider only the sequence-independent test-cost-sensitive decision system. There are a number of test-cost-sensitive decision systems. A hierarchy of decision systems consisting of six models was proposed [18]. For any $B \subseteq C$, the test cost function tc is given by $\mathrm{tc}(B) = \sum_{a \in B} \mathrm{tc}(a)$.

The test cost function can be stored in a vector. An example of text cost vector is listed in Table 3.

The misclassification cost [38–40] is the penalty that we receive while deciding that an object belongs to class i when its real class is j [8]. The misclassification cost function mc is defined as follows:

(1) $\mathrm{mc} : k \times k \rightarrow \mathbb{R}^+ \cup \{0\}$ is the misclassification cost function, which can be represented by a matrix $\mathrm{MC} = \{\mathrm{mc}_{k \times k}\}$, where $k = |I_d|$,

(2) $\mathrm{mc}[m, n]$ is the cost of misclassifying an example from "class m" to "class n",

(3) $\mathrm{mc}[m, m] = 0$.

The following example gives us an intuitive understanding of the decision system based on measurement errors with test costs and misclassification costs.

Example 4. Table 1 is a *Liver* decision system. Tables 2 and 3 are error boundary vector and test cost vector of *Liver* decision system, respectively. consider

$$\mathrm{mc} = \begin{bmatrix} 0 & 2000 \\ 200 & 0 \end{bmatrix}. \tag{5}$$

That is, the test costs of Mcv, Alkphos, Sgpt, Sgot, Gammagt, and Drinks are \$26, \$17, \$34, \$45, \$38, and \$5, respectively. In *Liver* dataset, the *Selector* field is used to split data into two sets. Here, a false negative prediction (FN), that is, failing to detect *liver* disorders, may well have fatal consequences, whereas a false positive prediction (FP), that is, diagnosing *liver* disorders for a patient that does not actually have them, may be less serious [41]. Therefore, a higher penalty of \$2000 is paid for FN prediction and \$200 is paid for FP prediction.

Obviously, if tc and mc are not considered, the MEDS-TM degrades to a decision system with measurement errors (MEDS) (see, e.g., [28]). Therefore, the MEDS-TM is a generalization of the MEDS.

3. Covering-Based Rough Set with Measurement Errors

As a technique to deal with granularity in information systems, rough set theory was proposed by Pawlak [42]. Since then, we have witnessed a systematic, worldwide growth of interest in rough set theory [43–52] and its applications [53, 54]. Recently, there has been growing interest in covering-based rough set. In this section, we introduce normal distribution measurement errors to covering-based rough set. The new model is called covering-based rough set with measurement errors. Then, we define a new cost-sensitive feature selection problem on this covering-based rough set.

3.1. Covering-Based Rough Set with Measurement Errors. The covering-based rough set with measurement errors is a natural extension of the classical rough set. If all attributes are error free, the covering-based rough set model degenerates to the classical one. With the definition of the MEDS, a new neighborhood is defined as follows.

Definition 5 (see [28]). Let $S = (U, C, d, V, I, n)$ be a decision system with measurement errors. Given $B \subseteq C$ and $x_i \in U$, the neighborhood of x_i with reference to measurement errors on the feature set B is defined as

$$n_B(x_i) = \{x \in U \mid \forall a \in B, |a(x) - a(x_i)| \leq 2n(a)\}. \quad (6)$$

That means the value of the measurement error of attribute a in $[-n(a), +n(a)]$. According to Definition 5, we know that the neighborhood $n_B(x_i)$ is the intersection of multiple basic neighborhoods. Therefore, we obtain

$$n_B(x_i) = \bigcap_{a \in B} n_{\{a\}}(x_i). \quad (7)$$

Although showing similarities, the neighborhood defined in [35] is essentially different from ours in two ways. First, a fixed boundary of neighborhood is used for different datasets. In contrast, the boundaries of neighborhood in our model are computed according to the values of attributes. Then, the uniform distribution is considered in [35]. In contrast, we introduce the normal distribution to our model. As mentioned earlier, the normal distribution is found to be applicable over almost the whole of science measurement.

Normal distribution is a plausible distribution for measurement errors. In statistics, "3-sigma" rule states that over 99.73% (95.45%) of measurement data will fall within three (two) standard deviations of the mean [55]. We introduce this rule to our model and present a new neighborhood considering both the error distribution and the confidence interval. The proportion of small measurement errors is higher than large ones. Any value in the measurement that exceeds the three standard deviations from the mean should be discarded. Therefore, the measurement errors with no more than a difference of 3σ (2σ) should be viewed as a granule. In view of this, we introduce the relationship between the error boundary and the standard deviation in the following proposition.

Proposition 6. *Let the error boundary $n(a) = 3\sigma$ and Pr be the confidence level. one has about $\Pr = 99.73\%$ of cases within $n(a) = \pm 3\sigma$.*

According to Proposition 6, we have about $\Pr = 99.73\%$ of cases within $n(a) = \pm 3\sigma$. If $n(a) = 2\sigma$, we have about $\Pr = 95.45\%$ of cases within $\pm n(a)$. According to Definition 5, every item belongs to its own neighborhood. This is formally given by the following theorem.

Theorem 7. *Let $S = (U, C, d, V, I, n)$ be a decision system with measurement errors and $B \subseteq C$. The set $\{n_B(x_i) \mid x_i \in U\}$ is a covering of U.*

Proof. Given for all $x \in U$, for all $a \in B$, $|a(x) - a(x)| = 0$, $|a(x) - a(x)| \leq 2n(a)$, $x \in n_B(x)$.

Therefore, for all $x \in U$, $n_B(x) \neq \emptyset$, and for any $B \subseteq C$, $\bigcup_{x \in U} n_B(x) = U$.

Hence, the set $\{n_B(x_i) \mid x_i \in U\}$ is a covering of U. This completes the proof. □

Now, we discuss the lower and upper approximations as well as the boundary region of rough set in the new model.

Table 4: A subtable of the *Liver* decision system.

