International Journal of Innovative
Computing, Information and Control
Volume **13**, Number **6**, December **2017**

 ISSN 1349-4198
pp. **2099–2111**

AN EFFICIENT APPROACH OF TEST-COST-SENSITIVE ATTRIBUTE REDUCTION FOR NUMERICAL DATA

Shujiao Liao[1,2], Qingxin Zhu[1] and Rui Liang[1]

[1]School of Information and Software Engineering
University of Electronic Science and Technology of China
No. 4, Section 2, North Jianshe Road, Chengdu 610054, P. R. China
sjliao2011@163.com; qxzhu@uestc.edu.cn

[2]School of Mathematics and Statistics
Minnan Normal University
Zhangzhou 363000, P. R. China

Received April 2017; revised July 2017

Abstract. *In real applications of data mining, test costs often occur, and numerical data exists widely. In recent years, based on neighborhood rough set, some algorithms have been proposed to deal with the minimal-test-cost reduct problem of numerical data. Nevertheless, they do not perform very well on the computational efficiency. To overcome this disadvantage, we present a novel test-cost-sensitive attribute reduction approach in this paper. First, the properties of inconsistent neighborhoods are discussed for numerical decision systems. Then, a fast forward test-cost-sensitive attribute reduction algorithm is designed by using the obtained properties. The performance of the algorithm is tested with six UCI (University of California – Irvine) datasets. Experimental results show that the proposed algorithm is significantly more efficient than the existing algorithms. Moreover, the algorithm can achieve good results on minimizing the total test cost of data processing.*

Keywords: Test cost, Attribute reduction, Fast forward algorithm, Inconsistent neighborhood, Neighborhood rough set

1. **Introduction.** Cost-sensitive learning is one of key issues in data mining and machine learning communities [1, 2]. There are multiple types of cost in real applications [3], among which test cost is the time, money, or other resources paid for obtaining a data item of an object. In real world, data are usually not free to be acquired. For example, in a clinic system, a patient often needs to undertake a number of medical tests. With the test results, the doctor can diagnose whether the patient has a particular illness or disease. In this case, money and time consumed in performing these tests are the test costs.

Rough set theory, which was proposed by Pawlak [4], is a powerful mechanism to handle the uncertainty and the granulation of data. As an important task in rough sets, attribute reduction is a popular technique because it can select a suitable attribute subset, also called a reduct, to reduce the data dimensionality and meanwhile to keep the ability of original decision system [5, 6]. In last several years, by introducing cost-sensitive learning into the rough set theory, cost-sensitive attribute reduction has been studied [7, 8, 9], among which test-cost-sensitive attribute reduction is a branch of importance. In fact, when tests must be undertaken, attribute reduction is mandatory to decrease the total test cost. Based on the hierarchical models constructed in [10], minimal-test-cost reduct problems, which aim at finding a reduct to minimize the total test cost, have been explored to some extent [7, 11].

As is well known, numerical data exists widely in real applications, but numerical attributes are more complex than nominal ones and require more computational resources. Classical rough set and most of its extended models mainly address nominal data [12, 13]. To address this situation, neighborhood rough set was presented in [14, 15], which has been verified to be powerful in dealing with the attribute reduction of numerical data and hybrid data. Based on neighborhood rough set, a backtracking algorithm and a heuristic algorithm have been proposed to address the minimal-test-cost reduct problem of numerical data [16, 17]. However, both of them are not efficient enough [16]. Since the problem of finding a minimal-test-cost reduct is at least NP-hard [7], more efficient heuristic algorithms are needed.

To address the above-mentioned issue, in this paper we propose a new test-cost-sensitive attribute reduction approach by using the properties of inconsistent neighborhoods. Traditional neighborhood rough set models use neighborhoods to construct the theoretical and algorithmic framework. In [18, 19], a new concept called inconsistent neighborhood was defined for hybrid data, and relevant properties were explored. In this paper, inconsistent neighborhood is redefined for numerical data and its properties are discussed. A typical computational example is also given. It is known from the theoretical analysis that using inconsistent neighborhoods is often more efficient than using traditional neighborhoods for computing the quantities in neighborhood rough set. Based on the properties of inconsistent neighborhoods, a fast forward test-cost-sensitive attribute reduction algorithm is designed. To test the performance of the algorithm, experiments are carried out on six datasets from the UCI library [20]. Experimental results validate the effectiveness of the algorithm. In general the algorithm runs much faster than the existing neighborhood-rough-set-based test-cost-sensitive attribute reduction algorithms for numerical data. Moreover, it performs well on minimizing the total test cost consumed in data processing. In particular, the rational value of neighborhood radius δ is also discussed according to the experimental results.

The rest of the paper is organized as follows. Section 2 mainly discusses some key concepts and properties about test-cost-sensitive decision systems and inconsistent neighborhoods. Section 3 proposes the fast forward test-cost-sensitive attribute reduction algorithm and gives some evaluation metrics for the algorithm. Experiment results and the analyses are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Problem Statement.

This section discusses the basic knowledge of the paper. Firstly, the concept of test-cost-independent decision system is reviewed. Then, we give the notion and some crucial properties of inconsistent neighborhood with respect to numerical data. Finally, the minimal-test-cost reduct problem is introduced.

2.1. Test-cost-independent decision systems.

Decision system is a fundamental concept in data mining and machine learning.

Definition 2.1. [21] *A* decision system *(DS) S is the 5-tuple:*

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

where U is a finite 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 $I_a : U \to V_a$ is an information function for each $a \in C \cup D$.

In most applications, $D = \{d\}$, that is, we are given only one decision attribute called the class. If $|D| > 1$, we can construct $|D|$ decision systems, each having only one class.

