Neurocomputing 000 (2017) 1–11

Contents lists available at ScienceDirect

Neurocomputing

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

A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data

Guillem Collell[a,d], Drazen Prelec[a,b,c], Kaustubh R. Patil[a,e,*]

[a] MIT Sloan Neuroeconomics Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] Department of Economics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[c] Brain & Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[d] Computer Science Department, KU Leuven, Heverlee 3001, Belgium
[e] Institute of Neuroscience and Medicine, Brain & Behaviour (INM-7), Research Centre Jülich, Jülich 52425, Germany

ARTICLE INFO

Article history:
Received 10 October 2016
Revised 26 July 2017
Accepted 29 August 2017
Available online xxx

Communicated By Dr. V. Palade

Keywords:
Imbalanced data
Binary classification
Multiclass classification
Bagging ensembles
Resampling
Posterior calibration

ABSTRACT

Class imbalance presents a major hurdle in the application of classification methods. A commonly taken approach is to learn ensembles of classifiers using rebalanced data. Examples include bootstrap averaging (bagging) combined with either undersampling or oversampling of the minority class examples. However, rebalancing methods entail asymmetric changes to the examples of different classes, which in turn can introduce their own biases. Furthermore, these methods often require specifying the performance measure of interest *a priori*, i.e., before learning. An alternative is to employ the threshold moving technique, which applies a threshold to the continuous output of a model, offering the possibility to adapt to a performance measure *a posteriori*, i.e., a *plug-in* method. Surprisingly, little attention has been paid to this combination of a bagging ensemble and threshold-moving. In this paper, we study this combination and demonstrate its competitiveness. Contrary to the other resampling methods, we preserve the *natural* class distribution of the data resulting in well-calibrated posterior probabilities. Additionally, we extend the proposed method to handle multiclass data. We validated our method on binary and multiclass benchmark data sets by using both, decision trees and neural networks as base classifiers. We perform analyses that provide insights into the proposed method.

1. Introduction

Dealing with a class imbalance in classification is an important problem that poses major challenges [1]. Imbalanced data sets frequently appear in real-world problems, such as in fault and anomaly detection [2,3], fraudulent phone call detection [4] and medical decision-making [5], to name a few. Standard learning algorithms are often guided by global error rates and hence may ignore instances of the minority class, leading to models biased towards predicting the majority class. Several methods have been proposed to alleviate this problem (see, e.g., [6,7] for reviews). Often, a first choice consists of preprocessing the data by resampling to balance the class distribution [8,9]. This is often achieved by either randomly oversampling (ROS) the minority class [9] or randomly undersampling (RUS) the majority class [10]. More sophisticated methods that generate synthetic minority class instances are also a popular choice, e.g., the synthetic minority oversampling technique (SMOTE [9]). We will collectively call these data preprocessing methods as rebalancing mechanisms as they, in general, aim to make the training data more balanced. This will also avoid confusion with other resampling mechanisms, e.g., the simple bootstrap. Rebalancing is often combined with ensembles as they show superior performance to a single classifier [11]. Many such combinations have been shown to be effective for imbalanced data classification [6,12,13]. However, there are several potential drawbacks of rebalancing methods: (1) potential loss of informative data when undersampling, (2) changes in the properties of the data, such as asymmetric changes in the density of examples of different classes, which in turn can cause the models to induce unwanted biases, e.g., miscalibrated posterior probability estimates [14,15], (3) it is often not evident which class distributions to use for a given dataset and a performance measure of interest [16] (wrapper methods [17] can be employed to tune the model for a given measure, but they are computationally expensive and often cater towards only a single measure, e.g., either accuracy or

* Corresponding author at: MIT Sloan Neuroeconomics Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
E-mail addresses: gcollell@kuleuven.be (G. Collell), dprelec@mit.edu (D. Prelec), kaustubh.patil@gmail.com (K.R. Patil).

http://dx.doi.org/10.1016/j.neucom.2017.08.035

F1-score), and (4) it is nontrivial to extend the sampling heuristics normally defined for binary data to multiclass data as there can be multiple minority/majority classes [18].

Moving decision thresholds is another technique to deal with class imbalance. The main difference between rebalancing and threshold-based methods is that the former relies on data preprocessing before learning happens, whereas the latter relies on manipulating the continuous output of a learned model, e.g., class weights or posterior probabilities. Among other proponents, Provost [19] advocated for threshold-moving as a method to deal with class imbalance. Nevertheless, surprisingly, little attention has been paid to this technique, often to an extent that it is not even considered for comparison when new methods are proposed.

While this technique has been utilized in combination with some popular learning methods including a small ensemble [19–21]. However, to our knowledge, the combination of threshold-moving with a bagging ensemble has not been thoroughly investigated. As is evident, threshold-moving depends on reliable continuous estimation of the output; therefore, bagging ensembles are a good candidate to combine with threshold-moving as they are known to provide good probability estimates [22,23]. In this work, we study threshold-moving combined with bagging ensembles and show that it is a competitive method with several advantages.

