Knowledge-Based Systems 21 (2008) 753–763

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A comparative study on rough set based class imbalance learning

Jinfu Liu *, Qinghua Hu, Daren Yu

Harbin Institute of Technology, Harbin 150001, China

ARTICLE INFO

Article history:
Received 12 August 2007
Accepted 28 March 2008
Available online 4 April 2008

Keywords:
Rough sets
Class imbalance learning
Sample weighting

ABSTRACT

This paper performs systematic comparative studies on rough set based class imbalance learning. We compare the strategies of weighting, re-sampling and filtering used in the rough set based methods for class imbalance learning. Weighting is better than re-sampling, and re-sampling is better than filtering. The weighted rough set based method achieves the best performance in class imbalance learning. Furthermore, we compare various configurations of the weighted rough set based method. The weighted rule extraction and weighted decision have greater influence on the performance of the weighted rough set based method than the weighted attribute reduction. The weighted attribute reduction based on the weighted degree of dependency, the rule extraction for the exhaustive set of rules and the weighted decision based on the majority voting of the factor of weighted strength are the optimal configurations for class imbalance learning. Finally, we compare the weighted rough set based method with the decision tree and SVM based methods. The experimental results show that the weighted rough set based method outperforms the decision tree and SVM based methods. It can be concluded from the comparisons that the weighted rough set based method is effective for class imbalance learning.

1. Introduction

The class imbalance problem is recognized as a crucial problem in machine learning and data mining because such a problem is encountered in a large number of domains, such as fraud detection [11], medical diagnosis [23] and text classification [43]. It usually causes serious negative effects on the performance of a learning method that assumes a balanced class distribution [56–58]. Much work has been done to deal with the class imbalance problem [21,23,29,40]. A recognized solution to the class imbalance problem is to take into account the a priori knowledge of class distribution at the data or algorithmic level [6,22]. At the data level, re-sampling training data is a popular solution to the class imbalance problem, and it over-samples the minority class or under-samples the majority class to balance the class distribution of a data set. A traditional learning method can be directly employed to deal with the class imbalance problem by learning from the re-sampled data set [1,9,59]. At the algorithmic level, weighting training data is a popular solution to the class imbalance problem, and it assigns a larger weight to the minority class than to the majority class to balance the class distribution of a data set. A traditional learning method must be modified to make use of the weights when this strategy is used. Compared to re-sampling, weighting can usually be used to achieve better performance [21,23]. Many researchers have employed this strategy to improve the performance of decision tree [9,51] and SVM [5,8,50] in class imbalance learning.

Rough set theory is a powerful mathematical tool proposed by Pawlak [33,34] for dealing with inexact, uncertain or vague information, and a large number of studies have been directed to its development and applications [2,3,14,25,26,30,45,60]. The main advantage of rough set theory in data analysis is that it does not need any preliminary or additional information about data like probability distributions in statistics, basic probability assignments in Dempster–Shafer theory, or a grade of membership in fuzzy set theory [36–39]. In the traditional rough set based method, all samples are considered equally important, and the same probability $1/n$, where n is the size of a training data set, is assigned to each sample for computing the accuracy of approximation, reducing attributes and extracting decision rules. When the class distribution of a data set is skewed, the traditional rough set based method is biased towards the majority class and usually has a poor prediction for the minority class because the a priori knowledge of class distribution is not taken into account.

Re-sampling training data can be used in the traditional rough set based method to perform class imbalance learning at the data level. At the algorithmic level, Stefanowski and Wilk [49] introduced removing and filtering techniques to process inconsistent samples from the majority class in boundary regions. Their experimental results indicated that the removing and filtering techniques could be used to improve the performance of a rough set based method in class imbalance learning, and the filtering technique performed better than the removing technique. Liu et al.

* Corresponding author. Tel.: +86 451 86413241 252.
E-mail address: liujinfu@hcms.hit.edu.cn (J. Liu).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.03.031

[28] introduced weights to represent the a priori knowledge of class distribution and proposed a weighted rough set based method to perform class imbalance learning. Their experimental results showed that the weighted rough set based method achieved better performance than the traditional rough set based method in class imbalance learning.

To the best of our knowledge, there are no systematic comparative researches on rough set based class imbalance learning so far. In this study, we first compare the strategies of weighting, re-sampling and filtering used in the rough set based methods for class imbalance learning. We find that the weighted rough set based method outperforms the methods based on re-sampling and filtering. Secondly, we compare various configurations of the weighted rough set based method, and optimize the configurations for class imbalance learning. Finally, we compare the weighted rough set based method with the decision tree and SVM based methods used for class imbalance learning, and find that the weighted rough set based method outperforms the comparative methods.

The remainder of this paper is organized as follows. Section 2 describes preliminary notions related to rough sets. Section 3 reviews various configurations of learning and classification based on rough sets. Section 4 discusses rough set based methods for class imbalance learning. Section 5 discusses other methods for class imbalance learning. Section 6 presents systematic comparative studies on rough set based class imbalance learning. Finally, Section 7 concludes this work.

2. Preliminary notions related to rough sets

$IS = \langle U, A, V, f\rangle$ is called an information system, where $U = \{x_1, \ldots, x_i, \ldots, x_n\}$ is a set of samples, $A = \{a_1, \ldots, a_j, \ldots, a_m\}$ is a set of attributes, V is the value domain of A, and f: $U \times A \to V$ is an information function.

Let $B \subseteq A$. B induces an equivalence (indiscernibility) relation on U as shown below

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

The family of all equivalence classes of $IND(B)$, i.e., the partition induced by B, is denoted as

$$\Pi_B = U/B = \{[x_i]_B : x_i \in U\}, \quad (2)$$

where $[x_i]_B$ is the equivalence class containing x_i. All the elements in $[x_i]_B$ are equivalent (indiscernible) with respect to B. Equivalence classes are elementary sets in rough set theory.

Let $B \subseteq A$ and $X \subseteq U$. The lower and upper approximations of X with respect to B, denoted by $\underline{B}X$ and $\overline{B}X$, respectively, are defined as

$$\begin{cases} \underline{B}X = \cup\{[x_i]_B \mid [x_i]_B \subseteq X\} \\ \overline{B}X = \cup\{[x_i]_B \mid [x_i]_B \cap X \neq \varnothing\} \end{cases}. \quad (3)$$

Lower approximation $\underline{B}X$ is the greatest union of equivalence classes contained in X, and it is the set of all samples that can be certainly classified as belonging to X using B. Upper approximation $\overline{B}X$ is the least union of equivalence classes containing X and it is the set of all samples that can be possibly classified as belonging to X using B. $BN_B(X) = \overline{B}X - \underline{B}X$ is called the boundary region of X with respect to B. X is definable with respect to B if $BN_B(X) = \varnothing$, otherwise X is rough with respect to B. In contrast to a definable set, any rough set has a non-empty boundary region.

A rough set can be characterized using the accuracy of approximation as defined below

$$\alpha_B(X) = \mid \underline{B}X \mid / \mid \overline{B}X \mid, \quad (4)$$

where $|\bullet|$ denotes the cardinality of a set. X is definable with respect to B if $\alpha_B(X) = 1$, otherwise X is rough with respect to B.

Let $B \subseteq A$ and $a \in B$. a is redundant in B if $U/B = U/(B - a)$, otherwise a is indispensable in B. B is independent if every $a \in B$ is indispensable in B. B is a reduct of A if B is independent and $U/B = U/A$.

A reduct is a minimal subset of attributes that preserves the indiscernibility relation determined by full attributes. There is usually more than one reduct for a given information system, and the intersection of all the reducts is called the core. The core is the most important subset of attributes, since none of its elements can be removed without changing the indiscernibility relation.

$IS = \langle U, A, V, f\rangle$ is called a decision table if $A = C \cup D$, where C is the condition attribute set and D is the decision attribute set. The degree of dependency of D on C can be defined as

$$\gamma_C(D) = \mid POS_C(D) \mid / \mid U \mid, \quad (5)$$

where $POS_C(D) = \cup_{X \in U/D} \underline{C}X$ is called the positive region of the partition U/D with respect to C, and it is the set of all samples that can be certainly classified as belonging to blocks of U/D using C.

D depends totally on C if $\gamma_C(D) = 1$, otherwise D depends partially on C. $\gamma_C(D)$ can be used as a significance measure of C with respect to D.

