A resampling ensemble algorithm for classification of imbalance problems

Yun Qian a,b, Yanchun Liang a, Mu Li a, Guoxiang Feng a, Xiaohu Shi a,⁎

a Key Laboratory for Symbol Computation and Knowledge Engineering of National, Education Ministry, College of Computer Science and Technology, Jilin University, #2699 Qianjin Street, Changchun 130012, Jilin Province, China
b College of Electrical and Information Engineering, Beihua University, #1 Xinshan Street, Jilin 132021, China

ARTICLE INFO

Article history:
Received 19 February 2014
Received in revised form 10 May 2014
Accepted 1 June 2014
Available online 18 June 2014

Keywords:
Ensemble Learning
Imbalanced classification
Resampling scale

ABSTRACT

In this paper, a resampling ensemble algorithm is developed focused on the classification problems for imbalanced datasets. In the method, the small classes are oversampled and large classes are undersampled. The resampling scale is determined by the ratio of the min class number and max class number. And multiple machine learning methods are selected to construct the ensemble. Numerical results show that the algorithm performance is highly related to the ratio of minority class number and attribute number. When the ratio is less than 3, the performance will be greatly hindered. Experimental results also show that the ensemble of different types of methods could improve the algorithm performance efficiently.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Learning classifiers from imbalanced data has attracted a significant amount of interest in recent years. This is because in real world, imbalanced data exist in many applications, such as fault diagnosis [1], medical diagnosis [2], intrusion detection [3,4], text classification [5,6], financial fraud detection [7], data stream classification [8], and so on. In those applications, there are often one or some minority classes possessing very few samples compared with the other classes. And most of time, the “small” classes are more important than those “large” ones. Because of the unbalance data distribution of imbalanced learning problems, it is often difficult to obtain good performance for most cases by using traditional classifiers where a balanced distribution of classes is assumed and an equal misclassification cost for each class is assigned. As a result, traditional classifiers tend to be overwhelmed by the majority classes and ignore the minority ones, which is not acceptable in many real applications [9].

Many approaches have been developed to address this problem, which could be divided into three categories depending on the style for dealing with the imbalance of classes [9]. The first category could be called data level techniques, which attempt to balance the data distribution by over-sampling and (or) under-sampling methods [10]. The second one is algorithm level approaches. They develop new algorithms or modify existed methods to take into account the significance of “small” classes [11]. The third one is cost-sensitive methods which combine both algorithm and data level approaches by assigning different misclassification costs for each class [12].

While in recent years, with the rapid developing of ensemble methods for classification, they have been applied to imbalanced data classification. Ensemble learning is a machine learning paradigm where multiple learners (called base learners) are trained to solve the same problem [13]. Usually, an ensemble is much stronger than those base learners which are contained. This is because that different base learners are accurate on different instances, specializing in different subdomains of the problem, so that they can complement each other [13]. The base learners are generated from the training set by any machine learning algorithms, such as neural networks, support vector machine, decision tree, and so on. If all the base learners are trained by the same learning algorithm, the base learners are homogeneous, otherwise they are heterogeneous. Many kinds of ensemble methods are developed, among of them, Boosting, Bagging and Stacking are three representative ones [9].

Because of the outstanding performance of ensemble methods, they are applied to imbalanced dataset by combining with other techniques. For example, Chawla et al. have developed SMOTEBoost algorithm by integrating Adaboost (the most famous boosting algorithm) and Synthetic Minority Oversampling Technique (SMOTE) [14]. Similarly with SMOTEBoost, RUSBoost also introduces data sampling into the Adaboost algorithm, while it applies
random undersampling to the majority class, but SMOTEBoost creates synthetic new minority class instances by operating in the “feature space” [15]. Blaszczyński et al. integrate a selective data pre-processing method SPIDER with Ivotes ensemble algorithm developing the framework called Ivotes [16]. EasyEnsemble creates several training subsets from the majority class, and trains a learner using each of them together with the minority class, and then ensembles all the outputs of those learners. BalanceCascade trains the base learners sequentially, where in each step the majority class examples which are correctly classified by the current trained learners are removed from further consideration [17]. Focusing on the class imbalance and non-stationary feature of current trained learners are removed from further consideration.

In this paper, both oversampling for “small” classes and undersampling for “large” classes are performed, where oversampling is conducted according to the SMOTE technique and undersampling is conducted by randomly selecting. And then, multiple machine learning methods are selected to construct an ensemble. The combination of base learners is performed according to the Bagging strategy. The scales of oversampling and undersampling are analyzed and empirical equations are derived. Numerical results show that our developed schedule is more effective than existing methods.

2. Method

2.1. Framework of Bagging algorithm

In the ensemble method, a number of base learners should be generated first, whether in parallel or sequential style. And then, the base learners are combined to construct an ensemble. According to the combination style, ensemble methods could be divided into three groups, namely Boosting, Bagging, and Stacking [9]. In this paper, the combination strategy is performed like the Bagging algorithm. Therefore, we give a brief introduction of the Bagging algorithm here.