In neighborhood rough set models, the attribute values of numerical conditional attributes are usually normalized to facilitate the data processing. An example of numerical

TABLE 1. An example of numerical decision system

Plant	sepal-length	sepal-width	petal-length	petal-width	class
x_1	0.24	0.64	0.10	0.05	setosa
x_2	0.18	0.41	0.10	0.05	setosa
x_3	0.12	0.50	0.07	0.05	setosa
x_4	0.76	0.45	0.60	0.52	versicolor
x_5	0.35	0.09	0.38	0.43	versicolor
x_6	0.65	0.32	0.52	0.52	versicolor
x_7	0.59	0.55	0.86	1.00	virginica
x_8	0.44	0.27	0.64	0.71	virginica
x_9	0.82	0.41	0.83	0.81	virginica

decision system is shown in Table 1, where $D = \{\text{class}\}$. In fact, Table 1 is a subtable of the normalized Iris dataset. For an attribute, the normalization of attribute values is executed by using the function $y = (x - \min)/(\max - \min)$, where x is the initial value, y is the normalized value, and min and max are the minimal value and the maximal value respectively.

Test costs often occur in data collecting. Test-cost-sensitive decision systems can be categorized into test-cost-independent decision systems and common-test-cost decision systems according to the relations of test costs between different attributes [10]. Here we only consider the former one.

Definition 2.2. *A* test-cost-independent decision system *(TCI-DS) S is the 6-tuple:*

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

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

An example of TCI-DS is given by Tables 1 and 2.

TABLE 2. An example of test cost vector

a	sepal length	sepal width	petal length	petal width
$c(a)$	\$3	\$5	\$4	\$6

As will be shown in Section 4, we generate different test cost settings for the same decision system in the experiments. Therefore, we can have as many test-cost-independent decision systems as we need.

2.2. **Some key properties of inconsistent neighborhoods.** In traditional models of neighborhood rough set [14, 15], there are a number of fundamental concepts, such as neighborhood, positive region, boundary region and reduct; and neighborhoods are used to describe other concepts. In [18, 19], a new concept called inconsistent neighborhood was introduced for hybrid data, and relevant properties were discussed thoroughly. Here we revise the definition of inconsistent neighborhood given in [18, 19] to make it be suitable for numerical decision systems.

Definition 2.3. *Let $S = (U, C, D, V, I)$ be a numerical decision system. Given $x_i \in U, B \subseteq C$ and $\delta > 0$, the inconsistent neighborhood of x_i with respect to attribute set B*

and neighborhood radius δ is defined as

$$in_B(x_i) = \{x_j \in U | \Delta_B(x_i, x_j) \leq \delta, D(x_j) \neq D(x_i)\}, \tag{1}$$

where Δ is a distance function.

Assuming that $x_1, x_2 \in U$, $C = (a_1, a_2, \ldots, a_n)$, and $v(x, a_i)$ denotes the value of object x on attribute a_i, then a frequently-used metric, named Minkowsky distance [22], is defined as

$$\Delta_p(x_1, x_2) = \left(\sum_{i=1}^{n} |v(x_1, a_i) - v(x_2, a_i)|^p \right)^{1/p}. \tag{2}$$

We use Euclidean distance Δ_2 in the paper.

Since numerical data can be seen as the special case of hybrid data, the properties of inconsistent neighborhoods with respect to hybrid data, which have been discussed in [18, 19], are also applicable for numerical decision systems. In the following, we show the key properties and obtain some new ones further.

Proposition 2.1. *Let $S = (U, C, D, V, I)$ be a numerical decision system. Given any $x, x_i, x_j \in U$ and $B \subseteq C$, we have*
(1) $in_\emptyset(x) = \{y \in U | D(y) \neq D(x)\}$;
(2) $x_j \in in_B(x_i) \Leftrightarrow x_i \in in_B(x_j)$.

Proposition 2.2. *Let $S = (U, C, D, V, I)$ be a numerical decision system, $B \subseteq C$. For any $X \subseteq U$, lower and upper approximations of X can be described as*

$$\begin{aligned} \underline{N_B}(X) &= \{x \in X | in_B(x) = \emptyset\}, \\ \overline{N_B}(X) &= X \cup \{x \notin X | in_B(x) \cap X \neq \emptyset\}. \end{aligned} \tag{3}$$

Proposition 2.3. *Let $S = (U, C, D, V, I)$ be a numerical decision system, $B \subseteq C$. The positive region and the boundary region can be described as*

$$\begin{aligned} POS_B(D) &= \{x \in U | in_B(x) = \emptyset\}, \\ BN_B(D) &= \{x \in U | in_B(x) \neq \emptyset\}. \end{aligned} \tag{4}$$

Proposition 2.4. *Let $S = (U, C, D, V, I)$ be a numerical decision system. Any $B \subseteq C$ is a decision-relative reduct if:*
(1) $\forall x \in POS_C(D)$, $in_B(x) = \emptyset$;
(2) $\forall a \in B$, $\exists x \in POS_C(D)$, s.t. $in_{B-\{a\}}(x) \neq \emptyset$.

In fact, $POS_C(D) = U$ in most cases, then we have

Corollary 2.1. *Let $S = (U, C, D, V, I)$ be a numerical decision system. Any $B \subseteq C$ is a decision-relative reduct if:*
(1) $\forall x \in U$, $in_B(x) = \emptyset$;
(2) $\forall a \in B$, $\exists x \in U$, s.t. $in_{B-\{a\}}(x) \neq \emptyset$.

Proposition 2.5. *(Type-1 monotonicity). Let $S = (U, C, D, V, I)$ be a numerical decision system, $B_1 \subseteq B_2 \subseteq C$. We have $\forall x \in U$, $in_{B_1}(x) \supseteq in_{B_2}(x)$.*

Proposition 2.6. *(Type-2 monotonicity). Let $S = (U, C, D, V, I)$ be a numerical decision system, $B \subseteq C$, $\delta_1 \leq \delta_2$. We have $\forall x \in U$, $in_1(x) \subseteq in_2(x)$.*

According to Proposition 2.5, the following corollary can be obtained.

Corollary 2.2. *Let $S = (U, C, D, V, I)$ be a numerical decision system, $B \subseteq C$. Assuming that $B_i \subseteq B$ ($1 \leq i \leq L$, L is a finite positive integer), then for any $x \in U$, we have*

$$in_B(x) \subseteq \bigcap_{i=1}^{L} in_{B_i}(x).$$

Proof: Based on Proposition 2.5, for any $x \in U$ and any B_i, we have $in_B(x) \subseteq in_{B_i}(x)$, so $in_B(x) \subseteq \bigcap_{i=1}^{L} in_{B_i}(x)$.