Patient	a_1	a_2	a_3	d
x_1	0.31	0.23	0.08	y
x_2	0.14	0.38	0.23	y
x_3	0.25	0.40	0.40	y
x_4	0.60	0.46	0.51	n
x_5	0.41	0.64	0.62	n
x_6	0.35	0.50	0.75	n

Table 5: An example of adaptive neighborhood boundary vector.

a	a_1	a_2	a_3
Neighborhood boundaries	±0.069	±0.087	±0.086

Table 6: The neighborhood of objects on different test sets.

x	$\{a_1\}$	$\{a_1, a_2\}$	$\{a_1, a_3\}$	$\{a_1, a_2, a_3\}$
x_1	$\{x_1, x_3, x_5, x_6\}$	$\{x_1, x_3\}$	$\{x_1\}$	$\{x_1\}$
x_2	$\{x_2, x_3\}$	$\{x_2, x_3\}$	$\{x_2, x_3\}$	$\{x_2, x_3\}$
x_3	$\{x_1, x_2, x_3, x_6\}$	$\{x_1, x_2, x_3, x_6\}$	$\{x_2, x_3\}$	$\{x_2, x_3\}$
x_4	$\{x_4\}$	$\{x_4\}$	$\{x_4\}$	$\{x_4\}$
x_5	$\{x_1, x_5, x_6\}$	$\{x_5, x_6\}$	$\{x_5, x_6\}$	$\{x_5, x_6\}$
x_6	$\{x_1, x_3, x_5, x_6\}$	$\{x_3, x_5, x_6\}$	$\{x_5, x_6\}$	$\{x_5, x_6\}$

Definition 8 (see [28]). Let $S = (U, C, d, V, I, n)$ be a decision system with measurement errors and N_B a neighborhood relation on U, where $B \subseteq C$. We call $\langle U, N_B \rangle$ a neighborhood approximation space. For arbitrary $X \subseteq U$, the lower approximation and the upper approximation of X in $\langle U, N_B \rangle$ are defined as

$$\begin{gathered} \underline{N_B}(X) = \{x_i \mid x_i \in U \wedge n_B(x_i) \subseteq X\}, \\ \overline{N_B}(X) = \{x_i \mid x_i \in U \wedge n_B(x_i) \cap X \neq \emptyset\}. \end{gathered} \quad (8)$$

The positive region of $\{d\}$ concerning $B \subseteq C$ is defined as $\mathrm{POS}_B(\{d\}) = \bigcup_{X \in U/\{d\}} \underline{N_B}(X)$ [42, 56].

Definition 9. Let $S = (U, C, d, V, I, n)$ be a decision system with measurement errors, for all $X \subseteq U$, $\overline{N_B}(X) \supseteq X \supseteq \underline{N_B}(X)$. The boundary region of X in $\langle U, N_B \rangle$ is defined as

$$BN_B(X) = \overline{N_B}(X) - \underline{N_B}(X). \quad (9)$$

Generally, a covering is produced by a neighborhood boundary. The inconsistent object in a neighborhood is defined as follows.

Definition 10 (see [28]). Let $S = (U, C, d, V, I, n)$ be a decision system with measurement errors, $B \subseteq C$, and $x, y \in U$. In the set of $n_B(x)$, for all $y \in n_B(x)$ is called an inconsistent object if $d(y) \neq d(x)$. The set of inconsistent objects in $n_B(x)$ is

$$\mathrm{ic}_B(x) = \{y \in n_B(x) \mid d(y) \neq d(x)\}. \quad (10)$$

The number of inconsistent objects is denoted as $|\mathrm{ic}_B(x)|$.

Table 7: Approximations of object subsets on different test sets.

	X	$\{a_1\}$	$\{a_1, a_2\}$	$\{a_1, a_3\}$	$\{a_1, a_2, a_3\}$
$\underline{N_B}(X)$	X_1	$\{x_2\}$	$\{x_1, x_2\}$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3\}$
	X_2	$\{x_4\}$	$\{x_4, x_5\}$	$\{x_4, x_5, x_6\}$	$\{x_4, x_5, x_6\}$
$\overline{N_B}(X)$	X_1	$\{x_1, x_2, x_3, x_5, x_6\}$	$\{x_1, x_2, x_3, x_6\}$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3\}$
	X_2	$\{x_1, x_3, x_4, x_5, x_6\}$	$\{x_3, x_4, x_5, x_6\}$	$\{x_4, x_5, x_6\}$	$\{x_4, x_5, x_6\}$

Using a specific example, we explain the lower approximations, the upper approximations, the boundary regions, and the inconsistent objects of the neighborhood.

Example 11. A decision system with neighborhood boundaries is given in Tables 4 and 5. Table 4 is a subtable of Table 1. Let $U = \{x_1, x_2, \ldots, x_6\}$, $C = \{a_1, a_2, a_3\}$, and $D = \{d\} = \{\text{Selector}\}$, where $a_1 = \text{Mcv}$, $a_2 = \text{Alkphos}$, and $a_3 = \text{Sgpt}$. $n_B(x)$ are listed in Table 6, where B takes values listed as column headers, and x takes values listed in each row. According to Definition 10, the inconsistent object in $n_{\{a_1\}}(x_1)$ is $\text{ic}_{\{a_1\}}(x_1) = \{x_5, x_6\}$.

In addition, U is divided into a set of equivalence classes by $\{d\}$. $U/\{d\} = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$. Let $X_1 = \{x_1, x_2, x_3\}$ and $X_2 = \{x_4, x_5, x_6\}$. $\underline{N_B}(X)$ and $\overline{N_B}(X)$ are listed in the first part and the second part of Table 7, respectively. Here, B takes values listed as column headers, and X takes values listed in each row.

The positive regions and the boundary regions of U on different test sets can be computed from Table 7:

(1) $\text{POS}_{\{a_1\}}(\{d\}) = \{x_2, x_4\}$, $BN_{\{a_1\}}(\{d\}) = \{x_1, x_3, x_5, x_6\}$,

(2) $\text{POS}_{\{a_1,a_2\}}(\{d\}) = \{x_1, x_2, x_4, x_5\}$, $BN_{\{a_1,a_2\}}(\{d\}) = \{x_3, x_6\}$,

(3) $\text{POS}_{\{a_1,a_3\}}(\{d\}) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $BN_{\{a_1,a_3\}}(\{d\}) = \emptyset$,

(4) $\{a_1, a_3\}$ has the same approximating power as C.

3.2. Minimal Cost Feature Selection Problem. In this work, we focus on cost-sensitive feature selection based on test costs and misclassification costs. Unlike reduction problems, we do not require any particular property of the decision system to be preserved. The objective of feature selection is to minimize the average total cost through considering a trade-off between test costs and misclassification costs. Cost-sensitive feature selection problem is called the *feature selection with minimal average total cost (FSMC) problem*.

Problem 1. The FSMC problem:

input: $S = (U, C, d, V, I, n, \text{tc}, \text{mc})$,

output: $R \subseteq C$,

optimization objective: minimize the average total cost (ATC).