Denote \( X \) and \( Y \) as the sample space and class label space, respectively. Assume that there are \( m \) samples of \( L \) classes in the training data set. And the training dataset is denoted as \( D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \), where \( x_i \in X \) and \( y_i \in Y = \{1,2,\ldots,L\} \). Then the Bagging algorithm will train a number of base learners from different bootstrap samples. Generally, a bootstrap sample is randomly selected from the training set with replacement, with the same size of the training set. After all the base learners are obtained, they will be aggregated by the majority voting method. The Bagging algorithm could reduce the model variance and hence improve the accuracy.

2.2. Resampling ensemble algorithm for classification of imbalance problems

In our proposed framework, undersampling for “large” classes and SMOTE oversampling for “small” classes are integrated. Moreover, several different machine learning methods are conducted to construct the ensemble. Both undersampling and oversampling could improve the imbalance situation of the dataset. However, oversampling too much might cause over fitting and hindering to improve the performance of the algorithm. On the other hand, undersampling of the big classes could enhance the diversity of the base learners, which is a crucial factor affecting the performance [18]. But we also should avoid losing information too much and pay attention to the balance between different classes. Therefore, how to determine the final sizes of processed classes is of great importance. We will give detailed empirical analysis in Section 4. The framework of our proposed Resampling Ensemble Algorithm (REA) is shown as Fig. 1.

Firstly, the original training data are divided into different classes. Next those majority classes are performed undersampling and minority classes are performed oversampling according to the SMOTE technique. Denote the sample number of the \( i \)-th class as \( n_i \) and assume \( A = n_1 \leq n_2 \leq \ldots \leq n_L = B \), then we have \( \sum n_i = m \). The resampling scale ratio of the \( l \)-th class \( S_l \) could be calculated according to Eq. (1):

\[
S_l = \frac{\beta B - \alpha A}{B - A} + \frac{(\alpha - \beta)AB}{(B - A)l},
\]

(1)

where \( \alpha (\alpha \geq 1) \) is the scale parameter of the smallest class and \( \beta (0 \leq \beta \leq 1) \) is the scale parameter of the biggest class. It is easy to find that when \( l = 1 \), \( S_1 = \alpha \), and when \( l = L \), \( S_L = \beta \). Therefore, the resampling number of the \( l \)-th class \( re_num_l \) is

\[
re_num_l = \text{Round}(S_l \cdot n_l)
\]

(2)

where \( \text{Round}(\cdot) \) is the round function. Then in the \( l \)-th bootstrap step, for a majority class, say the \( i \)-th class, we just need to randomly select \( re_num_l \) samples from the original \( n_i \) samples into \( D_i \); while for a minority class, say the \( j \)-th class, besides putting all the original \( n_j \) samples into \( D_j \), we also need to produce \( re_num_l - n_j \) new samples according to SMOTE technique. Each new sample is produced as follows: Randomly select a sample in class \( l \) \((x_l, y_l)\), and then compute \( K \) nearest neighbors of the sample, at last generate a new sample to the Gaussian distribution. In this experiment, REA is applied to 7 multi-classification problems, which are illustrated in Table 1. In the second experiment, REA is applied to 7 multi-classification problems, which are illustrated in Table 2. In this experiment, NB, k-Nearest Neighbors (kNN) algorithm, and Back Propagation (BP) networks are selected as the base learning methods. Here, parts of data sets are modified from multiclass data sets. The detailed information of binary data sets is summarized in Table 1. In the second experiment, REA is applied to 7 multi-classification problems, which are illustrated in Table 2. In this experiment, NB, k-Nearest Neighbors (kNN) algorithm, and Back Propagation (BP) networks are selected as the base learning methods. Here, parts of data sets are modified from multiclass data sets. The detailed information of binary data sets is summarized in Table 1. In the second experiment, REA is applied to 7 multi-classification problems, which are illustrated in Table 2.

3. Experimental results

3.1. Data sets and experimental settings

The experiment data sets are all selected from UCI data sets (http://archive.ics.uci.edu/ml) [19], which include 20 binary classification data sets and 7 multi-classification data sets. We apply the proposed REA method on the binary data sets by only using the Naïve Bayesian (NB) algorithm as the base learning methods. Here, parts of data sets are modified from multiclass data sets. The detailed information of binary data sets is summarized in Table 1. In the second experiment, REA is applied to 7 multi-classification problems, which are illustrated in Table 2. In this experiment, NB, k-Nearest Neighbors (kNN) algorithm, and Back Propagation (BP) networks are selected as the base learning methods, separately and combined together. The Gaussian naive Bayes is selected as the NB models, which assumes that the continuous features associated with each class are distributed according to the Gaussian distribution. For BP networks and kNN, the model structures are randomly selected for each base learner. Denote the attribute number as \( q \) and the training data size as \( n \), respectively. Naturally, in a BP model, the input node number should be \( q \) and the output node number is set as one for all datasets, while the node number in the hidden layer is randomly selected within the interval \([q, 2q]\). The step length is set as 0.05 for all the BP models. In the kNN method, neighbor’s number is also varied for different base learners, which is randomly selected within the interval \([2, \sqrt{q(n)}]\). All the experiments have been performed on a PC with 3.2 GHz processor and 2 G memory.