From Corollary 2.2, it is known that

$$in_B(x) \subseteq \bigcap_{a \in B} in_{\{a\}}(x). \tag{5}$$

An example is given to illustrate the concepts and properties discussed above.

Example 2.1. *A numerical decision system is indicated by Table 1, from which it is known that $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$ and $D = \{class\}$. For convenience, let $a_1 = sepal\text{-}length$, $a_2 = sepal\text{-}width$, $a_3 = petal\text{-}length$, and $a_4 = petal\text{-}width$. Let $\delta = 0.2$, then the inconsistent neighborhoods with respect to any attribute subset B can be computed. Some exemplary results are shown in Table 3, where B takes values listed as column headers. Then according to Corollary 2.1 and Table 3, it can be immediately known that $\{a_1, a_3\}$ is a reduct of the numerical decision system with $\delta = 0.2$.*

TABLE 3. Inconsistent neighborhoods of objects on some attribute subsets

x	$\{a_1\}$	$\{a_2\}$	$\{a_3\}$	$\{a_1, a_2\}$	$\{a_1, a_3\}$	$\{a_2, a_3\}$	$\{a_1, a_2, a_3\}$
x_1	$\{x_5, x_8\}$	$\{x_4, x_7\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_2	$\{x_5\}$	$\{x_4, x_6, x_7, x_8, x_9\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_3	$\emptyset$	$\{x_4, x_6, x_7, x_9\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_4	$\{x_7, x_9\}$	$\{x_1, x_2, x_3, x_7, x_8, x_9\}$	$\{x_8\}$	$\{x_7, x_9\}$	$\emptyset$	$\{x_8\}$	$\emptyset$
x_5	$\{x_1, x_2, x_8\}$	$\{x_8\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x_6	$\{x_7, x_9\}$	$\{x_2, x_3, x_8, x_9\}$	$\{x_8\}$	$\{x_9\}$	$\emptyset$	$\{x_8\}$	$\emptyset$
x_7	$\{x_4, x_6\}$	$\{x_1, x_2, x_3, x_4\}$	$\emptyset$	$\{x_4\}$	$\emptyset$	$\emptyset$	$\emptyset$
x_8	$\{x_1, x_5\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_4, x_6\}$	$\emptyset$	$\emptyset$	$\{x_4, x_6\}$	$\emptyset$
x_9	$\{x_4, x_6\}$	$\{x_2, x_3, x_4, x_6\}$	$\emptyset$	$\{x_4, x_6\}$	$\emptyset$	$\emptyset$	$\emptyset$

Besides, the universe U is divided into a set of equivalence classes by the decision attribute, $U/D = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}, \{x_7, x_8, x_9\}\}$. In other words, the objects are grouped into three subsets $X_1 = \{x_1, x_2, x_3\}$, $X_2 = \{x_4, x_5, x_6\}$ and $X_3 = \{x_7, x_8, x_9\}$. The lower and upper approximations with respect to the three object subsets can be computed by combining Table 3 with Proposition 2.2. Some results are depicted in Table 4.

According to Table 3 and Proposition 2.3, we can obtain the positive regions and the boundary regions on the exemplary attribute subsets, which are
$POS_{\{a_1\}}(D) = \{x_3\}$, $BN_{\{a_1\}}(D) = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8, x_9\}$;
$POS_{\{a_2\}}(D) = \emptyset$, $BN_{\{a_2\}}(D) = U$;
$POS_{\{a_3\}}(D) = POS_{\{a_2,a_3\}}(D) = \{x_1, x_2, x_3, x_5, x_7, x_9\}$,
$BN_{\{a_3\}}(D) = BN_{\{a_2,a_3\}}(D) = \{x_4, x_6, x_8\}$;
$POS_{\{a_1,a_2\}}(D) = \{x_1, x_2, x_3, x_5, x_8\}$, $BN_{\{a_1,a_2\}}(D) = \{x_4, x_6, x_7, x_9\}$;
$POS_{\{a_1,a_3\}}(D) = POS_{\{a_1,a_2,a_3\}}(D) = U$, $BN_{\{a_1,a_3\}}(D) = BN_{\{a_1,a_2,a_3\}}(D) = \emptyset$.

Comparing the concept and properties of inconsistent neighborhood with the counterparts of neighborhood in [14], it can be known that the introduction of inconsistent neighborhood provides some new solutions for computing the quantities in neighborhood

TABLE 4. Lower and upper approximations of object subsets on some attribute subsets, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$

	X	$\{a_1\}$	$\{a_2\}$	$\{a_3\}$	$\{a_1, a_2\}$	$\{a_1, a_2, a_3\}$
	X_1	$\{x_3\}$	$\emptyset$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3\}$
$\underline{N_B}(X)$	X_2	$\emptyset$	$\emptyset$	$\{x_5\}$	$\{x_5\}$	$\{x_4, x_5, x_6\}$
	X_3	$\emptyset$	$\emptyset$	$\{x_7, x_9\}$	$\{x_8\}$	$\{x_7, x_8, x_9\}$
	X_1	$\{x_1, x_2, x_3, x_5, x_8\}$	$U - \{x_5\}$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3\}$	$\{x_1, x_2, x_3\}$
$\overline{N_B}(X)$	X_2	$U - \{x_3\}$	U	$\{x_4, x_5, x_6, x_8\}$	$\{x_4, x_5, x_6, x_7, x_9\}$	$\{x_4, x_5, x_6\}$
	X_3	$U - \{x_2, x_3\}$	U	$\{x_4, x_6, x_7, x_8, x_9\}$	$\{x_4, x_6, x_7, x_8, x_9\}$	$\{x_7, x_8, x_9\}$

rough set (i.e., lower and upper approximations, positive region, boundary region, and reduct). In general, these solutions are more direct than the existing ones which use neighborhoods. For example, now the reducts can be captured immediately according to the situation of inconsistent neighborhoods, while in previous solutions they cannot be obtained until positive regions or related values such as dependency degrees have been computed. Furthermore, the inconsistent neighborhoods are often much narrower than neighborhoods. Therefore, the solutions employing inconsistent neighborhoods are usually more efficient than those employing neighborhoods.

2.3. **The minimal-test-cost reduct problem.** Traditional attribute reduction aims to find the minimal-length reduct, while in test-cost-sensitive attribute reduction, people are interested in the reduct with minimal total test cost. The relevant concept has been given in [7].