The FSMC problem is a generalization of classical minimal reduction problem. On the one hand, several factors should be considered such as the test costs and misclassification costs as well as normal distribution measurement errors. These factors are all intrinsic to data in real applications. On the other hand, the minimal average total cost is the optimization objective through considering the trade-off between the two kinds of costs. Compared with the accuracy, the average total cost is more general metric in data mining applications [36]. The following is a five-step process to compute the average total cost.

(1) Let B be a selected feature set. Given for all $x \in U$, we compute the neighborhood space $n_B(x)$.

(2) Let $U' = n_B(x)$ and let $d(x)$ be the decision value of object x. Let $|U'_m|$ and $|U'_n|$ be the number of m-class and n-class, respectively, where $m, n \in \{I_d\}$. Let the misclassification cost $\text{MC}_m = \text{mc}[m, n] \times |U'_m|$ and $\text{MC}_n = \text{mc}[n, m] \times |U'_n|$, respectively. In order to minimize the misclassification cost of the set U', we assign one class $d'(x)$ for all objects in U'. Let $\text{mc}(U', B)$ be the minimal value of MC_m and MC_n.

(3) For any $x \in U'$, the assigned class $d'(x) = n$-class if $\text{mc}(U', B) = \text{MC}_m$ and $d'(x) = m$-class if $\text{mc}(U', B) = \text{MC}_n$, where $\text{mc}[m, n]$ is the cost of classifying an object of the m-class to the n-class.

(4) The decision value of the object x depends on the value with the max number of $d'(x)$. The misclassification cost of the object x is $\text{mc}^*(x)$. If $d(x) = m$ and $d'(x) = n$, $\text{mc}^*(x) = \text{mc}[m, n]$. Conversely, $\text{mc}^*(x) = \text{mc}[n, m]$ if $d(x) = n$ and $d'(x) = m$. Therefore, we compute the average misclassification cost (AMC) as follows:

$$\overline{\text{mc}}(U, B) = \frac{\sum_{x \in U} \text{mc}^*(x)}{|U|}. \tag{11}$$

(5) The average total cost (ATC) is given by

$$\text{ATC}(U, B) = \text{tc}(B) + \overline{\text{mc}}(U, B). \tag{12}$$

The main aim of feature selection is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features [57]. In this context, rather than selecting a minimal feature subset, we choose a feature subset in order to minimize the average total cost. The minimal average total cost is given by

$$\text{ATC}(U, B) = \min\left\{\text{ATC}\left(U, B'\right) \mid B' \subseteq C\right\}. \tag{13}$$

The following example gives an intuitive understanding.

Example 12. A decision system with neighborhood boundaries is given by Tables 4 and 5. Let $C = \{a_1, a_2, a_3\}$, $B = \{a_1, a_2\}$, and $D = \{d\}$. Let $\text{tc} = [8, 23, 19]$ and $\text{mc} = \left[\begin{smallmatrix} 0 & 180 \\ 60 & 0 \end{smallmatrix}\right]$.

Table 8: The neighborhood of objects on $B\{a_1, a_2\}$.

U	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1	0	1	0	0	0
x_2	0	1	1	0	0	0
x_3	1	1	1	0	0	1
x_4	0	0	0	1	0	0
x_5	0	0	0	0	1	1
x_6	0	0	1	0	1	1

Table 9: The number of different classes.

d	x_1	x_2	x_3	x_4	x_5	x_6
y	2	2	4	0	1	2
n	0	0	0	1	1	1

Table 10: The difference of decision attributes.

U	x_1	x_2	x_3	x_4	x_5	x_6
$d'(x)$	Y	Y	Y	N	Y	Y
$d(x)$	Y	Y	Y	N	N	N

Step 1. $n_B(x_i)$ is the neighborhood of $x_i \in U$, which is listed in Table 8. If $x_j \in n_B(x_i)$, the value at ith row and jth column is set to 1; otherwise, it is set to 0.

Step 2. Since the set of $n_B(x_i) \subseteq \mathrm{POS}_B(\{d\})$, the $\mathrm{mc}(n_B(x_i), B) = 0$, where $i = 1, 2, 4, 5$. The set of $n_B(x_3) = \{x_1, x_2, x_3, x_6\}$ has two kinds of classes, which should be adjusted to one class. Since $\mathrm{mc}(n_B(x_3), B) = \min(60 \times 1, 180 \times 3)$, for any $x \in U'$, $d(x) =$ “y”. In the same way, in order to minimize the cost of $\mathrm{mc}(n_B(x_6), B) = \min(60 \times 2, 180 \times 1)$, we adjust all classes of elements in $n_B(x_6)$ to “y”.

Step 3. We can obtain the new class of each test. We count the number of different classes of each test, which is listed in Table 9.

Step 4. From Table 9, we select d_m with the maximal of $d'(x_i)$ as the class value of x_i. The original decision attribute values $d(x)$ and $d'(x)$ are listed in Table 10. From this Table, we know $d(x_5) \neq d'(x_5)$ and $d(x_6) \neq d'(x_6)$. Therefore, the average misclassification cost $\overline{\mathrm{mc}}(U, B) = (60 + 60)/6 = 20$.

Step 5. The average total cost is $\mathrm{ATC}(U, B) = (8 + 23) + 20 = 51$.

In order to search a minimal cost feature subset, we can define a problem to deal with this issue. Under the context of MEDS-TM, this problem will be called cost-sensitive feature selection problem or the minimal cost feature selection (FSMC) problem. Compared with the minimal test cost reduct (MTR) problem (see, e.g., [15, 16]), the FSMC problem should not only consider the test costs but also take the misclassification costs into account. When the misclassification costs are too large compared with test costs, the total test cost equals the total cost. In this case, the FSMC problem coincides with the MTR problem.

4. Algorithms

We propose the δ-weighted heuristic algorithm to address the minimal cost feature selection problem. In order to evaluate the performance of a heuristic algorithm, an exhaustive algorithm is also needed. Exhaustive searches are also known as backtracking algorithms which look for every possible way to search for an optimal result. In this section, we review our exhaustive algorithm and propose a heuristic algorithm for this new feature selection problem.

4.1. The Backtracking Feature Selection Algorithm. We have proposed an exhaustive algorithm in [37] that is based on the backtracking. The backtracking algorithm can reduce the search space significantly through three pruning techniques. The backtracking feature selection 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 feature subset with minimal average total cost,

(2) $\mathrm{cmc} = \overline{\mathrm{mc}}(U, R)$ is currently minimal average total cost,

(3) backtracking$(R, 0)$.