3.2. Evaluations

For imbalance classification problems, the precision (or error rate) is not an appropriate evaluation criterion, or at least not only...
Algorithm: Resampling Ensemble Algorithm (REA)

Input
Training Data Set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
Number of base learners $T$
Base learning algorithms $L_t (t = 1, \ldots, T)$

Process
For $t = 1, \ldots, T$

$D_t = \text{Bootstrap}(D)$; % Generate a bootstrap sample from $D$
$h_t = L_t(D_t)$ % Train a base learner $h_t$ from the bootstrap sample

End for

Output $H(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} \mathbf{1}(y = h_t(x))$ % the value of $\mathbf{1}(\alpha)$ is 1 if $\alpha$ is true and 0 otherwise

Fig. 1. Framework of Resampling Ensemble Algorithm (REA).

Fig. 2. Pseudo code of the Resampling Ensemble Algorithm (REA).
Algorithm: Bootstrap Algorithm

Input
- Training Data Set $D = \{(x_i, y_i), (x_2, y_2), \ldots, (x_m, y_m)\}$
- Data number of each class in $D$: $A = n_1 \leq n_2 \leq \cdots \leq n_L = B$
- Scale parameter of the smallest class $\alpha$ ($\alpha > 1$)
- Scale parameter of the biggest class $\beta$ (0 $\leq \beta \leq 1$)
- Number of nearest neighbors $K$

Process
For $l = 1, \ldots, L$
\[
S_l = \frac{\beta B - \alpha A + (\alpha - \beta) AB}{B - A} \frac{1}{n_l} \quad \text{% Calculate the Scale parameter of each class}
\]
End for
For $l = 1, \ldots, L$
\[
\text{% For each class}
\]
If $S_l > 1$
\[
\text{% For minority classes: oversampling}
\]
\[
\text{Population}_l = \{\text{All samples in class } l\}
\]
For $i = 1, 2, \ldots, \text{Round}((S_l - 1)n_l)$
\[
\text{Randomly select a sample in class } l, (x_i, y_i)
\]
Compute $K$ nearest neighbors of the sample
Generate a new sample of class $l$, by averaging the $K$ nearest neighbors with randomly weighs
Add the new sample in $\text{Population}_l$
End For
Else
\[
\text{% For majority classes: undersampling}
\]
\[
\text{Population}_l = \{\text{All samples in class } l\}
\]
For $i = 1, 2, \ldots, \text{Round}((1-S_l)n_l)$
\[
\text{Randomly select a sample in class } l, (x_i, y_i)
\]
Delete the selected sample from $\text{Population}_l$
End For
\[
\text{% For each class}
\]
\[
\text{Population}_l = \{\text{All samples in class } l\}
\]
For $i = 1, 2, \ldots, \text{Round}((1-S_l)n_l)$
\[
\text{Randomly select a sample in class } l, (x_i, y_i)
\]
Delete the selected sample from $\text{Population}_l$
End For
\[
\text{% Generate bootstrap sample for base learner } h_l
\]
Output $D_l = \cup \text{Population}_l$

Fig. 3. Pseudo code of the Bootstrap Algorithm.

one appropriate criterion. F-measure and G-mean are two commonly used ones to evaluate the performance of an algorithm for imbalance data classification. For binary classification problems, the confusion matrix is shown in Table 3. Generally, the minority class is called negative class. Then the precision, recall, F-measure and G-mean are defined by Eqs. (3)-(6)

Precision $= TP/(TP + FP)$ (3)
Recall $= TP/(TP + FN)$ (4)
\[
F - \text{measure} = 2\text{PrecisionRecall}/(\text{Precision + Recall}) (5)
\]
\[
G - \text{mean} = \sqrt{TN/TN + FP \times TP/TP + FN} (6)
\]
For multi-class problems, F-measure and G-mean are also two most popular used measurements. However, they could not be applied directly as defined for bi-class problems. Table 4 gives the confusion matrix for a multi-class problem. The true prediction of the ith class is the number of $n_{ii}$ and the false prediction of the ith class into the jth class is the number of $n_{ij}$, respectively. Let $R_i$, $P_i$ and $F_i$ respectively denote recall and precision of class $C_i$, $P_i$ and $F_i$ are then defined as
\[
R_i = \frac{n_{ii}}{\sum_j n_{ij}} (7)
\]
\[
P_i = \frac{n_{ii}}{\sum_j n_{ji}} (8)
\]
\[
F_i = \frac{2R_iP_i}{R_i + P_i} (9)
\]
Therefore, F-measure of the ith class is calculated as an average of precision and recall of the ith class:
\[
\text{F-measure} = \frac{2R_iP_i}{R_i + P_i} (9)
\]
And F-measure of all classes could be calculated as
\[
\text{F-measure} = \frac{2}{L} \sum_{i=1}^{L} R_i \sum_{i=1}^{L} P_i (10)
\]
In bi-classification problems, G-mean could be considered as the geometric means to the recall values of two classes [20]. In multi-classification problems, it is extended by defining G-mean as the geometric means of recall values to all classes [21]
\[
G - \text{mean} = \left( \prod_{i=1}^{L} R_i \right)^{1/L} (11)
\]
So, in our experiments, Eqs. (5), (6) are used to measure the algorithms performance for bi-class problems and Eqs. (10), (11) for multi-class problems, respectively.
3.3. Experiment for bi-class problems and scale parameter analysis

In this experiment, only the Bayes algorithm is used as the base learner. Firstly, we perform REA on the first 14 bi-class datasets listed in Table 1. According to numerical results, the scale parameter selection is analyzed and two empirical equations are proposed. Parts of numerical results are also compared to the existing method. And then, REA is executed on the other 6 datasets in Table 1 to verify the two empirical equations. Lastly, the relationship between experimental performance and the ratio of attribute number and data scale is also studied using numerical results of the former 14 datasets and testing by the latter 6 ones.

