A comparative study on rough set based class imbalance learning

Jinfu Liu *, Qinghua Hu, Daren Yu

Harbin Institute of Technology, Harbin 150001, China

A R T I C L E   I N F O

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

Keywords:
Rough sets
Class imbalance learning
Sample weighting

A B S T R A C T

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.

© 2008 Elsevier B.V. All rights reserved.

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 [19,59]. 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.
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

\[
IS = (U, A, V, f) \text{ is called an information system, where } U = \{x_1, \ldots, x_n\} \text{ is a set of samples, } A = \{a_1, \ldots, a_m\} \text{ is a set of attributes, } V \text{ is the value domain of } A, \text{ and } f: U \times A \rightarrow V \text{ is an information function.}
\]

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

\[
IND(B) = \{(x, y) \in U \times U \mid f(x, a) = f(y, a), \forall a \in B\}.
\]

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

\[
\Pi_B = U/B = \{[x]_B \mid x \in U\},
\]

where \([x]_B\) is the equivalence class containing \( x \). All the elements in \([x]_B\) 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 upper and lower approximations of \( X \) with respect to \( B \), denoted by \( \text{BX} \) and \( \text{BX} \), respectively, are defined as

\[
\text{BX} = \{y \mid [x]_B \subseteq X\}, \quad \text{BX} = \{y \mid [x]_B \cap X = \emptyset\}.
\]

Lower approximation \( \text{BX} \) 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 \( \text{BX} \) 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 \( B \). \( B_X(X) = \text{BX} \cap \text{BX} \) is called the boundary region of \( X \) with respect to \( B \). \( X \) is definable with respect to \( B \) if \( B_X(X) = \emptyset \), 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

\[
\gamma_B(X) = \frac{|\text{BX}|}{|\text{BX}|},
\]

where \(|\cdot|\) denotes the cardinality of a set. \( X \) is definable with respect to \( B \) if \( \gamma_B(X) = 1 \), 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 reducible of \( A \) if \( B \) is independent and \( U/B = U/A \).

A reducible is a minimal subset of attributes that preserves the indiscernibility relation determined by full attributes. There is usually more than one reducible for a given information system, and the intersection of all the reducibles 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.

\[
\gamma_C(D) = |\text{POS}_C(D)|/|U|,
\]

where \( \text{POS}_C(D) = \text{U}x \cap D \) 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 \( D \).

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 reducibles is an NP-hard problem. However, it is usually adequate enough for most real-world applications to find one of the reducibles. Based on the significance of an attribute, a heuristic attribute reduction algorithm can be designed to find a reducible.

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 \( IS = (U, A, V, f) \) be a given decision table, \( B \subseteq A \) 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 reducible of \( C \) if \( B \) is independent with respect to \( D \) and \( \gamma_B(D) = \gamma_C(D) \).

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

3.2.1. Two popular significance measures of attributes

Based on Shannon’s entropy, the significance of \( a \in C \) on the basis of \( B \) with respect to \( D \) is defined as

\[
\text{SIG}(a, B, D) = \gamma_{B-a}(D) - \gamma_B(D).
\]

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 \( a \in C \) on the basis of \( B \) with respect to \( D \) is defined as

\[
\text{SIG}(a, B, D) = H(D \mid B) - H(D \mid B \cup \{a\})
\]

where \( H(D \mid B) \) denotes the conditional entropy of \( D \) given \( B \). Suppose that \( P_B = \{X_1, \ldots, X_m\} \) are the partition induced by \( B \), and \( P_B = \{Y_1, \ldots, Y_m\} \) are the partition induced by \( D \).

\[
H(D \mid B) = - \sum_{i=1}^{m} \sum_{j=1}^{n} p(X_i, Y_j) \log p(Y_j \mid X_i).
\]

Based on the significance of an attribute, a heuristic attribute reduction algorithm can be designed to find a reducible 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 $\hat{D}$, is first defined on a given decision table. Each element of $\hat{D}$ is a single or joint decision. According to $\hat{D}$, all the samples in the decision table is then partitioned into a family of disjoint subsets, denoted by $\hat{Y}$. Each element of $\hat{Y}$ is the lower approximation of a decision class which corresponds to a single decision of $\hat{D}$, or one of the disjoint subsets of the boundary region of a decision class which corresponds to a joint decision of $\hat{D}$. 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: $BND(Y_1) = (BY_1 \cap BY_2 - BY_1) \cup (BY_1 \cap BY_3 - BY_1) \cup (BY_1 \cap BY_2 \cap BY_3)$. It is clear that $\hat{Y}$ is consistent with $\hat{D}$. Finally, for each $k \in \hat{Y}$, LEM2 employs a heuristic strategy to extract a minimal set of rules.