Let $IS = \langle U, A = C \cup D, V, f\rangle$ be a given decision table, $B \subseteq C$ and $a \in B$. a is redundant in B with respect to D if $\gamma_{B-a}(D) = \gamma_B(D)$, otherwise a is indispensable in B with respect to D. B is independent with respect to D if every $a \in B$ is indispensable in B with respect to D. B is a D-relative reduct of C if B is independent with respect to D and $\gamma_B(D) = \gamma_C(D)$.

3. Various configurations of learning and classification based on rough sets

3.1. Two popular significance measures of attributes

Attribute reduction is an important problem which can be solved using rough sets. A number of algorithms have been proposed to perform attribute reduction [3,25,26,30,45]. Finding all reducts is a *NP*-hard problem. However, it is usually adequate enough for most real-world applications to find one of the reducts. Based on the significance of an attribute, a heuristic attribute reduction algorithm can be designed to find a reduct.

A straightforward way to measure the significance of an attribute is based on the degree of dependency. When an attribute is added to the condition attribute set, the degree of dependency of decision attributes on condition attributes usually changes, and the change can be defined as the significance of an attribute.

Let $IS = \langle U, A = C \cup D, V, f\rangle$ be a given decision table and $B \subseteq C$. Based on the degree of dependency, the significance of $a \in C - B$ on the basis of B with respect to D is defined as

$$SIG_\gamma(a, B, D) = \gamma_{B\cup\{a\}}(D) - \gamma_B(D). \quad (6)$$

Another popular way to measure the significance of an attribute is based on Shannon's entropy. In the framework of rough set methodology, attributes are considered knowledge describing samples. Shannon's entropy is a powerful tool for measuring the uncertainty of knowledge [10,46,55].

Based on Shannon's entropy, the significance of $a \in C - B$ on the basis of B with respect to D is defined as

$$SIG_H(a, B, D) = H(D \mid B) - H(D \mid B \cup \{a\}) \quad (7)$$

where $H(D|B)$ denotes the conditional entropy of D given B. Suppose that $\Pi_B = \{X_1, \ldots, X_i, \ldots, X_n\}$ are the partition induced by B, and $\Pi_D = \{Y_1, \ldots, Y_j, \ldots, Y_m\}$ are the partition induced by D. $H(D \mid B) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(X_i, Y_j) \log p(Y_j \mid X_i)$.

Based on the significance of an attribute, a heuristic attribute reduction algorithm can be designed to find a reduct by iteratively selecting an attribute with the maximum significance.

3.2. Two popular rule extraction algorithms

Nowadays, there are many known rule extraction algorithms inspired by rough set theory [17,20,24,31,44,47,52,53], and LEM2 algorithm proposed by Grzymala [17] is one of the most widely used algorithms for real-world applications.

Some preliminary descriptions about LEM2 algorithm can be given as shown below.

A family of generalized decisions, denoted by $\widetilde{D}$, is first defined on a given decision table. Each element of $\widetilde{D}$ is a single or joint decision. According to $\widetilde{D}$, all the samples in the decision table is then partitioned into a family of disjoint subsets, denoted by $\widetilde{Y}$. Each element of $\widetilde{Y}$ is the lower approximation of a decision class which corresponds to a single decision of $\widetilde{D}$, or one of the disjoint subsets of the boundary region of a decision class which corresponds to a joint decision of $\widetilde{D}$. Suppose that there are three decision classes Y_1, Y_2 and Y_3. The boundary region of Y_1 consists of three disjoint subsets: $BND(Y_1) = (\overline{B}Y_1 \cap \overline{B}Y_2 - \overline{B}Y_3) \cup (\overline{B}Y_1 \cap \overline{B}Y_3 - \overline{B}Y_2) \cup (\overline{B}Y_1 \cap \overline{B}Y_2 \cap \overline{B}Y_3)$. It is clear that $\widetilde{Y}$ is consistent with $\widetilde{D}$. Finally, for each $K \in \widetilde{Y}$, LEM2 employs a heuristic strategy to extract a minimal set of rules.

Let $IS = \langle U, A = C \cup D, V, f\rangle$ be a given decision table, $\widetilde{D}_K$ be a generalized decision, K be the subset of samples corresponding to $\widetilde{D}_K$, c be an elementary condition that has an expression (a, v), where $a \in C$ and $v \in V_a$, $\Phi = c_1 \wedge \cdots \wedge c_j \wedge \cdots \wedge c_q$ be the conjunction of q elementary conditions, $[\Phi]$ be the cover of Φ, i.e., the subset of samples that satisfy all the elementary conditions of Φ, $[\Phi]_K^+ = [\Phi] \cap K$ be the positive cover of Φ on K, and $[\Phi]_K^- = [\Phi] \cap (U - K)$ be the negative cover of Φ on K. Then a rule, denoted by r, is described as

$$\text{if } \Phi \quad \text{then} \quad \widetilde{D}_K, \tag{8}$$

where Φ is called the condition part of r, satisfying $[\Phi]_K^+ \neq \varnothing$, and $\widetilde{D}_K$ is called the decision part of r. If $\widetilde{D}_K$ is a single decision, r is called a certain rule. Otherwise, if $\widetilde{D}_K$ is a joint decision, r is called a possible rule.

Based on the preliminary descriptions above, LEM2 algorithm can be described as Algorithm 1. Extracting a rule is essentially finding the elementary conditions of the rule. From Algorithm 1, we find that LEM2 employs $|[c] \cap G|$ as the heuristic search strategy to iteratively find the elementary conditions of a rule. LEM2 extracts a minimal set of rules to cover samples, and the minimal set of rules contains the smallest number of strong rules.

Algorithm 1. [LEM2 algorithm]

Input: A subset of samples $K \in \tilde{Y}$.
Output: A minimum set of rules$R$.

```
1. begin
2.   G ← K, R ← ∅;
3.   while G ≠ ∅ do
4.   begin
5.     Φ ← ∅;
6.     Φ_G ← {c:[c] ∩ G≠∅};
7.     while (Φ = ∅) or (not([Φ ] ⊆ K)) do
8.     begin
9.       select c ∈ Φ_G such that |[c] ∩ G| is maximum. If ties
         occur then select c with the smallest |[c]|. If further
         ties occur then select the first c from the list;
         Φ ← Φ ∪ {c};
         G ← [c] ∩ G;
         Φ_G ← {c:[c] ∩ G≠∅};
         Φ_G ← Φ_G − Φ;
10.    end
11.    for each c ∈ Φ do
12.      if [Φ − {c}] ⊆ K then
13.        Φ ← Φ − {c};
14.  create a rule r based on Φ;
15.  R ← R ∪ {r};
16.  G ← K − ∪_{r∈R}[r];
17. end
18. for each r ∈ R do
19.      if ∪_{s∈R−{r}}[S] = K then
20.        R ← R − {r};
21. end
```

Another rule extraction strategy is extracting the exhaustive set of rules. Compared to the minimum set of rules, the exhaustive set of rules provide the richest information about patterns existing in a given decision table. Researches show that it is useful for some classification and discovery applications [16]. However, it is the most demanding from the viewpoint of time and memory complexity. It has a larger number of rules than the minimum set of rules, and besides strong rules, it has weak and very specific rules. A detailed rule extraction algorithm for the exhaustive set of rules can be found in [48].

3.3. Two popular decision algorithms

Pawlak introduced three factors to evaluate extracted rules, and they are strength, certainty and cover [35].

Let r be a given rule, $\widetilde{D}_K = \{d_1, \ldots, d_j, \ldots, d_n\}$ be the decision part of r, $[r]$ be the cover of r, $[\widetilde{D}_K]$ be the cover of $\widetilde{D}_K$ and $[r]_d^+ = [r] \cap [d]$ be the positive cover of r on d, where $d \in \widetilde{D}_K$. The factor of strength of r is defined as

$$\mu_{\text{str}}(r) = |\,[r]\,| \,/\, |\,U\,|, \tag{9}$$

the factor of cover of r is defined as

$$\mu_{\text{cov}}(r) = |\,[r]\,| \,/\, |\,[\widetilde{D}_K]\,|, \tag{10}$$

and the factor of certainty of r to d is defined as

$$\mu_{\text{cer}}(r, d) = |\,[r]_d^+\,| \,/\, |\,[r]\,|\,. \tag{11}$$

Extracted rules can be used to predict an unseen sample by matching the description of the sample to the condition part of every rule. This may lead to three possible cases:

(1) the sample matches exactly one rule;
(2) the sample matches more than one rule;
(3) the sample does not match any of the rules.