To analyze the scale parameter’s effectiveness, all experiments are calculated 100 times by iterating the parameter $\alpha$ from 1 to 2 and $\beta$ from 1 to 0.1 with step size 0.1. Fig. 4 gives F-measure results of the 14 datasets. In the figure, the $x$-axis represents the value of parameter $\alpha$, the $y$-axis the value of parameter $\beta$, and the $z$-axis the value of F-measure, respectively. The results of G-mean are similar with those of F-measure. Due to the limited space, they are not listed.

From Fig. 4, it could be found that the peak locations and scopes of all the datasets are different. For convenience, we summarize the peak scopes and centers of $\alpha$ and $\beta$ in Table 5. It is easy to find that in most cases, the peak scopes are really wide, and it is better to call them platforms rather than peaks. We find that the peak centers and the ratios of the min class number and max class number slightly comply with linear relationship. And also considering the peak scopes are wide, we assume $\alpha$ and $\beta$ are linear functions of the ratio of the min class number and max class number. Taking $\alpha$ for example, assume
Fig. 4. F-measure results. (a) Car, (b) Water, (c) Transfusion, (d) Pima, (e) Operative, (f) Vehicle, (g) Balance, (h) Wdbc, (i) Heart, (j) Wpbc, (k) Ionosphere, (l) Housing, (m) Cmc and (n) Haberman.
\[ \alpha = aR + b. \] Then, the parameters of \( a \) and \( b \) are determined by optimizing the following function:

\[
\min_{a,b} \sum_{i=1}^{N} \left( \frac{(aR_i + b - \frac{l_i + u_i}{2})^2 + \delta_i}{C_0} \right)
\]

(12)

where \( N \) is the number of datasets, and being taken 14 here, \( R_i \) is the ratio of the min class number and max class number of the \( i \)th dataset, and \( l_i \) and \( u_i \) are the lower border and upper border of the \( \alpha \) peak scope on the \( i \)th dataset, \( \delta_i \) is the penalty factor for the violation of the \( \alpha \) peak scope constraint, which is defined by

\[
\delta_i = \begin{cases} 
\xi & \text{if } aR_i + b > u_i \text{ or } aR_i + b < l_i \\
0 & \text{otherwise}
\end{cases}
\]

(13)

here \( \xi \) is a predefined constant, being taken 10 in our experiments.

Using particle swarm optimization algorithm to solve this problem, we obtain the optimized \( a \) and \( b \) as \(-0.097\) and \(1.428\). Therefore, we determine \( \alpha \) by the following equation:

\[
\alpha = -0.097R + 1.428
\]

(14)

Similarly, we get the empirical equation for \( \beta \) as follows:

\[
\beta = 0.198R + 0.738
\]

(15)

Fig. 5 shows the relationship between peak centers and the ratios of the min class number and max class number.

To examine the effectiveness of the proposed method, the results are compared with those in Ref. [17], which has executed 10 same datasets with us. The F-measure and G-mean comparison results are listed in Tables 6 and 7. In the tables, the results of Bagg, Ada, Asym, SMB, Under, Over, SMOTE, Chan, Cascade, Easy, RF, BRF, Under-RF, and Over-RF are cited from Ref. [17] (See Tables VII, VIII, X, and XI in Ref. [17]). The detailed information of the 14 methods should refer to Ref. [17].

“REA-Best” represents the best result of the proposed method on all \( \alpha \) and \( \beta \) values, while “REA_Eqs(14-15)” represents the result of the proposed method taking \( \alpha \) and \( \beta \) values according to Eqs. (14) and (15). From Table 6 and Table 7, we could find that our proposed method evidently outperforms the 14 comparison methods on both F-measure and G-mean metrics, except for Ionosphere dataset. Among all the 14 comparison methods, “Easy” obtains the best performance both on average F-measure and average G-means, which are \(0.661\) and \(0.775\) respectively. For “REA-Best”, the average F-measure and average G-means on the same 10 datasets are \(0.942\) and \(0.964\), and those of “REA_Eqs(14-15)” are \(0.932\) and \(0.955\), respectively. Therefore, “REA-Best” and “REA_Eqs(14-15)” are \(42.62\%\) and \(41.09\%\) higher than “Easy” on the average F-measure, and \(24.37\%\) and \(23.23\%\) on the average G-mean, respectively. Moreover, we
select the best F-measure of the 14 comparison methods for each of dataset, and compute the average value of the 10 “best F-measures”, which is 0.674, and that of G-means is 0.780. There also exists a big gap between the “average of best one” and “REA_Best” or “REA_Eqs(14-15)”. Therefore, we could conclude that the proposed method is significantly better than the existing methods on the selected datasets.