Definition 2.4. *Let $Red(S)$ denote the set of all reducts of a TCI-DS $S = (U, C, D, V, I, tc)$. Any $R \in Red(S)$ satisfying $tc(R) = \min\{tc(R')|R' \in Red(S)\}$ is called a* minimal-test-cost reduct.

Minimal-test-cost reducts are also called *optimal reducts* in test-cost-sensitive learning. The problem of finding such a reduct is called the minimal-test-cost reduct (MTR) problem. Generally speaking, exhaustive attribute reduction algorithms can always obtain the optimal reducts, while heuristic algorithms may get the reducts with sub-minimal total test cost.

3. **Algorithm Design and Evaluation Metrics.** In this section, we first propose a fast forward test-cost-sensitive attribute reduction (FTCAR) algorithm based on the properties of inconsistent neighborhoods, and then introduce some evaluation metrics for the algorithm.

3.1. **The fast forward test-cost-sensitive attribute reduction algorithm.** The proposed algorithm is described as follows.

Input: The TCI-DS $S = (U, C, D, V, I, tc)$, the neighborhood radius δ.
Output: The reduct R.
Step 1: Set $R = \emptyset$ and $S = U$, where S is the set of objects out of the positive region.
Step 2: Compute $in_\emptyset(x)$ for any $x \in U$.
Step 3: while $(S \neq \emptyset)$ do
  Step 3.1: for (each $a_i \in C - R$) do
    $IPR_i = \emptyset$; //IPR_i is the incremental positive region induced by a_i
    for (each $x_j \in S$) do
      $in_{R\cup\{a_i\}}(x_j) = \emptyset$;
      for (each $x_k \in in_R(x_j)$) do

```
                    if (Δ_{R∪{a_i}}(x_k, x_j) ≤ δ) then
                        in_{R∪{a_i}}(x_j) = in_{R∪{a_i}}(x_j) ⋃ {x_k};
                    endif
                endfor
                if (in_{R∪{a_i}}(x_j) = ∅) then
                    IPR_i = IPR_i ∪ {x_j};
                endif
            endfor
        endfor
    Step 3.2: Find a_l such that Sig_l = max_i(|IPR_i|/tc(a_i));
        if (Sig_l > 0) then
            R = R ∪ {a_l};
            S = S − IPR_l;
        else
            break;
        endif
    endwhile
Step 4: return R.
```

The FTCAR algorithm is essentially a heuristic algorithm, and there are four steps in the algorithm. Step 1 is to initialize some variables. Step 2 is to compute the initial inconsistent neighborhoods for each object in the universe. Step 3 is to select attributes into the reduct step by step, in which Step 3.1 is to calculate the incremental positive regions induced by unselected attributes, and Step 3.2 is to choose the attribute with the maximal significance. Step 4 is to return the obtained reduct.

In Step 3, the key step of the algorithm, the attributes are added into the reduct R one by one according to the attribute significances until none of the significances is larger than 0 or no object is outside the positive region. For each unselected attribute $a_i \in C - R$, its significance is evaluated by $|IPR_i|/tc(a_i)$, namely the ratio of the size of the incremental positive region induced by a_i to the test cost of a_i, in which the incremental positive region IPR_i is obtained by using the inconsistent neighborhoods.

As shown in Step 3, there are mainly two techniques to improve the efficiency of the algorithm. One is that, since $in_{R\cup\{a_i\}}(x) \subseteq in_R(x)$ according to Proposition 2.5, it is only required to judge whether the objects in $in_R(x)$, instead of all objects in U, belong to $in_{R\cup\{a_i\}}(x)$, and meanwhile $in_R(x)$ will get smaller and smaller as the attribute reduction proceeds. The other is that, one only needs to compute the inconsistent neighborhoods of the objects out of the positive region, while by using $S = S - IPR_l$, this kind of objects will get fewer and fewer with the attribute reduction going on. In general, the computation complexity will be reduced gradually at the sequential rounds of the while-loop, and the process of attribute reduction will be accelerated greatly.

3.2. **Some evaluation metrics.** Some metrics have been introduced in [7] to evaluate the performance of a heuristic test-cost-sensitive attribute reduction algorithm by comparing with the results obtained by the corresponding exhaustive algorithm. The metrics are mainly finding optimal factor and average exceeding factor.

Suppose that the number of experiments executed by a heuristic algorithm is K, and the number of successful searching an optimal reduct is k. The *finding optimal factor* (FOF) is formulated as

$$FOF = \frac{k}{K}. \tag{6}$$

Obviously, the higher the FOF is, the better the heuristic algorithm performs on finding the optimal reducts.

For a dataset with a particular test cost setting, let R' be an optimal reduct. The exceeding factor of a reduct R is

$$ef(R) = \frac{tc(R) - tc(R')}{tc(R')}. \tag{7}$$

Suppose that the number of experiments run by a heuristic algorithm is K, and let R_i denote the reduct obtained by the algorithm in the i-th experiment ($1 \leq i \leq K$). The *average exceeding factor* (AEF) is defined as

$$AEF = \frac{\sum_{i=1}^{K} ef(R_i)}{K}. \tag{8}$$

Naturally, the smaller the AEF is, the better the heuristic algorithm performs on minimizing the total test cost.