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

$$\text{if } \Phi \text{ then } \tilde{D}_k,$$  

where $\Phi$ is called the condition part of $r$, satisfying $[\Phi] \neq \emptyset$, and $\tilde{D}_k$ is called the decision part of $r$. If $\tilde{D}_k$ is a single decision, $r$ is called a certain rule. Otherwise, if $\tilde{D}_k$ 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 $[c] \cap \mathcal{G}$ 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 $K \in \hat{Y}$. 
Output: A minimum set of rules $\mathcal{R}$.

1. begin 
2. $G \leftarrow K$; $R \leftarrow \emptyset$; 
3. while $G \neq \emptyset$ do 
4. begin 
5. $\Phi \leftarrow \emptyset$; 
6. $\Phi_C \leftarrow \{ [c] \cap \mathcal{G} \}$; 
7. while ($\Phi = \emptyset$) or (not($\Phi \subseteq K$)) do 
8. begin 
9. select $c \in \Phi_C$ such that $[c] \cap \mathcal{G}$ is maximum. If ties occur then select $c$ with the smallest $[c]$. If further ties occur then select the first $c$ from the list; 
10. $\Phi \leftarrow \Phi \cup \{ c \}$; 
11. $G \leftarrow [c] \cap \mathcal{G}$; 
12. $\Phi_C \leftarrow \{ [c] \cap \mathcal{G} \}$; 
13. $\Phi_C \leftarrow \Phi_C - \Phi$; 
14. end 
15. for each $c \in \Phi$ do 
16. $G \subseteq [\{ \Phi - \{ c \} \} \subseteq K$ then 
17. $\Phi \leftarrow \Phi - \{ c \}$; 
18. create a rule $r$ based on $\Phi$; 
19. $R \leftarrow R \cup \{ r \}$; 
20. $G \leftarrow K - \cup_{r \in R} \{ r \}$; 
21. end 
22. end 
23. for each $r \in R$ do 
24. if $\cup_{r \in R} \{ r \} = K$ then 
25. $R \leftarrow R - \{ r \}$; 
26. end 
27. end 
28. 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, $D_k = \{ d_1, \ldots, d_j, \ldots, d_m \}$ be the decision part of $r$, $[r]$ be the cover of $r$, $[\hat{D}_k]$ be the cover of $D_k$ and $[r]_{\hat{D}_k} = [r] \cap [\hat{D}_k]$ be the positive cover of $r$ on $d$, where $d \in D_k$. The factor of strength of $r$ is defined as

$$\mu_{\text{st}}(r) = |[r]_{\hat{D}_k}| / |[\hat{D}_k]|,$$

the factor of cover of $r$ is defined as

$$\mu_{\text{cov}}(r) = |[r]| / |D_k|,$$

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

$$\mu_{\text{cert}}(r, d) = |[r]_{\hat{D}_k}| / |[r]|.$$

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 handled also.

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 $r_1, \ldots, r_n$ and decisions $d_1, \ldots, d_n$ are suggested. Let $\mu_{\text{cert}}(r_i, d_j)$ be the factor of certainty of $r_i$ to $d_j$, and $\mu_{\text{st}}(r_i)$ be the factor of strength of $r_i$. The sample can be predicted by using decision $d_j$ that maximizes $\mu_{\text{cert}}(r_i, d_j)$. If ties occur, then select $r_i$ that maximizes $\mu_{\text{cert}}(r_i)$.

Another of them is based on the majority voting of the factor of strength. Suppose that the sample matches rules $r_1, \ldots, r_n$ and decisions $d_1, \ldots, d_n$ are suggested. The factor of strength of $r_i$ to $d_j$ is defined as
\[ \mu_{\text{out}} (r_j, d_j) = \mu_{\text{mir}}(r_j, d_j) \mu_{\text{int}}(r_j). \]  

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

\[ M_{\text{w}}(d_j) = \sum_{i=1}^{n} \mu_{\text{out}}(r_i, d_j). \]

The sample can be predicted by using decision \( d_j \) that maximizes \( M_{\text{w}}(d_j) \).

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 \( i \)-th class is over-sampled until the size of the \( i \)-th 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 \( i \)-th class is under-sampled until the size of the \( i \)-th 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 \( i \)-th class is 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.

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.
\[ u^w_i(r) = |r|_W / |U|_W. \] 
\[ n^{w_i}_r(d) = |r|_W / |D_k|_W. \] 
\[ \min_{w, b} \left\{ \frac{1}{2} ||w|| + C \sum_{i=1}^{n} \zeta_i \right\}, \] 
\[ \text{s.t. } y_i(w \cdot \phi(x_i) + b) \geq 1 - \zeta_i, \] 
where \( \zeta_i \) 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 \( \phi \) 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:

\[ \min_{w, b, \zeta} \left\{ \frac{1}{2} ||w|| + C \left( w(1) \sum_{i=1}^{n} \zeta_i + w(-1) \sum_{i=1}^{n} \zeta_i \right) \right\}, \] 