Comparing the results of “REA-Best” and “REA_Eqs(14-15)”, they obtain the same results on 7 datasets from all the 14 datasets on both F-measure and G-mean metrics. For all the 14 datasets, the average F-measure value of “REA_Eqs(14-15)” is only 1.37% less than that of “REA-Best”, while for the average G-mean value, the former is only 0.90% less than the latter. To further demonstrate the effectiveness of the two equations, the proposed method is also performed on the latter 6 datasets in Table 1. The numerical results are shown in Table 8 and Table 9. “REA_Eqs(14-15)” and “REA-Best” get the same results on 3 datasets from all the 6 datasets on both F-measure and G-mean metrics. For all the 6 datasets, the average F-measure value of “REA_Eqs(14-15)” is 2.468% less than that of “REA-Best”, and its average G-mean value is 1.209% less than that of “REA-Best”. It shows that Eqs. (14) and (15) are effective to determine the parameters of $\alpha$ and $\beta$.

Table 6
F-measure comparison results.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Car</th>
<th>Water</th>
<th>Transfusion</th>
<th>Pima</th>
<th>Operative</th>
<th>Vehicle</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CART</td>
<td>0.857</td>
<td>–</td>
<td>–</td>
<td>0.584</td>
<td>–</td>
<td>0.523</td>
<td>0.000</td>
</tr>
<tr>
<td>Bagg</td>
<td>0.933</td>
<td>–</td>
<td>–</td>
<td>0.644</td>
<td>–</td>
<td>0.526</td>
<td>0.000</td>
</tr>
<tr>
<td>Ada</td>
<td>0.967</td>
<td>–</td>
<td>–</td>
<td>0.611</td>
<td>–</td>
<td>0.545</td>
<td>0.000</td>
</tr>
<tr>
<td>Asym</td>
<td>0.966</td>
<td>–</td>
<td>–</td>
<td>0.613</td>
<td>–</td>
<td>0.561</td>
<td>0.000</td>
</tr>
<tr>
<td>SMB</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.641</td>
<td>–</td>
<td>0.606</td>
<td>0.001</td>
</tr>
<tr>
<td>Over</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.644</td>
<td>–</td>
<td>0.623</td>
<td>0.175</td>
</tr>
<tr>
<td>Under</td>
<td>0.884</td>
<td>–</td>
<td>–</td>
<td>0.609</td>
<td>–</td>
<td>0.539</td>
<td>0.000</td>
</tr>
<tr>
<td>SMOTE</td>
<td>0.930</td>
<td>–</td>
<td>–</td>
<td>0.627</td>
<td>–</td>
<td>0.615</td>
<td>0.149</td>
</tr>
<tr>
<td>Chan</td>
<td>0.916</td>
<td>–</td>
<td>–</td>
<td>0.618</td>
<td>–</td>
<td>0.608</td>
<td>0.156</td>
</tr>
<tr>
<td>Cascade</td>
<td>0.917</td>
<td>–</td>
<td>–</td>
<td>0.649</td>
<td>–</td>
<td>0.623</td>
<td>0.194</td>
</tr>
<tr>
<td>Easy</td>
<td>0.880</td>
<td>–</td>
<td>–</td>
<td>0.660</td>
<td>–</td>
<td>0.638</td>
<td>0.184</td>
</tr>
<tr>
<td>RF</td>
<td>0.307</td>
<td>–</td>
<td>–</td>
<td>0.641</td>
<td>–</td>
<td>0.544</td>
<td>0.000</td>
</tr>
<tr>
<td>BRF</td>
<td>0.521</td>
<td>–</td>
<td>–</td>
<td>0.663</td>
<td>–</td>
<td>0.633</td>
<td>0.167</td>
</tr>
<tr>
<td>Under-RF</td>
<td>0.513</td>
<td>–</td>
<td>–</td>
<td>0.668</td>
<td>–</td>
<td>0.633</td>
<td>0.168</td>
</tr>
<tr>
<td>Over-RF</td>
<td>0.518</td>
<td>–</td>
<td>–</td>
<td>0.656</td>
<td>–</td>
<td>0.564</td>
<td>0.000</td>
</tr>
<tr>
<td>REA-Best</td>
<td>1</td>
<td>0.889</td>
<td>0.973</td>
<td>0.991</td>
<td>.947</td>
<td>0.994</td>
<td>1</td>
</tr>
<tr>
<td>REA_Eqs(14), (15)</td>
<td>1</td>
<td>0.851</td>
<td>0.966</td>
<td>0.981</td>
<td>.947</td>
<td>0.994</td>
<td>1</td>
</tr>
</tbody>
</table>