A feature subset with the ATC will be stored in R at the end of the algorithm execution. Generally, the search space of the feature selection algorithm is $2^{|C|}$. In order to deal with this issue, there are a number of algorithms such as particle swarm optimization algorithms [58], genetic algorithms [1], and backtracking algorithms [59] in real applications.

In Algorithm 1, three pruning techniques are employed to reduce the search space in feature selection. Firstly, Line 1 indicates that the variable i starts from l instead of 0. Whenever we move forward through the recursive procedure, the lower bound is increased. And then, the second pruning technique is shown in Lines 3 through 5. In the real applications, the misclassification costs are nonnegative. In this way, the feature subsets B will be discarded if the test cost of B is larger than the current minimal average total cost (cmc). This technique can prune most branches. Finally, Lines 6 through 8 indicate that if the new feature subset produce a high cost along with decreasing misclassification cost, the current branch will never produce the feature subset with the minimal total cost.

4.2. The δ-Weighted Heuristic Feature Selection Algorithm. In order to deal with the minimal feature selection problem, we design the δ-weighted heuristic feature selection algorithm. The algorithm framework is listed in Algorithm 2 containing two main steps. First, the algorithm adds the current best feature a to B according to the heuristic function $f(B, a_i, c(a_i))$ until B becomes a superreduct. Then, delete the feature a from B guaranteeing B with the current minimal total cost. In Algorithm 2, lines 5 and 7 contain the key code of the addition. Lines 10 to 14 show the steps of deletion.

According to Definition 10, the number of inconsistent objects $|\mathrm{ic}_B(x)|$ in neighborhood $n_B(x)$ is useful in evaluating

```
Input: $(U, C, d, \{V_a\}, \{I_a\}, n, \text{tc}, \text{mc})$, select tests $R$, current level test index lower bound $l$
Output: A set of features $R$ with ATC and cmc, they are global variables
Method: backtracking
for $(i = l; i < |C|; i++)$ do
    $B = R \cup \{a_i\}$
    //Pruning for too expensive test cost
    if $(\text{tc}(B) > \text{cmc})$ then
        continue;
    end if
    //Pruning for non-decreasing total cost and decreasing misclassification cost
    if $((\text{ATC}(U, B) \geq \text{ATC}(U, R))$ and $(\text{mc}(B) < \text{mc}(R))$ then
        continue;
    end if
    if $(\text{ATC}(U, B) < \text{cmc}))$ then
        $\text{cmc} = \text{ATC}(U, B)$; //Update the minimal total cost
        $R = B$; //Update the set of features with minimal total cost
    end if
    backtracking $(B, i + 1)$;
end for
```

Algorithm 1: A backtracking algorithm to the FSMC problem.

```
Input: $(U, C, d, \{V_a\}, \{I_a\}, n, \text{tc}, \text{mc})$
Output: A feature subset with minimal total cost
Method:
$B = \emptyset$;
//Addition
$CA = C$;
while $(\text{POS}_B(D) \neq \text{POS}_C(D))$ do
    for each $a \in CA$ do
        Compute $f(B, a, c(a'))$;
    end for
    Select $a'$ with the maximal $f(B, a', c(a'))$;
    $B = B \cup \{a'\}$; $CA = CA - \{a'\}$;
end while
//Deletion
while $(\text{ATC}(U, B) > \text{ATC}(U, B - \{a\}))$ do
    for each $a \in B$ do
        Compute $\text{ATC}(U, B - \{a\})$;
    end for
    Select $a'$ with the minimal $\text{ATC}(U, B - \{a'\})$;
    $B = B - \{a'\}$;
end while
return $B$;
```

Algorithm 2: An addition-deletion cost-sensitive feature selection algorithm.

the quality of a neighborhood block. Now, we introduce the following concepts.

Definition 13 (see [35]). Let $S = (U, C, D, V, I, n)$ be a decision system with measurement errors, $B \subseteq C$, and $x \in U$. The total number of such objects with respect to U is

$$\text{nc}_B(S) = \Sigma_{x \in U} \left|\text{ic}_B(x)\right|, \tag{14}$$

and the positive region is

$$\text{pc}_B(S) = \Sigma_{x \in \text{POS}_C(D)} \left|\text{ic}_B(x)\right|. \tag{15}$$

According to Definition 13, we know that B is a superreduct if and only if $\text{pc}_B(S) = 0$. Now, we propose the δ-weighted heuristic information function:

$$f(B, a_i, c(a_i)) = \left(\text{pc}_B(S) - \text{pc}_{B \cup \{a_i\}}(S)\right)\left(1 + \frac{\delta}{c(a_i)}\right), \tag{16}$$

where $c(a_i)$ is the test cost of the attribute a_i, and $\delta \geq 0$ is a user-specified parameter. In this heuristic information function, the attributes with lower cost have bigger significance.

Table 11: Database information.

No.	Name	Domain	$\|U\|$	$\|C\|$	$D = \{d\}$
1	*Liver*	Clinic	345	6	Selector
2	*Wdbc*	Clinic	569	30	Diagnosis
3	*Wpbc*	Clinic	198	33	Outcome
4	*Diab*	Clinic	768	8	Class
5	*Iono*	Physics	351	34	Class
6	*Credit*	Commerce	690	15	Class

We can adjust the significance of test cost through different δ settings. If $\delta = 0$, test costs are essentially not considered.

5. Experiments

In this section, we try to answer the following questions by experimentation. The first two questions concern the backtracking algorithm, and the others concern the heuristic algorithm.

(1) Is the backtracking algorithm efficient?

(2) Is the heuristic algorithm appropriate for the minimal cost feature selection problem?

(3) How does the minimal total cost change for different misclassification cost settings?

5.1. Data Generation. Experiments are carried out on six standard datasets obtained from the UCI repository: *Liver*, *Wdbc*, *Wpbc*, *Diab*, *Iono*, and *Credit*. The first four datasets are from medical applications where *Wpbc* and *Wdbc* are the Wisconsin breast cancer prognosis and diagnosis datasets, respectively. The *Liver* and *Diab* are *liver* disorder and diabetes datasets, respectively. The *iono* stands for the Ionosphere, which is from physics applications. The *Credit* dataset is from commerce applications.

Table 11 shows a brief description of each dataset. Most datasets from the UCI library [60] have no intrinsic measurement errors, test costs, and misclassification costs. In order help to study the performance of the feature selection algorithm, we will create these data for experimentations.