In particular, we seek a method that provides well-calibrated posterior probability estimates. An important advantage of such a method is that it can be utilized as a plug-in method where the threshold can be set *a posteriori*, i.e., at the test phase. This provides an opportunity to achieve good performance on different measures using the same model [24]. This is a major improvement over other methods, e.g., cost-sensitive methods and rebalancing, which require the performance measure of interest to be specified at the learning phase. Here, we propose *Probability Threshold bagging* (PT-bagging) that, as we will show, passes as a *plug-in* method. The main motivation behind PT-bagging is to leverage the advantages of bagging while avoiding the problems that rebalancing methods inevitably entail, as described above. The proposed method PT-bagging addresses those problems and possesses several desirable properties:

(1) It is a plug-in method that maximizes a performance measure of interest without retraining, but rather by just applying an appropriate threshold *a posteriori*. By contrast, rebalancing methods are not flexible and need computationally expensive parameter tuning, e.g., to find which class proportions to use for learning via a wrapper approach [17].
(2) It consistently performs close to the best possible macro-accuracy and macro F1 performances without the need to empirically find the optimal threshold (e.g., by cross-validation). Obtaining a validation set for tuning can be computationally costly, might not always be possible, or might be financially prohibitive (e.g., due to data collection costs).
(3) It can be extended to handle the multiclass setting when appropriate thresholds for a performance measure of interest are available, e.g., macro-accuracy.

We provide a theoretical analysis on when optimal macro-accuracy performance is guaranteed. However, for other measures, such as the macro F1-score, it is not always possible to obtain a closed-form expression for the optimal thresholds [25]. Nevertheless, we show that our new, simple and sensible threshold is close to the optimal threshold, and that PT-bagging achieves higher macro F1-score performance compared to other methods. In this respect, we make two additional contributions: (1) the proposal of a threshold for maximizing the macro F1-score, and (2) a comparison and analysis of the *full potential* of the methods, which we define as their maximum attainable performance if the optimal threshold were known.

The rest of this paper is organized as follows: in Section 2 we provide the relevant background, describe some popular resampling methods, and discuss their potential flaws. In Section 3, we describe our proposed method, PT-bagging, and provide a theoretical justification of its performance. In Section 4, we describe our experimental setup. In Section 5, we present a comprehensive set of empirical tests and discuss the results. Finally, we comment on the implications of our findings and propose future lines of research.

2. Background

We consider the standard classification setting where a learning algorithm learns from the training data tuples $\{x_i, y_i\}_{i=1}^{N}$, where $x_i \in X$ are features that can be either continuous, ordinal or categorical and $y_i \in C = \{1, \ldots, m\}$ are discrete class labels. The goal of learning is to estimate a predictor $\hat{f}: X \to C$ that approximates the true underlying function f: $X \to C$. The model learned, $\hat{f}$, is then used to make predictions on unseen test data $\{x_j\}_{j=1}^{M}$. For binary data, we have $y_i \in \{0, 1\}$ and without loss of generality we denote the minority class (i.e., the class with lower frequency in the training data) as the class 1. We refer to the class-specific thresholds as λ_i, $i = 1, \ldots, m$. Their application to the classifier output is described below (Algorithm 1, step 2.4). We make two assumptions: (1) the probability distribution of the test data is similar to that of the training data, and (2) the class distribution of the training data provides an accurate estimate of their respective underlying prior probabilities.

2.1. Performance measures for imbalanced data

The commonly used measure of accuracy (correct classification rate) is a good metric when data sets are balanced. However, it can be misleading for imbalanced data. For example, the naïve strategy of classifying all the examples into the majority class would obtain 99% accuracy in a data set composed of 99% examples of this class. Therefore, other measures are necesary when dealing with imbalanced data.

Several performance measures have been proposed in imbalanced learning, all of which are computable from the elements of the confusion matrix (Table 1). Some of the most extensively used measures are:

$$\text{TNR} = \frac{\text{TN}}{\text{TN}+\text{FP}};\ \text{Recall}\ (=\ \text{TPR})\ = \frac{\text{TP}}{\text{TP}+\text{FN}};$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}};\ \text{FPR}\ = \frac{\text{FP}}{\text{TN}+\text{FP}}$$

$$\text{Macro}-\text{accuracy}\ = \frac{\text{TPR}\ +\text{TNR}}{2};\ \text{G}-\text{mean}\ = \sqrt{\text{TPR}\ \times\ \text{TNR}};$$

$$\text{F1}-\text{score}\ = \frac{2\ \times\ \text{Precision}\ \times\ \text{Recall}}{\text{Precision}+\text{Recall}} = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$$

The macro F1-score is a widely used measure and is calculated by considering each class separately as the positive class and then averaging their corresponding F1-scores. In addition, the receiver operating characteristic (ROC) curve is often employed [6]. The ROC curve is generated by plotting the TPR (y-axis) and the FPR (x-axis) while moving through the whole spectrum of decision thresholds.

Table 1
Confusion matrix in binary classification.

	Predicted positive	Predicted negative
Actual positive	TP (true positive)	FN (false negative)
Actual negative	FP (false positive)	TN (true negative)

Please cite this article as: G. Collell et al., A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.035