In case (1), if the matched rule is a certain one, it is clear that the sample can be predicted by using the decision of the matched rule. However, if the matched rule is a possible one, the prediction is ambiguous. Similar difficulties occur in case (2). Case (3) must be also handled.

We can predict the sample by using the most frequent decision if the sample does not match any of the rules. There are usually two popular decision algorithms for dealing with the remaining cases.

One of them is based on the maximum factor of certainty. Suppose that the sample matches rules $r_1, \ldots, r_i, \ldots, r_n$, and decisions $d_1, \ldots, d_j, \ldots, d_m$ are suggested. Let $\mu_{\text{cer}}(r_i, d_j)$ be the factor of certainty of r_i to d_j, and $\mu_{\text{str}}(r_i)$ be the factor of strength of r_i. The sample can be predicted by using decision d_j that maximizes $\mu_{\text{cer}}(r_i, d_j)$. If ties occur, then select r_i that maximizes $\mu_{\text{str}}(r_i)$.

Another of them is based on the majority voting of the factor of strength. Suppose that the sample matches rules $r_1, \ldots, r_i, \ldots, r_n$, and decisions $d_1, \ldots, d_j, \ldots, d_m$ are suggested. The factor of strength of r_i to d_j is defined as

$$\mu_{str}(r_i, d_j) = \mu_{cer}(r_i, d_j)\mu_{str}(r_i). \quad (12)$$

The voting of the factor of strength of all matched rules to d_j is computed as

$$M_{str}(d_j) = \sum_{r_i} \mu_{str}(r_i, d_j). \quad (13)$$

The sample can be predicted by using decision d_j that maximizes $M_{str}(d_j)$.

4. Rough set based methods for class imbalance learning

4.1. Re-sampling training data

4.1.1. Over-sampling

This method over-samples the minority class to balance the class distribution of a training data set. Concretely, the ith class is over-sampled until the size of the ith class is equal to the size of the maximum class. Over-sampling is a popular method for addressing the class imbalance problem, and studies have shown that over-sampling is effective for class imbalance learning [21,23]. However, it should be noted that over-sampling usually increases training time and may lead to over-fitting since it introduces some exact copies of samples into a training data set [9].

4.1.2. Under-sampling

This method under-samples the majority class to balance the class distribution of a training data set. Concretely, the ith class is under-sampled until the size of the ith class is equal to the size of the minimum class. Some studies showed that under-sampling was better than over-sampling in class imbalance learning [9]. It should also be noted that under-sampling usually discards some potentially useful training samples and may degrade the performance of a resulting classifier [1].

4.1.3. Middle-sampling

This method balances the class distribution of a training data set by integrating the over-sampling and under-sampling methods. Concretely, the ith class is over-sampled or under-sampled until the size of the ith class is equal to the mean size of the maximum and minimum classes.

4.2. Filtering inconsistent samples in boundary regions

In order to deal with the class imbalance problem using a rough set based method, Stefanowski and Wilk [49] introduced removing and filtering techniques to process inconsistent samples in boundary regions. In their works, the inconsistent samples from the majority class in boundary regions are removed or relabeled as belonging to the minority class. Their experimental results indicated that the removing and filtering techniques improved the performance of a rough set based method in class imbalance learning, and the filtering technique performed better than the removing technique. However, no matter which of them is used, the a priori knowledge of class distribution is introduced into boundary regions rather than the whole set of samples. Consequently, their techniques can be used to improve the learning from boundary regions only.

4.3. Weighted rough sets

Liu et al. [28] introduced weights to represent the a priori knowledge of class distribution and proposed a weighted rough set based method to perform class imbalance learning. The weighted rough set based method can be described as detailed below.

$WIS = \langle U, A, W, V, f\rangle$ is called a weighted information system, where $U = \{x_1, \ldots, x_i, \ldots, x_n\}$ is a set of samples, $A = \{a_1, \ldots, a_j, \ldots, a_m\}$ is a set of attributes, $W = \{w(x_1), \ldots, w(x_i), \ldots, w(x_n)\}$ is a weight distribution on U, V is the value domain of A, and f: $U \times A \to V$ is an information function.

For a given weighted information system $WIS = \langle U, A, W, V, f\rangle$, weight distribution W is used to represent a priori knowledge about data. The introduction of weights does not change equivalence relations on U, and so it does not change the upper and lower approximations of arbitrary subset $X \subseteq U$. However, the introduction of weights changes the accuracy of approximation of X.

Let $B \subseteq A$, $X \subseteq U$, and $\underline{B}X$ and $\overline{B}X$ be the lower and upper approximations of X with respect to B, respectively. The weighted accuracy of approximation of X with respect to B is defined as

$$\alpha_B^W(X) = |\underline{B}X|_W / |\overline{B}X|_W, \quad (14)$$

where $|\underline{B}X|_W = \sum_{x_i \in \underline{B}X} w(x_i)$ is the weighted cardinality of $\underline{B}X$ and $|\overline{B}X|_W = \sum_{x_i \in \overline{B}X} w(x_i)$ is the weighted cardinality of $\overline{B}X$.

The weighted accuracy of approximation is computed based on the weighted cardinality of a set. Similarly, the weighted degree of dependency of decision attributes on condition attributes can also be defined.

$WIS = \langle U, A, W, V, f\rangle$ is called a weighted decision table if attribute set $A = C \cup D$, where C is the condition attribute set and D is the decision attribute set. The weighted degree of dependency of D on C is defined as

$$\gamma_C^W(D) = |POS_C(D)|_W / |U|_W, \quad (15)$$

where $POS_C(D) = \cup_{X \in U/D} \underline{C}X$ is the positive region of the partition U/D with respect to C.

Based on weighted rough sets, the traditional learning and classification algorithms can be improved as detailed below.

For attribute reduction, the introduction of weights changes the significance of an attribute.

Based on the weighted degree of dependency, the weighted significance of $a \in C - B$ on the basis of B with respect to D is defined as

$$SIG_\gamma^W(a, B, D) = \gamma_{B\cup\{a\}}^W(D) - \gamma_B^W(D). \quad (16)$$

Shannon's entropy is a popular tool for measuring the uncertainty of knowledge in the traditional rough set theory, but any a priori knowledge about data is not taken into account in Shannon's entropy. Guiasu proposed weighted entropy to address this problem [15]. Suppose that $\Pi_C = \{X_1, \ldots, X_i, \ldots, X_n\}$ and $\Pi_D = \{Y_1, \ldots, Y_j, \ldots, Y_m\}$ are the partitions induced by C and D, respectively. The conditional weighted entropy of D given C is defined as

$$H_W(D \mid C) = -\sum_{i=1}^{n}\sum_{j=1}^{m} (w(X_i \cap Y_j).p(X_i \cap Y_j) \log p(Y_j \mid X_i)) \quad (17)$$

where $w(X_i \cap Y_j) = |X_i \cap Y_j|_W / |X_i \cap Y_j|$ is the weight of $X_i \cap Y_j$.

Based on the conditional weighted entropy, the weighted significance of $a \in C - B$ on the basis of B with respect to D is defined as

$$SIG_H^W(a, B, D) = H_W(D \mid B) - H_W(D \mid B \cup \{a\}). \quad (18)$$

Weighted attribute reduction can be performed based on the weighted significance of an attribute.

For rule extraction, the a priori knowledge of class distribution can be introduced into LEM2 algorithm by employing $|[c] \cap G|_W$ as the heuristic search strategy. The introduction of weights does not change the rule extraction for the exhaustive set of rules.

For rule evaluation and decision, in order to take into account the a priori knowledge of class distribution, extracted rules are evaluated using several weighted factors.

The factor of weighted strength of r is defined as

$$\mu_{\mathrm{str}}^{W}(r) = |\,[r]|_W / \,|\,U|_W, \tag{19}$$

the factor of cover of r is defined as

$$\mu_{\mathrm{cov}}^{W}(r) = |\,[r]|_W / \,|\,[\widetilde{D}_K]|_W, \tag{20}$$

and the factor of certainty of r to d is defined as

$$\mu_{\mathrm{cer}}^{W}(r,d) = |\,[r]_d^+|_W / \,|\,[r]|_W. \tag{21}$$

Based on the weighted facts, weighted decision algorithms similar to the traditional ones can be designed to predict an unseen sample.