Data set       | Wdbc | Heart | Wpbc | Ionosphere | Housing | Cmc | Haberman |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CART</td>
<td>0.895</td>
<td>–</td>
<td>0.373</td>
<td>0.831</td>
<td>0.420</td>
<td>0.356</td>
<td>0.335</td>
</tr>
<tr>
<td>Bagg</td>
<td>0.938</td>
<td>–</td>
<td>0.410</td>
<td>0.883</td>
<td>0.419</td>
<td>0.362</td>
<td>0.334</td>
</tr>
<tr>
<td>Ada</td>
<td>0.956</td>
<td>–</td>
<td>0.432</td>
<td>0.907</td>
<td>0.475</td>
<td>0.388</td>
<td>0.348</td>
</tr>
<tr>
<td>Asym</td>
<td>0.956</td>
<td>–</td>
<td>0.444</td>
<td>0.910</td>
<td>0.485</td>
<td>0.400</td>
<td>0.360</td>
</tr>
<tr>
<td>SMB</td>
<td>–</td>
<td>–</td>
<td>0.452</td>
<td>–</td>
<td>0.530</td>
<td>0.393</td>
<td>0.377</td>
</tr>
<tr>
<td>Over</td>
<td>–</td>
<td>–</td>
<td>0.449</td>
<td>–</td>
<td>0.529</td>
<td>0.429</td>
<td>0.442</td>
</tr>
<tr>
<td>Under</td>
<td>0.952</td>
<td>–</td>
<td>0.427</td>
<td>0.900</td>
<td>0.470</td>
<td>0.383</td>
<td>0.338</td>
</tr>
<tr>
<td>SMOTE</td>
<td>0.957</td>
<td>–</td>
<td>0.459</td>
<td>0.907</td>
<td>0.532</td>
<td>0.421</td>
<td>0.405</td>
</tr>
<tr>
<td>Chan</td>
<td>0.954</td>
<td>–</td>
<td>0.448</td>
<td>0.910</td>
<td>0.523</td>
<td>0.437</td>
<td>0.380</td>
</tr>
<tr>
<td>Cascade</td>
<td>0.957</td>
<td>–</td>
<td>0.454</td>
<td>0.905</td>
<td>0.529</td>
<td>0.436</td>
<td>0.438</td>
</tr>
<tr>
<td>Easy</td>
<td>0.951</td>
<td>–</td>
<td>0.452</td>
<td>0.901</td>
<td>0.543</td>
<td>0.454</td>
<td>0.466</td>
</tr>
<tr>
<td>RF</td>
<td>0.954</td>
<td>–</td>
<td>0.393</td>
<td>0.906</td>
<td>0.445</td>
<td>0.347</td>
<td>0.321</td>
</tr>
<tr>
<td>BRF</td>
<td>0.945</td>
<td>–</td>
<td>0.401</td>
<td>0.887</td>
<td>0.515</td>
<td>0.441</td>
<td>0.468</td>
</tr>
<tr>
<td>Under-RF</td>
<td>0.948</td>
<td>–</td>
<td>0.419</td>
<td>0.895</td>
<td>0.537</td>
<td>0.435</td>
<td>0.445</td>
</tr>
<tr>
<td>Over-RF</td>
<td>0.955</td>
<td>–</td>
<td>0.397</td>
<td>0.904</td>
<td>0.490</td>
<td>0.408</td>
<td>0.348</td>
</tr>
<tr>
<td>REA-Best</td>
<td>0.971</td>
<td>1</td>
<td>0.633</td>
<td>0.905</td>
<td>1</td>
<td>1</td>
<td>0.985</td>
</tr>
<tr>
<td>REA_Eqs(14), (15)</td>
<td>0.965</td>
<td>1</td>
<td>0.571</td>
<td>0.884</td>
<td>1</td>
<td>0.996</td>
<td>0.985</td>
</tr>
</tbody>
</table>
We also find that the precision for a dataset seems to correlate to the population scale and attribute numbers. Fig. 6 illustrates the relationship between the maximum F-measure value and the ratio of minority class number and attribute number. In the figure, the 14 points corresponding to the former 14 datasets in Table 1 (represented by circles) are used to fit a logistic curve, the equation of which is

\[ y = \frac{0.145}{1 + \left(\frac{x}{2.319}\right)^{2.307}} \]

It could be found that the other 6 points corresponding to the latter datasets in Table 1 (represented by triangles) are close to the curve. The figure shows that when the ratio of minority class number and attribute number is less than 3, the probability of obtaining a worse performance can be increased significantly.

