Neurocomputing 143 (2014) 57–67

Contents lists available at ScienceDirect

Neurocomputing

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

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.

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

* Corresponding author. Tel.: +86 431 85153829; fax: +86 431 85168752.
E-mail addresses: qianyun_116@163.com (Y. Qian), ycliang@jlu.edu.cn (Y. Liang), lianquezhi@gmail.com (M. Li), fengguoxiang0418@163.com (G. Feng), shixh@jlu.edu.cn (X. Shi).

http://dx.doi.org/10.1016/j.neucom.2014.06.021
0925-2312/

random undersampling to the majority class, but SMOTEBoost creates synthetic new minority class instances by operating in the "feature space" [15]. Błaszczynski et al. integrate a selective data pre-processing method SPIDER with Ivotes ensemble algorithm developing the framework called IIvotes [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 data stream classification, Ghazikhani proposed an online ensemble of neural networks which is a two layer approach [8].

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 $D=\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2),\ldots, (\boldsymbol{x}_m, y_m)\}$, where $\boldsymbol{x}_i \in X$ and $y_i \in Y=\{1,2,\ldots L\}$ $(i=1,2,\ldots,m)$. 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 ith class as n_i and assume $A=n_1 \le n_2 \le \ldots \le n_L=B$, then we have $\sum n_i=m$. The resampling scale ratio of the lth class S_l could be calculated according to Eq. (1):

$$S_l=\frac{\beta B-\alpha A}{B-A}+\frac{(\alpha-\beta)AB}{(B-A)n_l} \tag{1}$$

where α $(\alpha \ge 1)$ is the scale parameter of the smallest class and β $(0 \le \beta \le 1)$ 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 lth class re_num_l is

$$re\_num_l=\mathrm{Round}(S_l \cdot n_l) \tag{2}$$

where $\mathrm{Round}(\cdot)$ is the round function. Then in the tth bootstrap step, for a majority class, say the ith class, we just need to randomly select re_num_i samples from the original n_i samples into D_t; while for a minority class, say the jth calss, besides putting all the original n_j samples into D_t, we also need to produce $(re\_num_i-n_j)$ new samples according to SMOTE technique. Each new sample is produced as follows: Randomly select a sample in class l, $(\boldsymbol{x}_i, y_i)$, and then compute K nearest neighbors of the sample, at last generate a new sample of class l, by averaging the K nearest neighbors with randomly weights. Fig. 2 gives the pseudo code of the method, and Fig. 3 illustrates the "Bootstrap algorithm" sub-process. In the framework of resampling ensemble algorithm, different learning algorithms are used to train different base learners.

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 naïve 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 $[2, sqrt(n)]$. 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