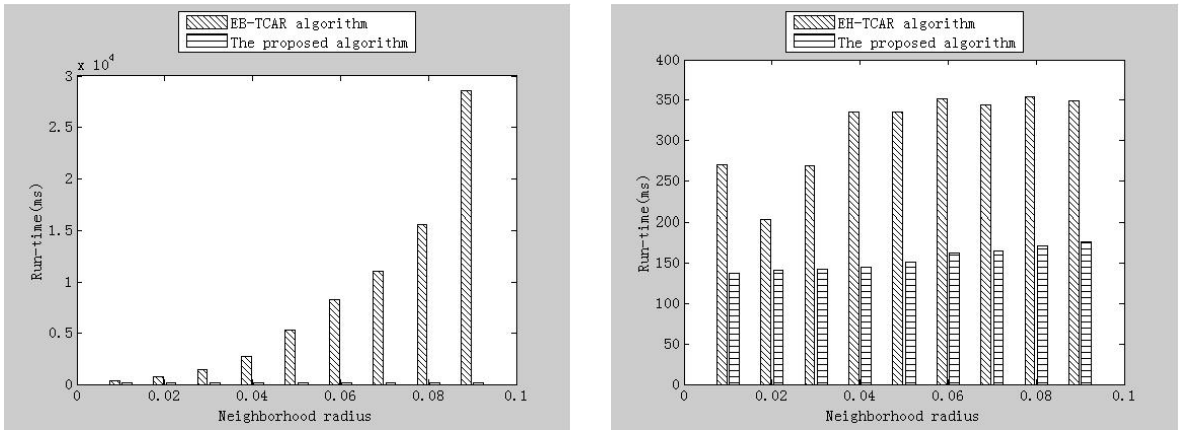
4. **Experiments.** In this section, two kinds of metrics are used to evaluate the performance of the proposed fast forward test-cost-sensitive attribute reduction algorithm by experimentation. One is the run-time compared with the two existing test-cost-sensitive attribute reduction algorithms presented in [16, 17]. The other is the ability of minimizing the total test cost, which is measured by using metrics *finding optimal factor* and *average exceeding factor* introduced in Section 3.2.

Six datasets from the UCI library are employed in the experiments. The basic information of the datasets is listed in Table 5, where $|U|$ is the number of objects, $|C|$ is the number of condition attributes, and D is the name of the decision. To make the data easier to handle, data items of condition attributes are normalized onto $[0, 1]$, and missing values are directly set to be 0.5. Because most datasets from the UCI library have no intrinsic test costs, we generate them for experimentation. For each dataset in Table 5, the test costs are set to be uniformly distributed random integers ranging from 1 to 100. Furthermore, there is a remark about the range of neighborhood radius δ. That is, for each dataset, the range of the neighborhood radius is selected according to the characteristic of the dataset. If the neighborhood radius is larger than the chosen range, the reduct obtained by the backtracking algorithm in [16] may be an empty set. Hence, the neighborhood radiuses are not necessarily the same between different datasets.

Table 5. Dataset information

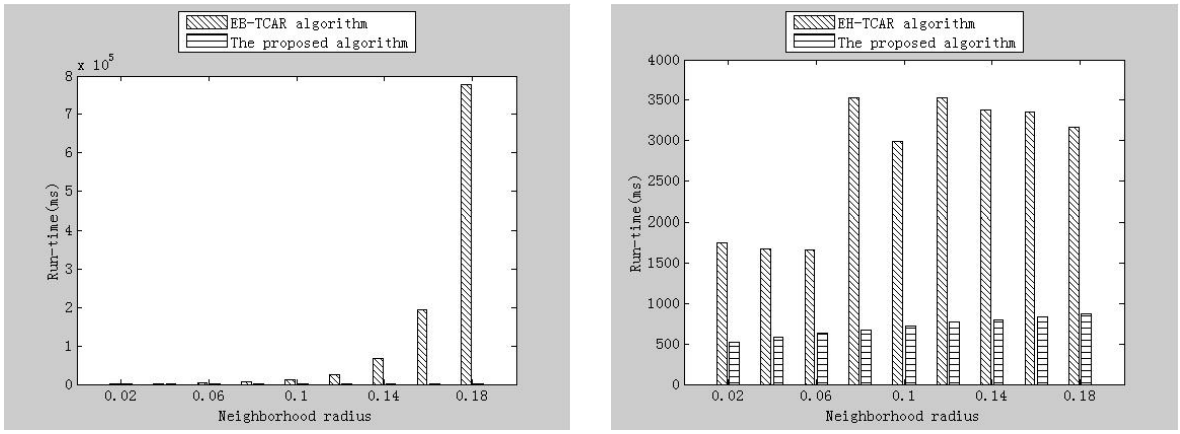
No.	Name	Domain	$\|U\|$	$\|C\|$	D
1	Image	graphics	210	19	class
2	Iono	physics	351	34	class
3	Sonar	physics	208	60	class
4	Wine	agriculture	178	13	class
5	Wdbc	clinic	569	30	diagnosis
6	Wpbc	clinic	198	33	outcome

For each dataset, we run the three algorithms for 1000 times with different test cost settings. The values of the above-mentioned metrics are computed accordingly. Firstly, we show the average run-time of each algorithm on each dataset in Figures 1-6, where "EB-TCAR" and "EH-TCAR" denote the existing backtracking test-cost-sensitive attribute reduction algorithm in [16] and the existing heuristic test-cost-sensitive attribute reduction algorithm in [17], respectively. From the figures, it can be found that the proposed fast



(a) (b)

FIGURE 1. Comparisons of average run-time on Image dataset, where "EB-TCAR" and "EH-TCAR" denote the existing backtracking test-cost-sensitive attribute reduction algorithm in [16] and the existing heuristic test-cost-sensitive attribute reduction algorithm in [17], respectively



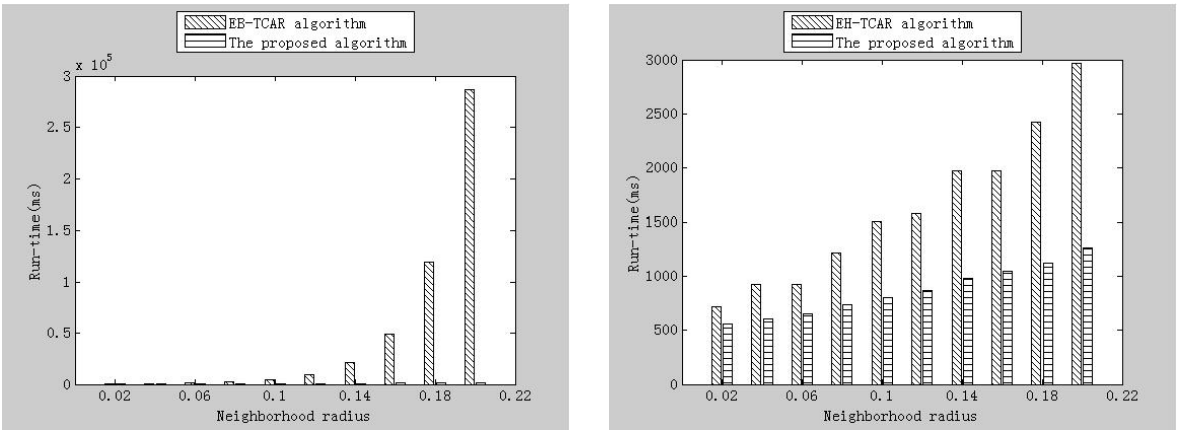
(a) (b)

FIGURE 2. Comparisons of average run-time on Iono dataset

forward algorithm runs much more quickly than the two existing algorithms in most cases. Moreover, although the run-time of the proposed algorithm usually grows with the increase of the neighborhood radius, in general the growth is slow. It means that the algorithm is relatively stable in terms of the computational efficiency.