### Table 7
G-mean comparison results.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Car</th>
<th>Water</th>
<th>Transfusion</th>
<th>Pima</th>
<th>Operative</th>
<th>Vehicle</th>
<th>Balance</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>CART</td>
<td>0.910</td>
<td>–</td>
<td>–</td>
<td>0.673</td>
<td>–</td>
<td>0.658</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Bagg</td>
<td>0.964</td>
<td>–</td>
<td>–</td>
<td>0.720</td>
<td>–</td>
<td>0.642</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Ada</td>
<td>0.980</td>
<td>–</td>
<td>–</td>
<td>0.694</td>
<td>–</td>
<td>0.664</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>Asym</td>
<td>0.981</td>
<td>–</td>
<td>–</td>
<td>0.696</td>
<td>–</td>
<td>0.679</td>
<td>0.002</td>
<td></td>
</tr>
<tr>
<td>SMB</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.719</td>
<td>–</td>
<td>0.728</td>
<td>0.002</td>
<td></td>
</tr>
<tr>
<td>Over</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.719</td>
<td>–</td>
<td>0.768</td>
<td>0.009</td>
<td></td>
</tr>
<tr>
<td>Under</td>
<td>0.956</td>
<td>–</td>
<td>–</td>
<td>0.692</td>
<td>–</td>
<td>0.657</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>SMOTE</td>
<td>0.969</td>
<td>–</td>
<td>–</td>
<td>0.708</td>
<td>–</td>
<td>0.743</td>
<td>0.465</td>
<td></td>
</tr>
<tr>
<td>Chain</td>
<td>0.970</td>
<td>–</td>
<td>–</td>
<td>0.700</td>
<td>–</td>
<td>0.738</td>
<td>0.465</td>
<td></td>
</tr>
<tr>
<td>Cascade</td>
<td>0.969</td>
<td>–</td>
<td>–</td>
<td>0.725</td>
<td>–</td>
<td>0.760</td>
<td>0.595</td>
<td></td>
</tr>
<tr>
<td>Easy</td>
<td>0.958</td>
<td>–</td>
<td>–</td>
<td>0.734</td>
<td>–</td>
<td>0.781</td>
<td>0.577</td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0.452</td>
<td>–</td>
<td>–</td>
<td>0.717</td>
<td>–</td>
<td>0.659</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>BRF</td>
<td>0.693</td>
<td>–</td>
<td>–</td>
<td>0.735</td>
<td>–</td>
<td>0.780</td>
<td>0.548</td>
<td></td>
</tr>
<tr>
<td>Under-RF</td>
<td>0.687</td>
<td>–</td>
<td>–</td>
<td>0.740</td>
<td>–</td>
<td>0.779</td>
<td>0.548</td>
<td></td>
</tr>
<tr>
<td>Over-RF</td>
<td>0.690</td>
<td>–</td>
<td>–</td>
<td>0.731</td>
<td>–</td>
<td>0.689</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>REA-Best</td>
<td>1</td>
<td>0.973</td>
<td>0.991</td>
<td>0.994</td>
<td>0.981</td>
<td>0.998</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>REA_Eqs. (14), (15)</td>
<td>1</td>
<td>0.962</td>
<td>0.989</td>
<td>0.986</td>
<td>0.981</td>
<td>0.998</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8
F-measure comparison results on the latter 6 datasets listed in Table 1.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Glass</th>
<th>Thyroid</th>
<th>Vertebral</th>
<th>Dermatology</th>
<th>Breast</th>
<th>Yeast</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>REA-Best</td>
<td>0.945</td>
<td>1</td>
<td>1</td>
<td>0.828</td>
<td>0.980</td>
<td>0.989</td>
<td>0.957</td>
</tr>
<tr>
<td>REA_Eqs. (14), (15)</td>
<td>0.897</td>
<td>1</td>
<td>1</td>
<td>0.754</td>
<td>0.980</td>
<td>0.981</td>
<td>0.935</td>
</tr>
<tr>
<td>Decrease (%)</td>
<td>5.079</td>
<td>0</td>
<td>0</td>
<td>8.880</td>
<td>0</td>
<td>0.847</td>
<td>2.468</td>
</tr>
</tbody>
</table>

### Table 9
G-means comparison results on the latter 6 datasets listed in Table 1.

<table>
<thead>
<tr>
<th>DataSet</th>
<th>Glass</th>
<th>Thyroid</th>
<th>Vertebral</th>
<th>Dermatology</th>
<th>Breast</th>
<th>Yeast</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>REA-Best</td>
<td>0.964</td>
<td>1</td>
<td>1</td>
<td>0.957</td>
<td>0.989</td>
<td>0.991</td>
<td>0.983</td>
</tr>
<tr>
<td>REA_Eqs. (14), (15)</td>
<td>0.938</td>
<td>1</td>
<td>1</td>
<td>0.920</td>
<td>0.989</td>
<td>0.985</td>
<td>0.972</td>
</tr>
<tr>
<td>Decrease (%)</td>
<td>2.697</td>
<td>0</td>
<td>0</td>
<td>3.919</td>
<td>0</td>
<td>0.636</td>
<td>1.209</td>
</tr>
</tbody>
</table>

Fig. 6. The relationship between the maximum F-measure value and the ratio of minority class number and attribute number.
3.4. Experiment for multi-class problems

Seven multi-classification problems in UCI dataset (listed in Table 2) are selected to test the proposed method for multi-class problems. In the experiment, kNN, Bayes, and BP networks are performed as the base learning algorithms separately and combined together. Therefore, we have four methods for comparison. Tables 10 and 11 list the comparison results of F-measure and G-mean of the experiment. As shown in Table 8, among the three separate algorithms, Bayes performs the best in 4 out of 7 data sets on F-measure, and BP obtains the best performance in the other 3 data sets and the best average as well. They evidently outperform kNN on F-measure. While on G-mean, BP obtains significantly better performance than both Bayes and kNN algorithms. Comparing the three separate algorithms and the combination of them, we could find that the combination obtains the best average on both F-measure and G-mean. It is 26.6%, 6.9%, and 11.1% better than kNN, BP and Bayes on the average F-measure, while on the average G-mean, the combination method outperforms the three separate algorithms even by 71.2%, 6.6%, and 84.1%, respectively. Numerical results show that the ensemble of different types of methods could improve the algorithm performance efficiently.