It should be noted that the weighted rough set based method degenerates to the traditional rough set based method when an equal weight is assign to each sample.

5. Other methods for class imbalance learning

5.1. C4.5_CS

C4.5 is a widely used decision tree algorithm introduced by Quinlan [42]. In order to perform cost-sensitive learning, Ting [51] extended the standard C4.5 algorithm, and proposed a cost-sensitive (weighted) C4.5 decision tree, denoted by C4.5_CS. In C4.5_CS, Ting changes the class distribution of a data set by sample weighting such that the induced decision tree is in favor of the class with high cost (weight). C4.5_CS can be used to perform class imbalance learning, and it is described as detailed below.

Let n be the size of a training data set, n_j be the number of samples from the jth class, and $n_j(t)$ be the number of samples from the jth class in node t of a decision tree. The probability that a sample from the jth class falls into node t is computed using the ratio of the number of samples from the jth class to the total number of samples in this node, i.e.

$$p(j \mid t) = n_j(t) / \sum_i n_i(t). \tag{22}$$

When weights are introduced into C4.5 decision tree and let $w(j)$ be the weight of the jth class, the weighted number of samples from the jth class in node t of a decision tree is defined as

$$n_j^W(t) = w(j) n_j(t). \tag{23}$$

Similar to $p(j|t)$, $p_W(j|t)$ can be computed using the ratio of the weighted number of samples from the jth class to the total weighted number of samples in node t, i.e.

$$p_W(j \mid t) = n_j^W(t) / \sum_i n_i^W(t). \tag{24}$$

C4.5_CS is the same as C4.5 except that $n_j^W(t)$ is used instead of $n_j(t)$ for computing the test selection criterion in the tree growing process and estimating errors in the pruning process.

5.2. Weighted SVM

Support vector machines (SVM) developed by Vapnik employs the structural risk minimization principle, and has been shown to be superior to traditional methods based on the empirical risk minimization principle [7,54].

SVM minimizes an upper bound on the expected risk rather than minimizes the errors on a training data set. Given training vectors $x_i \in R^m$, $i = 1, \ldots, n$ belonging to two separate classes and a vector $y \in R^n$ such that $y_i \in \{1, -1\}$ represents the class label of training vector x_i, SVM solves the following quadratic programming problem such that it yields the largest margin $(2/||\mathrm{w}||)$ between classes

$$\begin{aligned} &\min_{\mathrm{w},b,\xi} \left\{ \frac{1}{2}\|w\| + C\sum_{i=1}^{n} \xi_i \right\}, \\ &\text{s.t.} \quad y_i(\mathrm{w} \cdot \Phi(x_i) + b) \geqslant 1 - \xi_i, \\ &\xi_i \geqslant 0, i = 1, \ldots, n \end{aligned} \tag{25}$$

where ξ_i is a non-negative slack variable for a non-separable case which is a measure of the misclassification error, $C > 0$ is a penalty term which is applied to the misclassified samples, and Φ is a mapping function for a non-linear case that projects samples from the input space into a feature space.

In order to introduce a priori knowledge about data in SVM, some researchers [5,32] assigned different misclassification costs to different class errors in the objective function, which is naturally allowed in SVM, and developed weighted SVM. Weighted SVM can be used to perform class imbalance learning by assigning to the minority class a larger weight which assures that the minority class is not neglected.

The objective function of weighted SVM can be described as shown below

$$\min_{\mathrm{w},b,\xi} \left\{ \frac{1}{2}\|w\| + C\left(w(1)\sum_{y_i=1} \xi_i + w(-1)\sum_{y_j=-1} \xi_j \right) \right\}, \tag{26}$$

where $w(1)$ is the weight of class 1 and $w(-1)$ is the weight of class -1.

6. Systematic comparative experiments on rough set based class imbalance learning

6.1. Data sets

Twenty UCI data sets [4], which consist of ten two-class data sets and ten multi-class data sets, are used in our experiments, and they are described in Table 1. It can be seen from Table 1 that the class distribution of each data set is skewed. Concretely, the ratio of the majority class to the minority class in size ranges from 1.25 to 3.84 for the two-class data sets, and the ratio of the maximum class to the minimum class in size ranges from 1.48 to 85.5 for the multi-class data sets. Moreover, the size of the minimum class is below 10 for most multi-class data sets.

The missing values in each data set are filled with mean values for continuous attributes and majority values for nominal attributes. Moreover, when a rough set based method is used, all continuous attributes are discretized using the recursive minimal entropy partition (RMEP) proposed by Fayyad and Irani [13,27].

6.2. Performance indexes

The most straightforward way to evaluate the performance of a classifier is based on the confusion matrix analysis. Table 2 shows a confusion matrix for a two-class problem with positive and negative class values. In our study, the minority class is defined as the positive class and the majority class is defined as the negative class. From such a matrix it is possible to extract a number of widely used metrics for measuring the performance of a learning system, such as error rate, defined as $Err = \frac{FP+FN}{TP+FN+TN+FP}$, and overall accuracy, defined as $Acc = \frac{TP+TN}{TP+FN+TN+FP} = 1 - Err$.

However, when the class distribution of a data set is skewed, the use of such measures might lead to misleading conclusions because they are strongly biased towards the majority class. For instance, it is straightforward to create a classifier with the overall accuracy of 99% or the error rate of 1% in a domain where the proportion of the majority class corresponds to 99% of the samples, by simply forecasting every new sample as belonging to the majority class. Furthermore, these measures change as the class distribution

Table 1
Description of data sets (C, continuous; N, nominal)

	Data set	Size	Attribute		Class	Class distribution
1	Echocardiogram	131	6C	1N	2	43/88
2	Hepatitis	155	6C	13N	2	32/123
3	Heart_s	270	6C	7N	2	120/150
4	Breast	286		9N	2	85/201
5	Horse	368	7C	15N	2	136/232
6	Votes	435		16N	2	168/267
7	Credit	690	6C	9N	2	307/383
8	Breast_w	699	9C		2	241/458
9	Tic	958		9N	2	332/626
10	German	1000	24C		2	300/700
11	Zoo	101		16N	7	4/5/8/10/13/20/41
12	Lymphography	148		18N	4	2/4/61/81
13	Wine	178	13C		3	48/59/71
14	Machine	209	7C		8	2/4*2/7/13/27/31/121
15	Glass	214	9C		6	9/13/17/29/70/76
16	Audiology	226		69N	24	1*5/2*7/3/4*3/6/8/9/20/22*2/48/57
17	Heart	303	6C	7N	5	13/35/36/55/164
18	Solar	323		10N	3	7/29/287
19	Soybean	683		35N	19	8/14/15/16/20*9/44*2/88/91*2/92
20	Anneal	898	6C	32N	5	8/40/67/99/684

Table 2
Confusion matrix for a two-class problem

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

changes even if the fundamental performance of a classifier does not change, because overall accuracy and error rate use values from both lines of the confusion matrix.

It would be more interesting if we use a performance metric to disassociate the errors or hits that occur in each class. It is possible to derive four performance metrics from Table 2 to measure the classification performance on the positive and negative classes independently:

(1) True positive rate (accuracy of minority class): $TP_{rate} = TP/(TP + FN)$ is the percentage of positive cases correctly classified as belonging to the positive class.
(2) True negative rate (accuracy of majority class): $TN_{rate} = TN/(TN + FP)$ is the percentage of negative cases correctly classified as belonging to the negative class.
(3) False positive rate: $FP_{rate} = FP/(FP + TN)$ is the percentage of negative cases misclassified as belonging to the positive class.
(4) False negative rate: $FN_{rate} = FN/(FN + TP)$ is the percentage of positive cases misclassified as belonging to the negative class.

The four performance measures have the advantage of being independent of class costs and a priori probabilities. The aim of a classifier is to minimize the false positive and negative rates, or similarly to maximize the true negative and positive rates. Unfortunately, there is usually a tradeoff between FP_{rate} and FN_{rate}, or similarly between TN_{rate} and TP_{rate}. ROC (receiver operating characteristic) graphs can be used to analyze the relationship between FP_{rate} and FN_{rate}, or similarly between TN_{rate} and TP_{rate} [12,41]. ROC graphs are consistent for a given problem even if the distribution of positive and negative samples is highly skewed. The area under the ROC curve (AUC) represents the expected performance as a single scalar. AUC has a known statistical meaning: it is equivalent to the Wilconxon test of ranks, and is also equivalent to several other statistical measures for evaluating classification and ranking models [18].