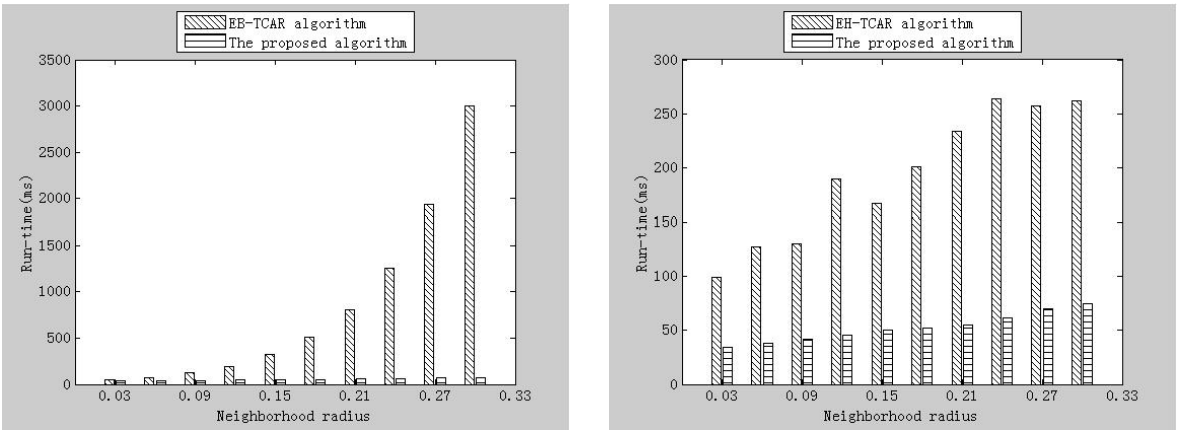
Then, the values of finding optimal factor and average exceeding factor are depicted in Figures 7-9. Note that, the two metrics are computed by comparing the results between the proposed algorithm and the algorithm in [16]. The reason is that, the latter is a backtracking algorithm whose results are optimal reducts. From the figures, it is known that although the finding optimal factors are not very high in some cases, most of the average exceeding factors are not more than 0.2. Hence, the results are satisfactory. The proposed algorithm can achieve good results on minimizing the total test cost.

Finally, we explore whether there is an optimal setting of neighborhood radius δ among different datasets. On one hand, from Figures 1-6, it is found that the smaller δ is, the more quickly the proposed algorithm runs. The reason is that, according to Proposition 2.6, the inconsistent neighborhoods, which play a crucial role in the attribute reduction,



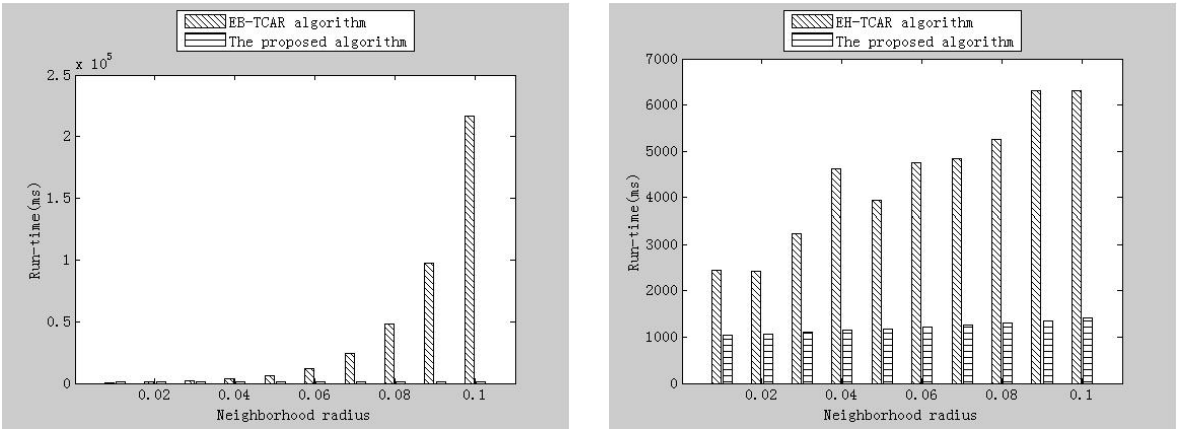
(a) (b)

FIGURE 3. Comparisons of average run-time on Sonar dataset



(a) (b)

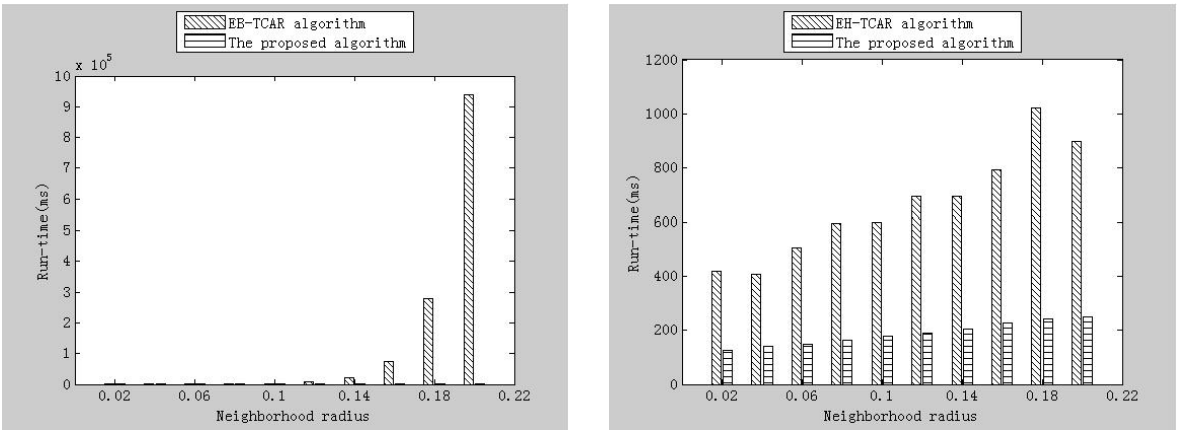
FIGURE 4. Comparisons of average run-time on Wine dataset