4. Conclusions

Focused on the classification problems of imbalanced datasets, this paper proposed a resampling ensemble algorithm. In the method, both oversampling for the “small” classes and undersampling for the “large” classes are performed, and the resampling scale is determined according to the ratio of the min class number and max class number. In our experiments, empirical analysis of resampling scale is performed, and empirical formulations have been derived to fix the parameters of resampling scale. Numerical results show that the algorithm performance is related to the size of classes and attribute number. We find that the relationship between the maximum F-measure value and the ratio of minority class number and attribute number can fit to a logistic curve very well. When the ratio is less than 3, the performance will be greatly hindered. Experimental results of the multiple class problems show that the ensemble of different types of methods could improve the algorithm performance efficiently.

Acknowledgments

The authors are grateful to the support of the National Natural Science Foundation of China (61272207, 61373050), and the Science-Technology Development Project of Jilin Province of China (201215022, 20130206003SF, and 20130101070JC).

References


Table 10
F-measure comparison results.

<table>
<thead>
<tr>
<th>DataSet</th>
<th>Car</th>
<th>Thyroid</th>
<th>Water</th>
<th>Verteral</th>
<th>Dermatopoly</th>
<th>Vehicle</th>
<th>Glass</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN</td>
<td>0.744</td>
<td>0.889</td>
<td>0.321</td>
<td>0.836</td>
<td>0.513</td>
<td>0.644</td>
<td>0.641</td>
<td>0.656</td>
</tr>
<tr>
<td>BP</td>
<td>0.790</td>
<td>0.918</td>
<td>0.482</td>
<td>0.765</td>
<td>0.907</td>
<td>0.795</td>
<td>0.783</td>
<td>0.777</td>
</tr>
<tr>
<td>Bayes</td>
<td>0.894</td>
<td>0.955</td>
<td>0.610</td>
<td>0.964</td>
<td>0.726</td>
<td>0.586</td>
<td>0.495</td>
<td>0.747</td>
</tr>
<tr>
<td>All</td>
<td>0.879</td>
<td>0.913</td>
<td>0.584</td>
<td>0.809</td>
<td>0.924</td>
<td>0.819</td>
<td>0.885</td>
<td>0.830</td>
</tr>
</tbody>
</table>

Table 11
G-mean comparison results.

<table>
<thead>
<tr>
<th>DataSet</th>
<th>Car</th>
<th>Thyroid</th>
<th>Water</th>
<th>Verteral</th>
<th>Dermatopoly</th>
<th>Vehicle</th>
<th>Glass</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN</td>
<td>0.729</td>
<td>0.836</td>
<td>0</td>
<td>0.829</td>
<td>0</td>
<td>0.590</td>
<td>0</td>
<td>0.426</td>
</tr>
<tr>
<td>BP</td>
<td>0.757</td>
<td>0.945</td>
<td>0</td>
<td>0.758</td>
<td>0.880</td>
<td>0.784</td>
<td>0.665</td>
<td>0.685</td>
</tr>
<tr>
<td>Bayes</td>
<td>0.856</td>
<td>0.928</td>
<td>0</td>
<td>0.957</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.397</td>
</tr>
<tr>
<td>All</td>
<td>0.844</td>
<td>0.928</td>
<td>0</td>
<td>0.801</td>
<td>0.912</td>
<td>0.807</td>
<td>0.818</td>
<td>0.730</td>
</tr>
</tbody>
</table>


Yun Qian received the Master Degree in 2005 from the Department of Computer Science and Engineering in Dalian University of Technology, China. She is currently a Ph.D. candidate in the College of Computer Science and Technology, Jilin University, China. Her research interests include data mining and machine learning.

Yanchun Liang received the Ph.D. degree in applied mathematics from Jilin University, Changchun, China, in 1997. He is currently a Professor in the College of Computer Science and Technology, Jilin University, China. He was a visiting scholar in Manchester University of U.K. from 1990 to 1991, a visiting professor in National University of Singapore from 2000 to 2001, a guest professor in Institute of High Performance Computing of Singapore from 2002 to 2004, and a guest professor in Trento University, Italy from 2006 to 2008. He has published more than 300 papers. He was the recipient of several grants from NSFC, EU, etc. His research interests include computational intelligence, machine learning methods, text mining, MEMS modeling and bioinformatics.

Mu Li received the Bachelor Degree in 2011 from the College of Mathematics and Applied Mathematics in Jilin University, China. He is currently a graduate student in the College of Computer Science at Jilin University. His research interests include Complex Networks and Bioinformatics.

Guoxiang Feng received the Bachelor Degree in 2012 from the College of Computer in Jilin Normal University, China. She is currently a professor in the College of Computer Science and Technology, Jilin University, China. Her research interests include bioinformatics, computational intelligence.

Xiaohu Shi received the Ph.D. degree in computer application technology from Jilin University, Changchun, China, in 2006. He is currently an associate professor in the College of Computer Science and Technology, Jilin University, China. He has published more than 50 journal and conference papers. His current research interests include machine learning and bioinformatics.