Step 1. Each dataset should contain exactly one decision attribute and have no missing value. To make the data easier to handle, data items are normalized from their value into a range from 0 to 1.

Step 2. We produce the $n(a)$ for each original test according to (3). The $n(a)$ is computed according to the value of databases without any subjectivity.

Three kinds of neighborhood boundaries of different databases are shown in Table 12. These neighborhood boundaries are the maximal, the minimal, and the average neighborhood boundaries of all attributes, respectively. The precision of $n(a)$ can be adjusted through Δ setting, and we set Δ to be 0.01 in our experiments.

Table 12: Generated neighborhood boundaries for different databases.

Dataset	Minimal	Maximal	Average
Liver	0.022	0.130	±0.058
Wdbc	0.012	0.080	±0.046
Wpbc	0.022	0.112	±0.062
Diab	0.018	0.118	±0.062
Iono	0.090	0.174	±0.122
Credit	0.002	0.112	±0.044

Table 13: Number of steps for the backtracking algorithm.

Dataset	Search space	Minimal steps	Maximal steps	Average steps
Liver	2^6	8	34	21.27
Wdbc	2^{30}	18	113	54.95
Wpbc	2^{33}	10	76	44.34
Diab	2^8	28	102	58.50
Iono	2^{34}	107	2814	663.41
Credit	2^{15}	105	2029	618.14

Step 3. We produce test costs, which are always represented by positive integers. For any $a \in U$, $c(a)$ is set to a random number in [12, 55] subject to the uniform distribution.

Step 4. The misclassification costs are always represented by nonnegative integers. We produce the matrix of misclassification costs mc as follows:

(1) $\mathrm{mc}[m, m] = 0$.

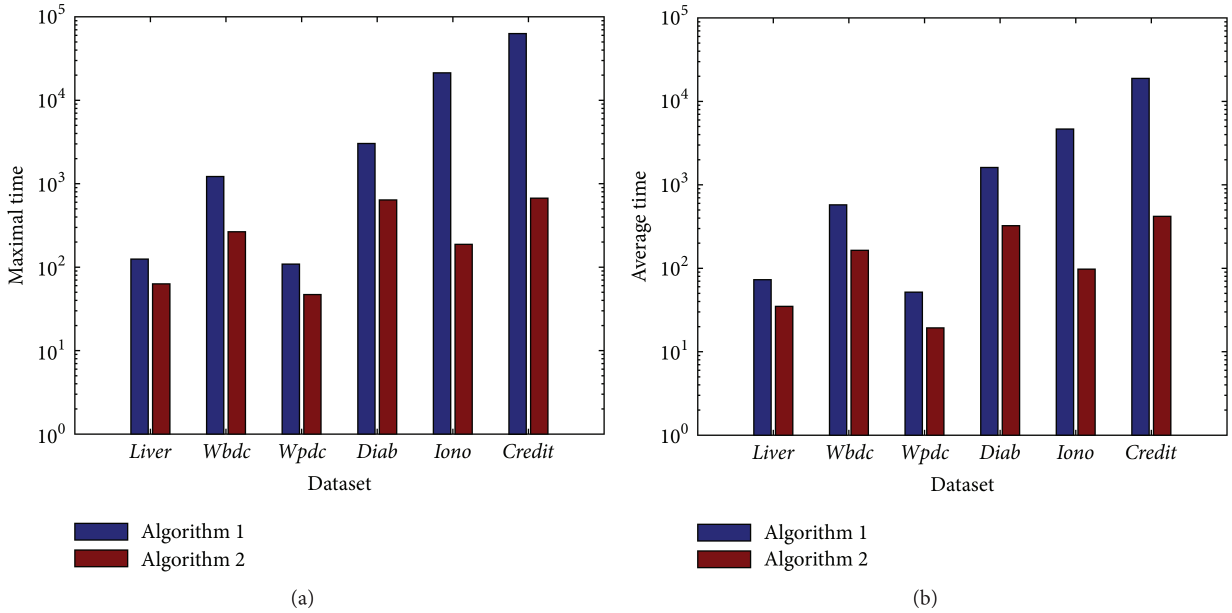
(2) $\mathrm{mc}[m, n]$ and $\mathrm{mc}[n, m]$ are set to a random number in $[100, 1000]$, respectively.

5.2. Efficiencies of the Two Algorithms. First, we study the efficiency of the backtracking algorithm. Specifically, experiments are undertaken with 100 different test cost settings. The search space and the number of steps for the backtracking algorithm are listed in Table 13. From the results, we note that the pruning techniques significantly reduce the search space. Therefore, the pruning techniques are very effective.

Second, from Table 13, we note that the number of steps does not simply rely on the size of the dataset. *Wpbc* is much larger than *Credit*; however, the number of steps is smaller. For some medium sized datasets, the backtracking algorithm is an effective method to obtain the optimal feature subset.

Third, we compare the efficiency of the heuristic algorithm and the backtracking algorithm. Specifically, experiments are undertaken with 100 different test cost settings on six datasets listed in Table 11. For the heuristic algorithm, λ is set to 1. The average and maximal run times for both algorithms are shown in Figure 1, where the unit of run time is on millisecond. From the results, we note that the heuristic algorithm is more stable in terms of run-time.

In a word, when we do not consider the run time, the backtracking algorithm is an effective method for many



(a)

(b)

FIGURE 1: Run time comparison: (a) maximal time and (b) average time.

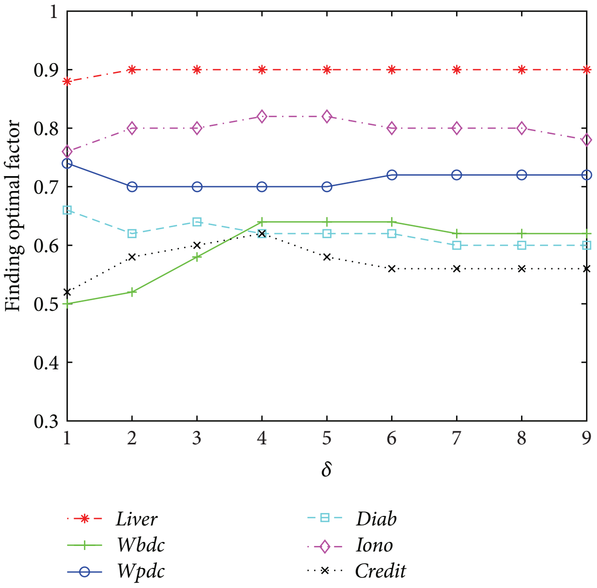


FIGURE 2: Finding optimal factor.