In our study, we employ accuracy of minority class, accuracy of majority class, overall accuracy and AUC as the performance indexes to evaluate our experiments. For the multi-class problems, the accuracy of the minimum class is defined as the accuracy of minority class, the accuracy of the maximum class is defined as the accuracy of majority class, and the AUC is computed using the method proposed in [19].

6.3. Comparison among three strategies for class imbalance learning

In the rough set based methods, there are usually three strategies used for class imbalance learning, and they are weighting, resampling and filtering. In order to evaluate the performance of each strategy, we compare the strategies in this section.

The methods employed for this comparison are described and configured as detailed below. Among the methods, RS is the traditional method, and the others are the methods for class imbalance learning.

(1) *WRS.* The weighted rough set based method with typical configurations. The weighted significance of attributes based on the weighted degree of dependency is employed to perform weighted attribute reduction, the weighted rule extraction algorithm for the minimum set of rules is employed to perform weighted rule extraction, the weighted decision algorithm based on the majority voting of the factor of weighted strength is employed to predict an unseen sample. Moreover, an inverse class probability weight is assigned to each sample for class imbalance learning. Suppose that there are $n_1, \ldots, n_i, \ldots, n_l$ samples in decision classes $Y_1, \ldots, Y_i, \ldots, Y_l$, respectively. Then the inverse class probability weight of each sample from decision class Y_i is $1/n_i$.
(2) *RS.* The traditional rough set based method. RS can be considered WRS with equal weighting. Instead of an inverse class probability weight, an equal weight is assigned to each sample.
(3) *OS.* The random over-sampling method. The ith class is randomly over-sampled until the size of the ith class is equal to the size of the maximum class, and then RS is used to perform learning and classification. The random over-sampling method is selected because it is simple and is competitive with other complicated over-sampling methods [1].

(4) *US*. The random under-sampling method. The *i*th class is randomly under-sampled until the size of the *i*th class is equal to the size of the minimum class, and then RS is used to perform learning and classification.
(5) *MS*. The random middle-sampling method. The *i*th class is randomly over-sampled or under-sampled until the size of the *i*th class is equal to the mean size of the maximum and minimum classes, and then RS is used to perform learning and classification.
(6) *FILTER*. The filtering method proposed by Stefanowski and Wilk [49]. The inconsistent samples from the majority class in boundary regions are relabeled as belonging to the minority class, and then RS is used to perform learning and classification.

The comparative experiments are performed using 10-fold cross validation. The accuracy of minority class, the accuracy of majority class, the overall accuracy and the AUC achieved by every method are listed in Tables 3 and 4. It can be seen from the experimental results that almost all the methods for class imbalance learning improve the accuracy of minority class and the AUC, and decrease the accuracy of majority class and the overall accuracy compared to the traditional rough set based method. Class imbalance learning aims at improving the accuracy of minority class but not decreasing the accuracy of majority class too much. The AUC can be used to analyze the relationship between the accuracy of minority class and the accuracy of majority class. Therefore, we employ the AUC as the primary performance index and the accuracy of minority class as the secondary performance index to evaluate the performance of every method in class imbalance learning.

Almost all the methods based on these strategies improve the AUC and the accuracy of minority class compared to the traditional rough set based method. This means that all the strategies are

Table 3
Accuracy of minority and majority classes achieved by different strategies

Data set	Accuracy of minority class						Accuracy of majority class					
	RS	WRS	FILTER	OS	US	MS	RS	WRS	FILTER	OS	US	MS
Echocardiogram	0.2450	0.7350	0.2900	0.6150	0.2850	0.5900	0.8639	0.6833	0.8639	0.6569	0.6194	0.6125
Hepatitis	0.6917	0.7333	0.6917	0.7167	0.8750	0.7583	0.9353	0.9256	0.9436	0.8763	0.8455	0.8923
Heart_s	0.7583	0.7583	0.7583	0.7250	0.7750	0.7417	0.8000	0.8267	0.8133	0.7800	0.7933	0.7933
Breast	0.2431	0.3847	0.2889	0.3986	0.5319	0.4708	0.8407	0.7860	0.8307	0.8060	0.6226	0.7167
Horse	0.9412	0.9412	0.9412	0.9555	0.9637	0.9341	0.9739	0.9739	0.9739	0.9652	0.9565	0.9739
Votes	0.9585	0.9647	0.9585	0.9346	0.9643	0.9522	0.9625	0.9738	0.9662	0.9701	0.9288	0.9662
Credit	0.8013	0.8472	0.7948	0.8403	0.8439	0.8243	0.8252	0.8409	0.8174	0.8252	0.8250	0.8174
Breast_w	0.9208	0.9418	0.9250	0.9292	0.9333	0.9418	0.9607	0.9672	0.9607	0.9629	0.9564	0.9540
Tic	0.7927	0.8137	0.8078	0.7957	0.8562	0.8709	0.9361	0.9136	0.9489	0.9073	0.8882	0.9041
German	0.0267	0.9067	0.0333	0.6033	0.4500	0.6933	0.9843	0.3043	0.9843	0.6500	0.2886	0.5343
Zoo	0.7000	0.8000	0.8000	0.8000	0.7000	0.7000	1.0000	1.0000	1.0000	1.0000	0.8500	0.9250
Lymphography	0.3333	0.5857	0.3333	0.6167	0.3476	0.7167	0.8903	0.8417	0.8903	0.7667	0.6069	0.7542
Wine	0.8850	0.9350	0.8850	0.9350	0.9350	0.9550	0.9714	0.9286	0.9714	0.9018	0.8571	0.8714
Machine	0.6000	0.8000	0.6000	0.5000	0.3000	0.4000	0.8096	0.7429	0.8179	0.8263	0.3667	0.6846
Glass	0.6000	0.7000	0.6000	0.2000	0.5000	0.5000	0.6946	0.6143	0.6821	0.5911	0.1018	0.5357
Audiology	0.7000	0.7000	0.7000	0.7000	0.3000	0.6000	0.9500	0.9333	0.9667	0.9333	0.1500	0.7767
Heart	0	0.3000	0	0.1000	0.3000	0.1500	0.8246	0.7937	0.8371	0.7750	0.5305	0.6963
Solar	0.0333	0.1000	0.0333	0.1000	0.1000	0.0667	0.9617	0.8159	0.9617	0.7454	0.4080	0.7564
Soybean	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.7956	0.8611	0.7956	0.8189	0.6056	0.8067
Anneal	1.0000	1.0000	1.0000	0.9500	1.0000	0.9750	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Average	0.6115	0.7474	0.6221	0.6708	0.6481	0.6920	0.8990	0.8363	0.9013	0.8379	0.6601	0.7986