(a) (b)

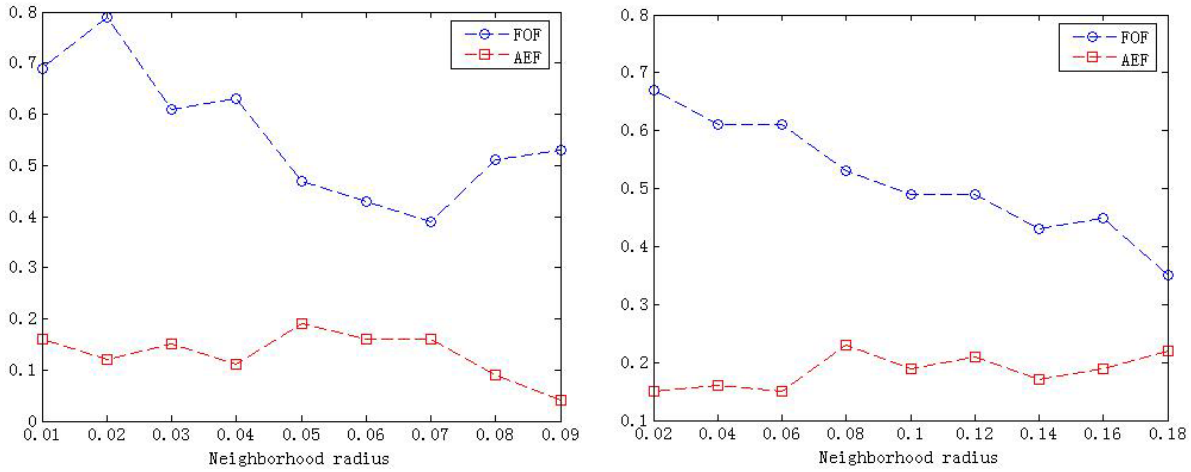
FIGURE 5. Comparisons of average run-time on Wdbc dataset

expand with the increase of δ. On the other hand, from Figures 7-9, it is known that there is not a universally optimal value of δ. Even if for the same dataset, the largest



(a) (b)

FIGURE 6. Comparisons of average run-time on Wpbc dataset



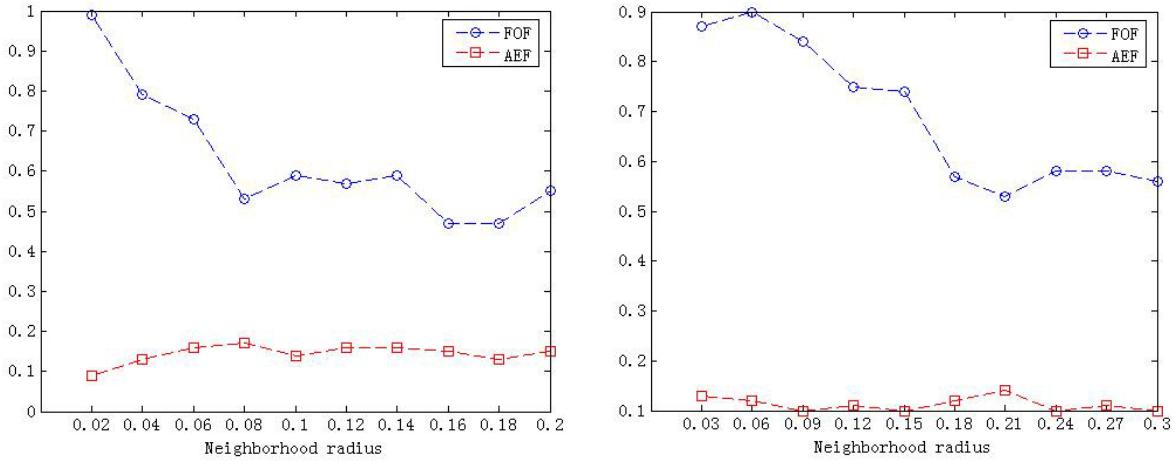
(a) (b)

FIGURE 7. Finding optimal factors (FOF) and average exceeding factors (AEF) on datasets: (a) Image, (b) Iono

finding optimal factor and the smallest average exceeding factor may be achieved at different δ. Since in general average exceeding factors do not change greatly, we mainly consider the setting of δ from the viewpoint of finding optimal factors, and conclude that $\delta = 0.02$ might be a rational setting if only a single value instead of a value range is given for δ (Note that, in fact we also run the three algorithms on Wine dataset with node $\delta = 0.02$. Although the results are not displayed in the paper, they can support the inference effectively).

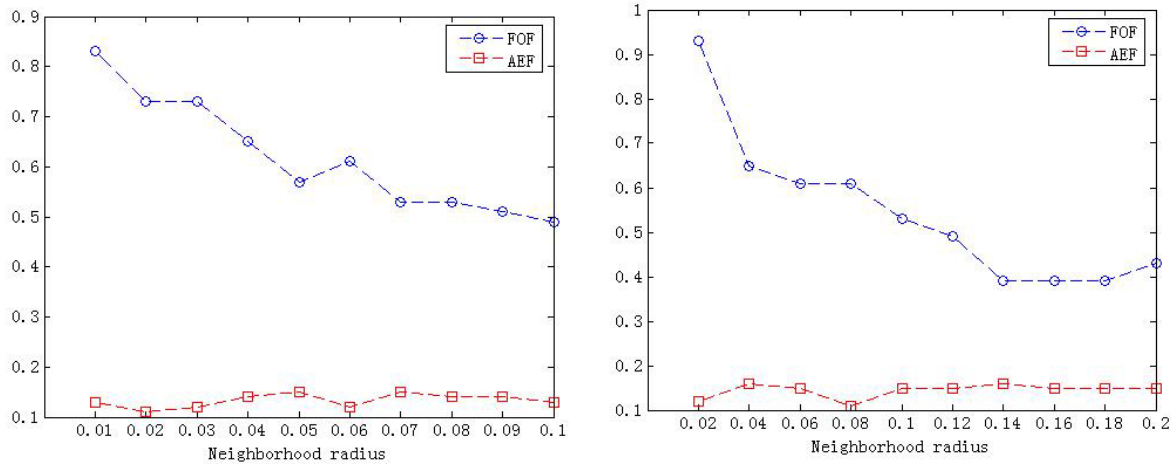
To sum up, the proposed fast forward test-cost-sensitive attribute reduction algorithm performs well not only on the computational efficiency but also on the effectiveness in terms of total test cost minimization.