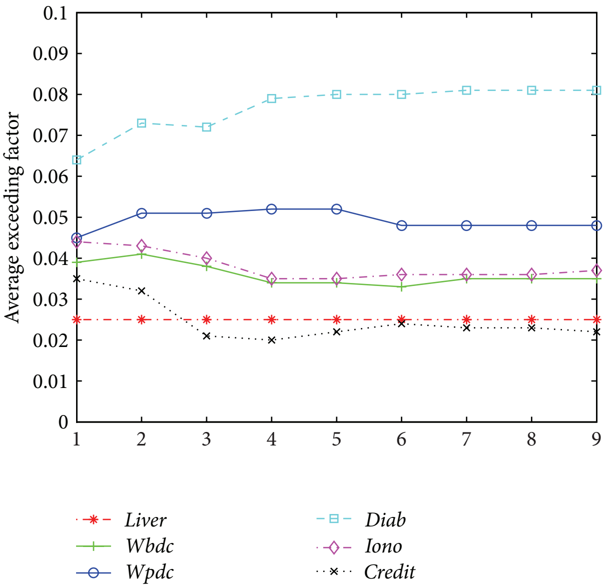
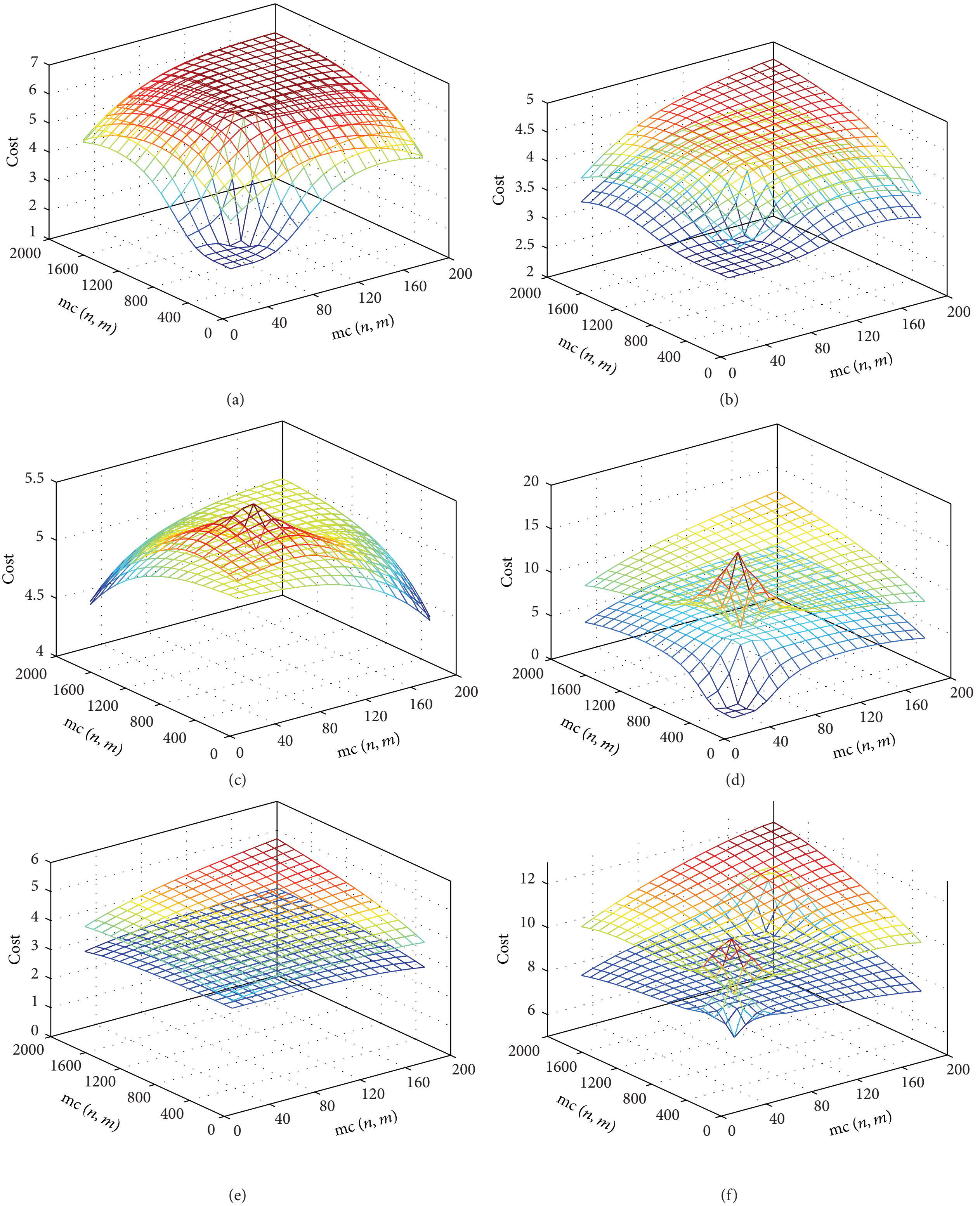


FIGURE 3: Average exceeding factor.

datasets. In real applications, when the run times of the backtracking algorithm are unacceptable, the heuristic algorithm must be employed.

5.3. Effectiveness of the Heuristic Algorithm. We let $\delta = 1, 2, \ldots, 9$. The precision of $n(a)$ can be adjusted through Δ setting, and we let Δ to be 0.01 on all datasets except *Wdbc* and *Wpbc*. The $\Delta = 0.01$ gets small neighborhood for *Wdbc* and *Wpbc* datasets; hence, we let $\Delta = 0.05$ for the two datasets. As mentioned earlier, the parameter Δ plays an important role. The data of our experiments come from real applications, and the errors are not given by the dataset. In this paper, we consider only some possible error ranges.

The algorithm runs 100 times with different test cost settings and different δ setting on all datasets. Figure 2 shows the results of finding optimal factors. From the results, we know that the test cost plays a key role in this heuristic algorithm. As shown in Figure 2, the performance of the algorithm is completely different for different settings of δ. Data for $\delta = 0$ are not included in the experiment results because respective results are incomparable to others. Figure 3 shows the average exceeding factors. These display



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Test cost and average total cost: (a) *Liver*; (b) *Wdbc*; (c) *Wpbc*; (d) *Diab*; (e) *Iono*; (f) *Credit*.