Table 4
Overall accuracy and AUC achieved by different strategies

Data set	Overall accuracy						AUC					
	RS	WRS	FILTER	OS	US	MS	RS	WRS	FILTER	OS	US	MS
Echocardiogram	0.6566	0.7038	0.6714	0.6484	0.5077	0.6093	0.5544	0.7092	0.5769	0.6360	0.4522	0.6013
Hepatitis	0.8842	0.8838	0.8904	0.8433	0.8512	0.8629	0.8135	0.8295	0.8176	0.7965	0.8603	0.8253
Heart_s	0.7815	0.7963	0.7889	0.7556	0.7852	0.7704	0.7792	0.7925	0.7858	0.7525	0.7842	0.7675
Breast	0.6644	0.6676	0.6712	0.6852	0.5954	0.6437	0.5419	0.5853	0.5598	0.6023	0.5773	0.5938
Horse	0.9618	0.9618	0.9618	0.9618	0.9590	0.9592	0.9576	0.9576	0.9576	0.9604	0.9601	0.9540
Votes	0.9610	0.9702	0.9633	0.9564	0.9427	0.9610	0.9605	0.9692	0.9623	0.9523	0.9466	0.9592
Credit	0.8145	0.8435	0.8072	0.8319	0.8333	0.8203	0.8133	0.8440	0.8061	0.8328	0.8345	0.8209
Breast_w	0.9471	0.9585	0.9485	0.9513	0.9485	0.9499	0.9408	0.9545	0.9428	0.9460	0.9449	0.9479
Tic	0.8863	0.8789	0.8998	0.8685	0.8768	0.8925	0.8644	0.8637	0.8783	0.8515	0.8722	0.8875
German	0.6970	0.4850	0.6990	0.6360	0.3370	0.5820	0.5055	0.6055	0.5088	0.6267	0.3693	0.6138
Zoo	0.9200	0.9500	0.9300	0.9500	0.8309	0.9100	0.8970	0.9362	0.9070	0.9362	0.8367	0.9187
Lymphography	0.8176	0.7767	0.8176	0.7171	0.4824	0.7033	0.7389	0.7658	0.7389	0.7427	0.5583	0.7531
Wine	0.9373	0.9382	0.9373	0.9324	0.9157	0.9271	0.9474	0.9534	0.9474	0.9509	0.9397	0.9483
Machine	0.6552	0.6893	0.6695	0.7024	0.4264	0.6214	0.6964	0.7697	0.7102	0.7009	0.6250	0.7028
Glass	0.6955	0.6346	0.6909	0.6130	0.3760	0.6210	0.7933	0.7892	0.7921	0.7018	0.6957	0.7734
Audiology	0.7749	0.7708	0.7968	0.7664	0.1860	0.6694	0.8044	0.7900	0.8150	0.7919	0.5594	0.7543
Heart	0.5216	0.5381	0.5578	0.5117	0.3797	0.4755	0.5485	0.6048	0.5773	0.5740	0.5528	0.5672
Solar	0.8671	0.7438	0.8671	0.6878	0.3848	0.6906	0.5488	0.5290	0.5488	0.5280	0.4520	0.5141
Soybean	0.8229	0.8916	0.8140	0.8843	0.7368	0.8800	0.9007	0.9585	0.8995	0.9552	0.8992	0.9528
Anneal	1.0000	1.0000	1.0000	0.9922	0.9922	0.9978	1.0000	1.0000	1.0000	0.9808	0.9904	0.9931
Average	0.8133	0.8041	0.8191	0.7948	0.6674	0.7774	0.7803	0.8104	0.7866	0.7910	0.7355	0.7924

effective for class imbalance learning. The detailed analysis of each strategy is given as shown below:

(1) Compared to RS, WRS has an average increase of 0.0301 in terms of the AUC and an average increase of 0.1359 in terms of the accuracy of minority class. Sample weighting remarkably improves the performance of a rough set based method in class imbalance learning.
(2) The methods based on the strategy of re-sampling comprise OS, MS and US. Among these methods, MS achieves the best performance, OS achieves the second best performance and US achieves the worst performance. Compared to RS, MS has an average increase of 0.0121 in terms of the AUC and an average increase of 0.0805 in terms of the accuracy of minority class.
(3) In our experiments, FILTER cannot remarkably increase the AUC and the accuracy of minority class compared to RS. This may be explained by the fact that FILTER introduced the a priori knowledge of class distribution into boundary regions rather than the whole set of samples. Consequently, it can be used to improve the learning from boundary regions only. Compared to RS, FILTER has an average increase of 0.0063 in terms of the AUC and an average increase of 0.0106 in terms of the accuracy of minority class.

Among the strategies for class imbalance learning, weighting is better than re-sampling, and re-sampling is better than filtering. Moreover, the weighted rough set based method achieves the best performance.

6.4. Comparison among various configurations of the weighted rough set based method

In the weighted rough set based method, several candidate configurations can be used. In order to optimize the performance of the weighted rough set based method in class imbalance learning, we compare the configurations in this section.

For this comparison, the weighted rough set based method is configured as follows.

(1) *WAR.* WRS with weighted attribute reduction only. An inverse class probability weight is assigned to each sample for weighted attribute reduction, while an equal weight is assigned to each sample for weighted rule extraction and weighted decision.
(2) *WRE.* WRS with weighted rule extraction and weighted decision only. An inverse class probability weight is assigned to each sample for weighted rule extraction and weighted decision, while an equal weight is assigned to each sample for weighted attribute reduction.
(3) *WRS_ENT.* WRS with the weighted attribute reduction based on weighted entropy, instead of that based on the weighted degree of dependency.
(4) *WRS_EXH.* WRS with the rule extraction for the exhaustive set of rules, instead of the weighted rule extraction for the minimum set of rules.
(5) *WRS_CER.* WRS with the weighted decision based on the maximum factor of weighted certainty, instead of that based on the majority voting of the factor of weighted strength.

The comparative experiments are performed using 10-fold cross validation. Table 5 lists the accuracy of minority class and the AUC obtained by the weighted rough set based method with different configurations. It can be seen from the results:

(1) Compared to RS, WAR has an average increase of 0.0097 and WRE has an average increase of 0.0165 in terms of the AUC. In terms of the accuracy of minority class, compared to RS, WAR has an average decrease of 0.0024 and WRE has an average increase of 0.0959. WRE achieves better performance than WAR, and this means that the weighted rule extraction and weighted decision have greater influence on the performance of the weighted rough set based method than the weighted attribute reduction.
(2) Compared to WRS, WRS_ENT has an average decrease of 0.0022 in terms of the AUC, and an average decrease of 0.0045 in terms of the accuracy of minority class. This means that the straightforward weighted significance measure of

Table 5
Accuracy of minority class and AUC achieved by the weighted rough set based method with different configurations

Data set	Accuracy of minority class					AUC				
	WAR	WRE	WRS_ENT	WRS_EXH	WRS_CER	WAR	WRE	WRS_ENT	WRS_EXH	WRS_CER
Echocardiogram	0.6100	0.6900	0.7350	0.7600	0.7100	0.7029	0.6568	0.7092	0.7099	0.7022
Hepatitis	0.5750	0.6917	0.8583	0.8250	0.7333	0.7670	0.8016	0.8962	0.8631	0.8295
Heart_s	0.7583	0.7583	0.7500	0.7667	0.7583	0.7925	0.7858	0.7850	0.7900	0.7858
Breast	0.2556	0.3847	0.3514	0.4444	0.3972	0.5381	0.5903	0.5662	0.6254	0.5940
Horse	0.9341	0.9412	0.9412	0.9484	0.9412	0.9540	0.9554	0.9597	0.9655	0.9554
Votes	0.9526	0.9647	0.9761	0.9529	0.9647	0.9632	0.9655	0.9692	0.9615	0.9692
Credit	0.8276	0.8212	0.8308	0.8504	0.8408	0.8304	0.8179	0.8319	0.8417	0.8395
Breast_w	0.9292	0.9292	0.9292	0.9583	0.9377	0.9493	0.9460	0.9471	0.9628	0.9525
Tic	0.7927	0.8137	0.8250	0.8496	0.8197	0.8644	0.8637	0.8606	0.8992	0.8643
German	0.1133	0.8100	0.9067	0.9100	0.8067	0.5267	0.6393	0.6055	0.6079	0.5769
Zoo	0.7000	0.8000	0.7000	0.8000	0.8000	0.9137	0.9129	0.9135	0.9362	0.9237
Lymphography	0.5857	0.3333	0.7190	0.6857	0.5857	0.7859	0.7188	0.7984	0.7901	0.7741
Wine	0.9150	0.9100	0.9350	0.9350	0.9150	0.9514	0.9585	0.9498	0.9647	0.9484
Machine	0.1000	0.7000	0.8000	0.8000	0.8000	0.6659	0.7226	0.7650	0.7803	0.7634
Glass	0.4000	0.6000	0.7000	0.8000	0.7000	0.7194	0.7921	0.7801	0.7893	0.7887
Audiology	0.7000	0.7000	0.5000	0.5000	0.8000	0.8015	0.7997	0.7272	0.7575	0.7819
Heart	0	0.2000	0.3000	0.3000	0.3000	0.5752	0.5686	0.6165	0.6107	0.6071
Solar	0.0333	0.1000	0.1000	0.1000	0.1000	0.5462	0.5306	0.5381	0.5246	0.5399
Soybean	1.0000	1.0000	1.0000	1.0000	1.0000	0.9525	0.9110	0.9447	0.9451	0.9532
Anneal	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Average	0.6091	0.7074	0.7429	0.7593	0.7455	0.7900	0.7968	0.8082	0.8163	0.8075

an attribute based on the weighted degree of dependency outperforms the complicated one based on weighted entropy for weighted attribute reduction.