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 performance measures. In our study, the minority class is defined as the positive class and the majority class is defined as the negative class.
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 Wilcoxon 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 \( n_1, \ldots, n_l \) samples in decision classes \( Y_1, \ldots, Y_l \), 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 \( i \)th class is randomly over-sampled until the size of the \( i \)th 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 ith class is randomly under-sampled until the size of the ith 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 ith class is randomly over-sampled or under-sampled until the size of the ith 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...
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 0.0059. 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>
<thead>
<tr>
<th>Data set</th>
<th>Accuracy of minority class</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WAR</td>
<td>WRE</td>
</tr>
<tr>
<td>Echocardiogram</td>
<td>0.6100</td>
<td>0.6900</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>0.5750</td>
<td>0.6917</td>
</tr>
<tr>
<td>Heart_s</td>
<td>0.7583</td>
<td>0.7583</td>
</tr>
<tr>
<td>Breast</td>
<td>0.2556</td>
<td>0.3847</td>
</tr>
<tr>
<td>Horse</td>
<td>0.9341</td>
<td>0.9412</td>
</tr>
<tr>
<td>Votes</td>
<td>0.9526</td>
<td>0.9647</td>
</tr>
<tr>
<td>Credit</td>
<td>0.8276</td>
<td>0.8212</td>
</tr>
<tr>
<td>Breast_w</td>
<td>0.5929</td>
<td>0.9292</td>
</tr>
<tr>
<td>Tic</td>
<td>0.7927</td>
<td>0.8137</td>
</tr>
<tr>
<td>German</td>
<td>0.1133</td>
<td>0.8100</td>
</tr>
<tr>
<td>Zoo</td>
<td>0.7000</td>
<td>0.8000</td>
</tr>
<tr>
<td>Lymphography</td>
<td>0.5887</td>
<td>0.3333</td>
</tr>
<tr>
<td>Wine</td>
<td>0.9150</td>
<td>0.9100</td>
</tr>
<tr>
<td>Machine</td>
<td>0.1000</td>
<td>0.7000</td>
</tr>
<tr>
<td>Glass</td>
<td>0.4000</td>
<td>0.6000</td>
</tr>
<tr>
<td>Audiology</td>
<td>0.7000</td>
<td>0.7000</td>
</tr>
<tr>
<td>Heart</td>
<td>0.0020</td>
<td>0.0000</td>
</tr>
<tr>
<td>Solar</td>
<td>0.0333</td>
<td>0.1000</td>
</tr>
<tr>
<td>Soybean</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Ankle</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Average</td>
<td>0.6091</td>
<td>0.7074</td>
</tr>
</tbody>
</table>
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.

(3) 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 0.0019 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

<table>
<thead>
<tr>
<th>Data set</th>
<th>Accuracy of minority class</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C4.5</td>
<td>C4.5_CS</td>
</tr>
<tr>
<td>Echocardiogram</td>
<td>0.5050</td>
<td>0.4600</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>0.7583</td>
<td>0.6917</td>
</tr>
<tr>
<td>Heart_s</td>
<td>0.8267</td>
<td>0.8067</td>
</tr>
<tr>
<td>Breast</td>
<td>0.3167</td>
<td>0.5042</td>
</tr>
<tr>
<td>horse</td>
<td>0.9264</td>
<td>0.9489</td>
</tr>
<tr>
<td>Votes</td>
<td>0.9765</td>
<td>0.9824</td>
</tr>
<tr>
<td>Credit</td>
<td>0.8435</td>
<td>0.8502</td>
</tr>
<tr>
<td>Breast_w</td>
<td>0.9458</td>
<td>0.9128</td>
</tr>
<tr>
<td>Tic</td>
<td>0.8915</td>
<td>0.9307</td>
</tr>
<tr>
<td>German</td>
<td>0.4833</td>
<td>0.5400</td>
</tr>
<tr>
<td>Zoo</td>
<td>0.7000</td>
<td>0.8000</td>
</tr>
<tr>
<td>Lymphography</td>
<td>0.7500</td>
<td>0.7857</td>
</tr>
<tr>
<td>Wine</td>
<td>0.9100</td>
<td>0.9500</td>
</tr>
<tr>
<td>Machine</td>
<td>0.6846</td>
<td>0.6929</td>
</tr>
<tr>
<td>Glass</td>
<td>0.6000</td>
<td>0.7000</td>
</tr>
<tr>
<td>Audiology</td>
<td>0.8500</td>
<td>0.5500</td>
</tr>
<tr>
<td>Heart</td>
<td>0.1000</td>
<td>0</td>
</tr>
<tr>
<td>Solar</td>
<td>0</td>
<td>0.1333</td>
</tr>
<tr>
<td>Soybean</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Anneal</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>Average</td>
<td>0.7034</td>
<td>0.7120</td>
</tr>
</tbody>
</table>
Through sample weighting, WSVM has a 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. W SVM 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 a great 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

[21] N. Japkowicz, Learning from imbalanced data sets: a comparison of various strategies, Working Notes of the AAM\textsuperscript{3}I\textsuperscript{0} Workshop Learning from Imbalanced Data Sets, 2000, pp. 10–15.
[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.