Table 14: The optimal feature subset based on different misclassification costs.

MisCost1	MisCost2	Test costs	Total cost	Feature subset
50	500	3.00	3.70	[1, 3, 27]
100	1000	4.00	4.35	[1, 3, 15, 29]
150	1500	4.00	4.53	[1, 3, 15, 29]
200	2000	4.00	4.70	[1, 3, 15, 29]
250	2500	4.00	4.88	[1, 3, 15, 29]
300	3000	5.00	5.00	[1, 12, 15, 27]

the overall performance of the algorithm from a statistical perspective.

From the results, we observe the following:

(1) the quality of the results is related to different datasets. It is because that the error range and heuristic information are all computed according to the values of dataset,

(2) the results of the finding optimal factor are acceptable on most of datasets except *Wdbc*. The better results can be obtained through the smaller Δ; however, the number of selected features will be smaller,

(3) the average exceeding factor is less than 0.08 in most cases. In other words, the results are acceptable.

5.4. The Results for Different Cost Settings. In this section, we study the changes of the minimal total cost for different misclassification cost settings. Table 14 is the optimal feature subset based on different misclassification costs for *Wdbc* dataset. The ratio of two misclassification costs is set 10 in this experiment.

As shown in this table, when the misclassification costs are low, the algorithm avoids undertaking expensive tests.

When the misclassification cost is too large compared with the test cost, the FSMC problem coincides with the MTR problem. Therefore, FSMC problem is a generalization of MTR problem.

In the last row of Table 14, the test cost of the subset [24, 31, 45, 55] equals the total cost; therefore, the misclassification cost is 0, and this feature subset is a reduct.

The changes of test costs versus the average minimal total cost are also shown in Figure 4. In real world, we could not select expensive tests when misclassification costs are low. Figure 4 shows this situation clearly. From the results, we observe the following.

(1) As shown in Figures 4(a), 4(b), 4(e), and 4(f), when the test costs remain unchanged, the total costs increase linearly along with the increasing misclassification costs.

(2) If the misclassification costs are small enough, we may give up the test. Figure 4(d) shows that when the misclassification costs are \$30 and \$300, the test cost is zero, and the total cost is the most expensive.

(3) As shown in Figures 4(a) and 4(c), the total costs increase along with the increasing misclassification costs. The total costs remain the same when the total costs equal test costs.

6. Conclusions

In this paper, we built a new covering-based rough set model with normal distribution measurement errors. A new cost-sensitive feature selection problem is defined based on this model. This new problem has a wide application area for two reasons. One is that the resource that one can afford is often limited. The other is that data with measurement errors under considered is ubiquitous. A backtracking algorithm and a heuristic algorithm are designed. Experimental results indicate the efficiency of the backtracking algorithm and the effectiveness of the heuristic algorithm.

With regard to future research, much work needs to be undertaken. First, other realistic data models with neighborhood boundaries can be built. Second, the current implementation of the algorithm deals only with binary class problems that is the principal limitation. In the future, the extending algorithm needs to be proposed to cope with multivariate class problems. A third point to be considered in future research is that one can borrow ideas from [61–63] to design other exhaustive and heuristic algorithms. In summary, this study suggests new research trends concerning covering-based rough set theory, feature selection problem, and cost-sensitive learning.

Acknowledgments

This work is in part supported by the National Science Foundation of China under Grant no. 61170128, the Natural Science Foundation of Fujian Province, China, under Grant no. 2012J01294, the State Key Laboratory of Management and Control for Complex Systems Open Project under Grant no. 20110106, and the Fujian Province Foundation of Higher Education under Grant no. JK2012028.

References

[1] P. Lanzi, "Fast feature selection with genetic algorithms: a filter approach," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1997.

[2] T. L. B. Tseng and C. C Huang, "Rough set-based approach to feature selection in customer relationship management," *Omega*, vol. 35, no. 4, pp. 365–383, 2007.

[3] N. Zhong, J. Z. Dong, and S. Ohsuga, "Using rough sets with heuristics to feature selection," *Journal of Intelligent Information Systems*, vol. 16, no. 3, pp. 199–214, 2001.

[4] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, vol. 454, Springer, 1998.

[5] Y. Weiss, Y. Elovici, and L. Rokach, "The CASH algorithm-cost-sensitive attribute selection using histograms," *Information Sciences*, vol. 222, pp. 247–268, 2013.

[6] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 2001.

[7] W. Fan, S. Stolfo, J. Zhang, and P. Chan, "Adacost: misclassification cost-sensitive boosting," in *Proceedings of the 16th International Conference on Machine Learning*, 1999.

[8] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in Induction*, Academic Press, New York, NY, USA, 1966.

[9] M. Pazzani, C. Merz, P. M. K. Ali, T. Hume, and C. Brunk, "Reducing misclassification costs," in *Proceedings of the 11th International Conference of Machine Learning (ICML '94)*, Morgan Kaufmann, 1994.

[10] G. Fumera and F. Roli, "Cost-sensitive learning in support vector machines," in *Proceedings of VIII Convegno Associazione Italiana per L' Intelligenza Artificiale*, 2002.

[11] C. X. Ling, Q. Yang, J. N. Wang, and S. C. Zhang, "Decision trees with minimal costs," in *Proceedings of the 21st International Conference on Machine learning*, 2004.

[12] R. Greiner, A. J. Grove, and D. Roth, "Learning cost-sensitive active classifiers," *Artificial Intelligence*, vol. 139, no. 2, pp. 137–174, 2002.

[13] S. Ji and L. Carin, "Cost-sensitive feature acquisition and classification," *Pattern Recognition*, vol. 40, pp. 1474–1485, 2007.

[14] N. Lavrac, D. Gamberger, and P. Turney, "Cost-sensitive feature reduction applied to a hybrid genetic algorithm," in *Proceedings of the 7th International Workshop on Algorithmic Learning Theory (ALT '96)*, 1996.

[15] F. Min, H. P. He, Y. H. Qian, and W. Zhu, "Test-cost-sensitive attribute reduction," *Information Sciences*, vol. 181, pp. 4928–4942, 2011.

[16] R. Susmaga, "Computation of minimal cost reducts," in *Foundations of Intelligent Systems*, Z. Ras and A. Skowron, Eds., vol. 1609 of *Lecture Notes in Computer Science*, pp. 448–456, Springer, Berlin, Germany, 1999.