(3) Compared to WRS, WRS_EXH has an average increase of 0.0059 in terms of the AUC, and an average increase of 0.0119 in terms of the accuracy of minority class. This means that the rule extraction for the exhaustive set of rules outperforms the weighted rule extraction for the minimum set of rules. However, it should be noted that WRS_EXH usually generates more rules than WRS. In our experiments, WRS_EXH generates averagely 632.9 rules on all the data sets, while WRS generates averagely 54.64.

(4) Compared to WRS, WRS_CER has an average decrease of 0.0029 in terms of the AUC, and an average decrease of 0.0019 in terms of the accuracy of minority class. This means that the weighted decision based on the majority voting of the factor of weighted strength outperforms that based on the maximum factor of weighted certainty.

It can be concluded from the comparative experiments that the weighted rule extraction and weighted decision have greater influence on the performance of the weighted rough set based method than the weighted attribute reduction, and the weighted attribute reduction based on the weighted degree of dependency, the rule extraction for the exhaustive set of rules and the weighted decision based on the majority voting of the factor of weighted strength are the optimal configurations for class imbalance learning.

6.5. Comparison between the weighted rough set based method and other developed methods

In this section, we compare the weighted rough set based method with the other developed methods for class imbalance learning. The methods employed for the comparison comprise two decision tree based methods and two SVM based methods. All the methods are described and configured as follows:

(a) Two decision tree based methods (C4.5 is the traditional method, and C4.5_CS is the method for class imbalance learning).
   (1) *C4.5*. C4.5 decision tree proposed by Quinlan [42].
   (2) *C4.5_CS*. Cost-sensitive (weighted) C4.5 decision tree proposed by Ting [51]. An inverse class probability weight is assigned to each sample for class imbalance learning.

(b) Two SVM based methods (SVM is the traditional method, and WSVM is the method for class imbalance learning).
   (1) *SVM*. SVM with linear kernel function and parameter C=100.
   (2) *WSVM*. Weighted SVM [5] with linear kernel function and parameter $C = 100$. An inverse class probability weight is assigned to each sample for class imbalance learning.

The comparative experiments are performed using 10-fold cross validation. The accuracy of minority class and the AUC obtained by the methods are listed in Table 6. It can be seen from the results:

(1) The decision tree based methods achieve satisfactory performance in class imbalance learning. Compared to RS, C4.5 has an average increase of 0.027 in terms of the AUC and an average increase of 0.0919 in terms of the accuracy of minority class. Through sample weighting, C4.5_CS has an average increase of 0.0055 in terms of the AUC and an average increase of 0.0086 in terms of the accuracy of minority class compared to C4.5. It can be seen from the comparison between C4.5 and C4.5_CS that the decision tree based methods are not sensitive to the class distribution of a data set. Compared to WRS_EXH, i.e. the weighted rough set based method with optimal configurations, C4.5_CS has an average decrease of 0.0035 in terms of the AUC and an average decrease of 0.0473 in terms of the accuracy of minority class. C4.5_CS is worse than WRS_EXH.

(2) Through comparing different kernel functions and different values of parameter C, we find that the linear kernel function is effective for class imbalance learning, and a big enough C is helpful for improving the performance of the SVM based methods. We select the linear kernel function and $C = 100$ in our experiments. Compared to RS, SVM has an average increase of 0.0335 in terms of the AUC and an average increase of 0.071 in terms of the accuracy of minority class.

Table 6
Accuracy of minority class and AUC achieved by C4.5 and SVM based methods

Data set	Accuracy of minority class				AUC			
	C4.5	C4.5_CS	SVM	WSVM	C4.5	C4.5_CS	SVM	WSVM
Echocardiogram	0.5050	0.4600	0.4750	0.6900	0.6219	0.5835	0.6528	0.7256
Hepatitis	0.7583	0.6917	0.6583	0.5667	0.8346	0.7929	0.7965	0.7465
Heart_s	0.8267	0.8067	0.7917	0.8667	0.7717	0.7450	0.8358	0.8333
Breast	0.3167	0.5042	0.3653	0.5569	0.5937	0.5978	0.6204	0.6264
horse	0.9264	0.9489	0.8747	0.8896	0.9588	0.9679	0.9072	0.9018
Votes	0.9765	0.9824	0.9699	0.9816	0.9751	0.9781	0.9662	0.9739
Credit	0.8435	0.8502	—	—	0.8552	0.8599	–	–
Breast_w	0.9458	0.9128	0.9625	0.9625	0.9565	0.9367	0.9670	0.9670
Tic	0.8915	0.9307	0.5357	0.6804	0.9338	0.9398	0.6928	0.6924
German	0.4833	0.5400	0.4900	0.6133	0.6645	0.6571	0.6893	0.7138
Zoo	0.7000	0.8000	0.7000	0.8000	0.9137	0.9600	0.9583	0.9667
Lymphography	0.7500	0.7857	1.0000	0.8000	0.7799	0.8100	0.9252	0.8407
Wine	0.9100	0.9500	0.9800	0.9600	0.9513	0.9607	0.9730	0.9678
Machine	0.6846	0.6929	–	–	0.6690	0.7128	–	–
Glass	0.6000	0.7000	0.6000	0.7000	0.7895	0.8176	0.7570	0.6863
Audiology	0.8500	0.5500	0.8000	0.8000	0.8262	0.8315	0.9415	0.9100
Heart	0.1000	0	0.1000	0.4000	0.5727	0.5587	0.5944	0.6192
Solar	0	0.1333	0.3000	0.6000	0.5000	0.5667	0.5750	0.6176
Soybean	1.0000	1.0000	1.0000	1.0000	0.9780	0.9799	0.9822	0.9833
Anneal	1.0000	1.0000	–	–	1.0000	1.0000	–	–
Average	0.7034	0.7120	0.6825	0.7569	0.8073	0.8128	0.8138	0.8102

Through sample weighting, WSVM has an average decrease of 0.0036 in terms of the AUC and an average increase of 0.0744 in terms of the accuracy of minority class compared to SVM. It can be seen from the comparison between SVM and WSVM that the SVM based methods are not sensitive to the class distribution of a data set in terms of the AUC, but sample weighting is helpful for improving the accuracy of minority class. Compared to WRS_EXH, WSVM has an average decrease of 0.0061 in terms of the AUC and an average decrease of 0.0024 in terms of the accuracy of minority class. WSVM is worse than WRS_EXH. Moreover, the SVM based methods with linear kernel function are quite time-consuming, and they cannot generate results on three data sets in an acceptable period of time.

It can be seen from these comparative experiments that WRS_EXH, i.e. the weighted rough set based method with optimal configurations, outperforms the decision tree and SVM based methods. Moreover, the performance of the weighted rough set based method with other configurations, i.e. WRS, WRS_ENT and WRS_CER, is also comparable to the best performance of the decision tree and SVM based methods. This means that the weighted rough set based method is effective for class imbalance learning.

We also find that the rough set based methods are more sensitive to the class distribution of a data set compared to the decision tree and SVM based methods. When the class distribution of a data set is skewed, it is necessary to employ some techniques for class imbalance learning to improve the performance of a rough set based method.

7. Conclusions

This paper performs systematic comparative researches on rough set based class imbalance learning.

Firstly, we compare the strategies of weighting, re-sampling and filtering used in the rough set based methods for class imbalance learning. We find that weighting is better than re-sampling, and re-sampling is better than filtering. The weighted rough set based method outperforms the methods based on re-sampling and filtering.

Secondly, in order to optimize the performance of the weighted rough set based method in class imbalance learning, we compare various candidate configurations of the weighted rough set based method. From the comparative experiments, we find that the weighted rule extraction and weighted decision have greater influence on the performance of the weighted rough set based method than the weighted attribute reduction, and the weighted attribute reduction based on the weighted degree of dependency, the rule extraction for the exhaustive set of rules and the weighted decision based on the majority voting of the factor of weighted strength are the optimal configurations for class imbalance learning.

Finally, we compare the weighted rough set based method with the other developed methods for class imbalance learning. WRS_EXH, i.e. the weighted rough set based method with optimal configurations, outperforms the decision tree and SVM based methods. Moreover, the performance of the weighted rough set based method with other configurations, i.e. WRS, WRS_ENT and WRS_CER, is also comparable to the best performance of the decision tree and SVM based methods. This means that the weighted rough set based method is effective for class imbalance learning. We also find that the rough set based methods are more sensitive to the class distribution of a data set compared to the decision tree and SVM based methods. When the class distribution of a data set is skewed, it is necessary to employ some techniques for class imbalance learning to improve the performance of a rough set based method.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant 60703013.