5. **Conclusions.** Test-cost-sensitive attribute reduction is an important issue in cost-sensitive learning. In recent years, some algorithms have been presented to solve the minimal-test-cost reduct problem of numerical data based on neighborhood rough set. Unfortunately, these algorithms are not efficient enough. To overcome this difficulty, in this paper an efficient approach is proposed by using the properties of inconsistent



(a) (b)

FIGURE 8. Finding optimal factors and average exceeding factors on datasets: (a) Sonar, (b) Wine



(a) (b)

FIGURE 9. Finding optimal factors and average exceeding factors on datasets: (a) Wdbc, (b) Wpbc

neighborhoods. Experimental results demonstrate that the designed algorithm is both efficient and effective.

In the future, we will study the test-cost-sensitive attribute reduction for more complex data, such as hybrid data, or under more complicated environments, such as the case where the total test cost has an upper bound.

Acknowledgement. This work is supported in part by the National Natural Science Foundation of China under Grant No. 61379021, the Natural Science Foundation of Fujian Province, China under Grant No. 2017J01771, and the Education Department of Fujian Province under Grant No. JAT160291.

REFERENCES

[1] D. H. Guan, W. W. Yuan, T. H. Ma et al., Cost-sensitive elimination of mislabeled training data, *Information Sciences*, vol.402, pp.170-181, 2017.

[2] Y. H. Zhou and Z. H. Zhou, Large margin distirbution learning with cost interval and unlabeled data, *IEEE Trans. Knowl. Data Eng.*, vol.28, no.7, pp.1749-1763, 2016.
[3] P. D. Turney, Types of cost in inductive concept learning, *Proc. of the Workshop on Cost-Sensitive Learning at the 17th ICML*, pp.1-7, 2000.
[4] Z. Pawlak, Rough sets, *International Journal of Computer and Information Sciences*, vol.11, pp.341-356, 1982.
[5] J. Qian, P. Lv, X. D. Yue et al., Hierarchical attribute reduction algorithms for big data using MapReduce, *Knowledge-Based Systems*, vol.73, pp.18-31, 2015.
[6] Z. Q. Meng and Z. Z. Shi, On quick attribute reduction in decision-theoretic rough set models, *Information Sciences*, vol.330, pp.226-244, 2016.
[7] F. Min, H. P. He, Y. H. Qian and W. Zhu, Test-cost-sensitive attribute reduction, *Information Sciences*, vol.181, no.22, pp.4928-4942, 2011.
[8] S. J. Liao, Q. X. Zhu and F. Min, Cost-sensitive attribute reduction in decision-theoretic rough set models, *Mathematical Problems in Engineering*, Article ID 875918, vol.2014, 2014.
[9] A. H. Tan, W. Z. Wu and Y. Z. Tao, A set-cover-based approach for the test-cost-sensitive attribute reduction problem, *Soft Computing*, doi:10.1007/s00500-016-2173-3, 2016.
[10] F. Min and Q. H. Liu, A hierarchical model for test-cost-sensitive decision systems, *Information Sciences*, vol.179, no.14, pp.2442-2452, 2009.
[11] J. B. Liu, S. J. Liao, F. Min et al., An improved genetic algorithm to minimal test cost reduction, *IEEE International Conference on Granular Computing*, Hangzhou, China, pp.304-309, 2012.
[12] T. Y. Lin and R. S. Yu, Unifying variable precision and classical rough sets: Granular approach, *Intelligent Systems Reference Library*, vol.43, pp.365-373, 2013.
[13] R. L. Pan, Z. C. Zhang, Y. L. Fan et al., Multi-objective optimization method for learning thresholds in a decision-theoretic rough set model, *International Journal of Approximate Reasoning*, vol.71, pp.34-49, 2016.
[14] Q. H. Hu, D. R. Yu, J. F. Liu and C. X. Wu, Neighborhood rough set based heterogeneous feature subset selection, *Information Sciences*, vol.178, no.18, pp.3577-3594, 2008.
[15] Q. H. Hu, W. Pedrycz, D. R. Yu and J. Lang, Selecting discrete and continuous features based on neighborhood decision error minimization, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol.40, no.1, pp.137-150, 2010.
[16] S. J. Liao, J. B. 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.
[17] H. Zhao, F. Min and W. Zhu, Test-cost-sensitive attribute reduction based on neighborhood rough set, *Proc. of the 2011 IEEE International Conference on Granular Computing*, Kaohsiung, Taiwan, pp.802-806, 2011.
[18] S. J. Liao, Q. X. Zhu, R. Liang et al., Inconsistent neighborhoods and relevant properties in neighborhood rough set models, *Proc. of International Conference on Computer Engineering and Information Systems*, Shanghai, China, pp.149-152, 2016.
[19] S. J. Liao, Q. X. Zhu and R. Liang, On the properties and applications of inconsistent neighborhood in neighborhood rough set models, *IEICE Trans. Information and Systems*, 2017.
[20] *https://archive.ics.uci.edu/ml/datasets.html.*
[21] Y. Y. Yao, A partition model of granular computing, *Lecture Notes in Computer Science*, vol.3100, pp.232-253, 2004.
[22] D. R. Wilson and T. R. Martinez, Improved heterogeneous distance functions, *Journal of Artificial Intelligence Research*, vol.6, no.1, pp.1-34, 1997.