[17] F. Min and W. Zhu, "Minimal cost attribute reduction through backtracking," in *Proceedings of the International Conference on Database Theory and Application*, vol. 258 of *FGIT-DTA/BSBT*, CCIS, 2011.

[18] F. Min and Q. Liu, "A hierarchical model for test-cost-sensitive decision systems," *Information Sciences*, vol. 179, no. 14, pp. 2442–2452, 2009.

[19] P. Turney, "Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm," *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 369–409, 1994.

[20] D. Margineantu, "Methods for cost-sensitive learning," 2001.

[21] S. Norton, "Generating better decision trees," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 1989.

[22] M. Núñez, "The use of background knowledge in decision tree induction," *Machine Learning*, vol. 6, no. 3, pp. 231–250, 1991.

[23] M. Tan, "Cost-sensitive learning of classification knowledge and its applications in robotics," *Machine Learning*, vol. 13, no. 1, pp. 7–33, 1993.

[24] N. Johnson and S. Kotz, *Continuous Distributions*, John Wiley, New York, NY, USA.

[25] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, vol. 4, Prentice Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1992.

[26] F. Min, W. Zhu, H. Zhao, G. Y. Pan, J. B. Liu, and Z. L. Xu, "Coser: cost-senstive rough sets," 2012, http://grc.fjzs.edu.cn/~fmin/ .

[27] Y. Y. Yao, "A partition model of granular computing," *Transactions on Rough Sets I*, vol. 3100, pp. 232–253, 2004.

[28] 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.

[29] T. Y. Lin, "Granular computing on binary relations-analysis of conflict and chinese wall security policy," in *Proceedings of Rough Sets and Current Trends in Computing*, vol. 2475 of *Lecture Notes in Artificial Intelligence*, 2002.

[30] T. Y. Lin, "Granular computing—structures, representations, and applications," in *Lecture Notes in Artificial Intelligence*, vol. 2639, 2003.

[31] L. Ma, "On some types of neighborhood-related covering rough sets," *International Journal of Approximate Reasoning*, vol. 53, no. 6, pp. 901–911, 2012.

[32] H. Zhao, F. Min, and W. Zhu, "Test-cost-sensitive attribute reduction based on neighborhood rough set," in *Proceedings of the IEEE International Conference on Granular Computing*, 2011.

[33] W. Zhu, "Generalized rough sets based on relations," *Information Sciences*, vol. 177, no. 22, pp. 4997–5011, 2007.

[34] W. Zhu and F.-Y. Wang, "Reduction and axiomization of covering generalized rough sets," *Information Sciences*, vol. 152, pp. 217–230, 2003.

[35] F. Min and W. Zhu, "Attribute reduction of data with error ranges and test costs," *Information Sciences*, vol. 211, pp. 48–67, 2012.

[36] Z. Zhou and X. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.

[37] H. Zhao, F. Min, and W. Zhu, "A backtracking approach to minimal cost feature selection of numerical data ," *Journal of Information & Computational Science*. In press.

[38] M. Kukar and I. Kononenko, "Cost-sensitive learning with neural networks," in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI '98)*, John Wiley & Sons, Chichester, UK, 1998.

[39] J. Lan, M. Hu, E. Patuwo, and G. Zhang, "An investigation of neural network classifiers with unequal misclassification costs and group sizes," *Decision Support Systems*, vol. 48, no. 4, pp. 582–591, 2010.

[40] P. Turney, "Types of cost in inductive concept learning," in *Proceedings of the ICML-2000 Workshop on Cost-Sensitive Learning*, 2000.

[41] S. Viaene and G. Dedene, "Cost-sensitive learning and decision making revisited," *European Journal of Operational Research*, vol. 166, no. 1, pp. 212–220, 2005.

[42] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[43] J. Błaszczyński, S. Greco, R. Słowiński, and M. Szeląg, "Monotonic variable consistency rough set approaches," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 979–999, 2009.

[44] Z. Bonikowski, E. Bryniarski, and U. Wybraniec-Skardowska, "Extensions and intentions in the rough set theory," *Information Sciences*, vol. 107, no. 1–4, pp. 149–167, 1998.

[45] M. Inuiguchi, Y. Yoshioka, and Y. Kusunoki, "Variable-precision dominance-based rough set approach and attribute reduction," *International Journal of Approximate Reasoning*, vol. 50, no. 8, pp. 1199–1214, 2009.

[46] Y. Kudo, T. Murai, and S. Akama, "A granularity-based framework of deduction, induction, and abduction," *International*

Journal of Approximate Reasoning, vol. 50, no. 8, pp. 1215–1226, 2009.

[47] J. A. Pomykała, "Approximation operations in approximation space," *Bulletin of the Polish Academy of Sciences: Mathematics*, vol. 35, no. 9-10, pp. 653–662, 1987.

[48] Y. Y. Yao, "Constructive and algebraic methods of the theory of rough sets," *Information Sciences*, vol. 109, no. 1–4, pp. 21–47, 1998.

[49] Y. Y. Yao, "Probabilistic rough set approximations," *Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 255–271, 2008.

[50] W. Zakowski, "Approximations in the space (u, π)," *Demonstratio Mathematica*, vol. 16, no. 40, pp. 761–769, 1983.

[51] W. Zhu, "Relationship among basic concepts in covering-based rough sets," *Information Sciences*, vol. 179, no. 14, pp. 2478–2486, 2009.

[52] W. Zhu and F. Wang, "On three types of covering-based rough sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1131–1144, 2007.

[53] S. Calegari and D. Ciucci, "Granular computing applied to ontologies," *International Journal of Approximate Reasoning*, vol. 51, no. 4, pp. 391–409, 2010.

[54] W. Zhu and F. Wang, "Covering based granular computing for conflict analysis," *Intelligence and Security Informatics*, pp. 566–571, 2006.

[55] Wikipedia, http://www.wikipedia.org/ .

[56] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic, Boston, Mass, USA, 1991.

[57] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131–156, 1997.

[58] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[59] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335–347, 1989.

[60] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998, http://www.ics.uci.edu/~mlearn/mlrepository.html.

[61] Q. H. Liu, F. Li, F. Min, M. Ye, and G. W. Yang, "An efficient reduction algorithm based on new conditional information entropy," *Control and Decision*, vol. 20, no. 8, pp. 878–882, 2005 (Chinese).

[62] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intelligent Decision Support*, 1992.

[63] G. Wang, "Attribute core of decision table," in *Proceedings of Rough Sets and Current Trends in Computing*, vol. 2475 of *Lecture Notes in Computer Science*, 2002.