References

[1] G. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explorations 6 (1) (2004) 20–29.
[2] J. Bazan, A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables, in: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery, Physica-Verlag, Heidelberg, 1998, pp. 321–365.
[3] M. Beynon, Reducts within the variable precision rough sets model: a further investigation, European Journal of Operational Research 134 (2001) 592–605.
[4] C. Blake, E. Keogh, C.J. Merz, UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine. Available from: <http://www.ics.uci.edu/~mlearn/MLRepository.html>, 1998.
[5] U. Brefeld, P. Geibel, F. Wysotzki, Support vector machines with sample dependent costs, Proceedings of 14th European Conference on Machine Learning, 2003, pp. 23–34.
[6] N. Chawla, N. Japkowicz, A. Kolcz (Eds.), ICML'2003 Workshop on Learning from Imbalanced Data Sets (II), Proceedings available at: <http://www.site.uottawa.ca/~nat/Workshop2003/workshop2003.html>, 2003.
[7] C. Cortes, V. Vapnik, Support-vector network, Machine Learning 20 (1995) 273–297.
[8] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, 2000.
[9] C. Drummond, R.C. Holte, C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, Working Notes of the ICML'03 Workshop Learning from Imbalanced Data Sets, 2003.
[10] I. Duntsch, G. Gediga, Uncertainty measures of rough set prediction, Artificial Intelligence 106 (1) (1998) 109–137.
[11] R.E. Fawcett, F. Provost, Adaptive fraud detection, Data Mining and Knowledge Discovery 3 (1) (1997) 291–316.
[12] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (2006) 861–874.
[13] U. Fayyad, K. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, Proceeding of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufman, 1993. pp. 1022–1027.
[14] S. Greco, B. Matarazzo, R. Slowinski, Rough set theory for multicriteria decision analysis, European Journal of Operational Research 129 (2001) 1–47.
[15] S. Guiasu, Information Theory with Applications, McGraw-Hill, International Book Company, New York, 1977.
[16] D.M. Grzymala-Busse, J.W. Grzymala-Busse, The usefulness of machine learning approach to knowledge acquisition, Computational Intelligence 11 (1995) 268–279.
[17] J.W. Grzymala-Busse, LERS- a system for learning from samples based on rough sets, in: R. Slowinski (Ed.), Intelligent Decision Support, Kluwer Academic Publishers, 1992, pp. 3–18.
[18] D.J. Hand, Construction and Assessment of Classification Rules, John Wiley and Sons, 1997.
[19] D.J. Hand, R.J. Till, A simple generalization of the area under the ROC curve to multiple class classification problems, Machine Learning 45 (2) (2001) 171–186.
[20] X.-H. Hu, N. Cercone, Data mining via discretization, generalization and rough set feature selection, Knowledge and Information Systems 1 (1) (1999) 33–60.
[21] N. Japkowicz, Learning from imbalanced data sets: a comparison of various strategies, Working Notes of the AAAI'00 Workshop Learning from Imbalanced Data Sets, 2000, pp. 10–15.
[22] N. Japkowicz (Eds.), AAAI Workshop on Learning from Imbalanced Data Sets, Technical Report WS-00-05, AAAI Press, Menlo Park, CA, 2003.
[23] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intelligent Data Analysis 6 (5) (2002) 429–450.
[24] M. Kryszkiewicz, Rules in incomplete information systems, Information Sciences 113 (3–4) (1999) 271–292.
[25] M. Kryszkiewicz, Comparative study of alternative type of knowledge reduction in inconsistent systems, International Journal of Intelligent Systems 16 (2001) 105–120.
[26] J.-Y. Liang, Z.-B. Xu, The algorithm on knowledge reduction in incomplete information systems, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (2002) 95–103.
[27] H. Liu, F. Hussain, C.L. Tan, M. Dash, Discretization: an enabling technique, Data Mining and Knowledge Discovery 6 (2002) 393–423.
[28] J.F. Liu, Q.H. Hu, D.R. Yu, A weighted rough set based method developed for class imbalance learning, Information Sciences 178 (4) (2008) 1235–1256.

[29] M.A. Maloof, Learning when data sets are imbalanced and when costs are unequal and unknown, in: Proceedings of Working Notes ICML'03 Workshop Learning from Imbalanced Data Sets, 2003.
[30] J.-S. Mi, W.-Z. Wu, W.-X. Zhang, Approaches to knowledge reduction based on variable precision rough set model, Information Sciences 159 (2004) 255–272.
[31] R.S. Michalski, A theory and methodology of inductive learning, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, Morgan Kaufman, San Mateo, CA, 1983, pp. 83–134.
[32] E. Osuna, R. Freund, F. Girosi, Support vector machines: training and applications, AI Memo 1602, Massachusetts Institute of Technology, 1997.
[33] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (1982) 341–356.
[34] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.
[35] Z. Pawlak, Rough sets and intelligent data analysis, Information Sciences 147 (2002) 1–12.
[36] Z. Pawlak, J.W. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough sets, Communications of the ACM 38 (11) (1995) 89–95.
[37] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 177 (1) (2007) 28–40.
[38] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 177 (1) (2007) 3–27.
[39] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Information Sciences 177 (1) (2007) 41–73.
[40] R.C. Prati, G.E.A.P.A. Batista, M.C. Monard, Class imbalances versus class overlapping: an analysis of a learning system behavior, MICAI, 2004, pp. 312–321.
[41] F.J. Provost, T. Fawcett, Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions, KDD, 1997, pp. 43–48.
[42] J.R. Quinlan, C4.5 Programs for Machine Learning, Morgan Kaufman, CA, 1988.
[43] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, P. atamatopoulos, Stacking classifiers for anti-spam filtering of E-mail, in: L. Lee, D. Harman (Eds.), Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001), Carnegie Mellon University, 2001, pp. 44–50.
[44] Q. Shen, A. Chouchoulas, A rough-fuzzy approach for generating classification rules, Pattern Recognition 35 (11) (2002) 2425–2438.
[45] D. Slezak, Approximate reducts in decision tables, Proceedings of Sixth Internation Conference on Information Management of Uncertainty in Knowledge-based System, Granada, 1996.
[46] D. Slezak, Approximate entropy reducts, Fundamenta Informaticae 53 (3–4) (2002) 365–390.
[47] J. Stefanowski, On rough set based approaches to induction of decision rules, in: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery, Physica-Verlag, Heidelberg, Germany, 1998, pp. 501–529.
[48] J. Stefanowski, D. Vanderpooten, A general two stage approach to rule induction from examples, in: W. Ziarko (Ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, London, UK, 1994, pp. 317–325.
[49] J. Stefanowski, S. Wilk, Rough sets for handling imbalanced data: combining filtering and rule-based classifiers, Fundamenta Informaticae 72 (1) (2006) 379–391.
[50] Q. Tao, G.-W. Wu, F.-Y. Wang, J. Wang, Posterior probability support vector machines for unbalanced data, IEEE Transactions on Neural Networks 16 (6) (2005) 1561–1573.
[51] K.M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Transactions on Knowledge and Data Engineering 14 (3) (2002) 659–665.
[52] S. Tsumoto, Automated extraction of medical expert system rules from clinical databases based on rough set theory, Information Sciences 112 (1–4) (1998) 67–84.
[53] S. Tsumoto, Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model, Information Sciences 162 (2) (2004) 65–80.
[54] V. Vapnik, Statistical Learning Theory, Wiley, New York, NY, 1998.
[55] G.Y. Wang, J. Zhao, J.J. An, et al., A comparative study of algebra viewpoint and information viewpoint in attribute reduction, Fundamenta Informaticae 68 (3) (2005) 289–301.
[56] G.M. Weiss, Mining with rarity – problems and solutions: a unifying framework, SIGKDD Explorations 6 (1) (2004) 7–19.
[57] G.M. Weiss, F. Provost, The effect of class distribution on classifier learning: an empirical study, Technical Report ML-TR-44, Rutgers University, Department of Computer Science, 2001.
[58] B. Zadrozny, C. Elkan, Learning and making decisions when costs and probabilities are both unknown, Technical Report CS2001-0664, UCSD, 2001.
[59] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Transactions on Knowledge and Data Engineering 18 (1) (2006) 63–77.
[60] W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences 46 (1993) 39–59.