Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Finding the Best Classification Threshold in Imbalanced Classification ☆

Quan Zou [a,b], Sifa Xie [b], Ziyu Lin [b], Meihong Wu [b], Ying Ju [b]

[a] School of Computer Science and Technology, Tianjin University, Tianjin, China
[b] Department of Computer Science, Xiamen University, Xiamen, China

ARTICLE INFO

Article history:
Received 26 August 2015
Received in revised form 22 October 2015
Accepted 1 December 2015
Available online xxxx

Keywords:
Receiver Operating Characteristic (ROC)
Protein remote homology detection
Imbalance data
F-score

ABSTRACT

Classification with imbalanced class distributions is a major problem in machine learning. Researchers have given considerable attention to the applications in many real-world scenarios. Although several works have utilized the area under the receiver operating characteristic (ROC) curve to select potentially optimal classifiers in imbalanced classifications, limited studies have been devoted to finding the classification threshold for testing or unknown datasets. In general, the classification threshold is simply set to 0.5, which is usually unsuitable for an imbalanced classification. In this study, we analyze the drawbacks of using ROC as the sole measure of imbalance in data classification problems. In addition, a novel framework for finding the best classification threshold is proposed. Experiments with SCOP v.1.53 data reveal that, with the default threshold set to 0.5, our proposed framework demonstrated a 20.63% improvement in terms of F-score compared with that of more commonly used methods. The findings suggest that the proposed framework is both effective and efficient. A web server and software tools are available via http://datamining.xmu.edu.cn/prht/ or http://prht.sinaapp.com/.

1. Background

A dataset is imbalanced if it contains a small amount of samples in one class as compared with the rest of the classes. Without loss of generality, a minority class is regarded as a positive class, whereas a majority class is viewed as a negative class. Imbalanced classification is one of most popular topics in the field of machine learning [1–4]. This issue is represented in many real-world applications, such as bioinformatics [5–11], telecommunications management [12], text classification [13], face recognition [14], and ozone level forecasting [15]. Traditional classifications algorithms perform poorly on imbalanced datasets because the applied evaluation metrics, such as the overall accuracy metric, force classifiers to minimize the error rate, i.e., the percentage of the incorrect prediction of class labels. As a result, classifiers demonstrate good accuracy on the majority class but poor accuracy on the minority class. However, in most imbalanced classification problems, the misclassification error of the minority class is far costlier than that of the majority class. For example, in the medical diagnosis of a certain cancer, misclassifying a cancer patient as healthy is more serious than misclassifying a non-cancer patient as unhealthy, because, in the former, the patient might lose his/her life.

Table 1
Confusion matrix for binary classification.

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

$\text{TPrate} = \frac{TP}{TP+FN}$, the percentage of positive samples correctly classified.
$\text{TNrate} = \frac{TN}{FP+TN}$, the percentage of negative samples correctly classified.
$\text{FPrate} = \frac{FP}{FP+TN}$, the percentage of negative samples misclassified.
$\text{FNrate} = \frac{FN}{TP+FN}$, the percentage of positive samples misclassified.

As previously mentioned, in imbalanced domains, a specific metric is needed to evaluate the performance of the classifier. The receiver operating characteristic (ROC) graphic [16–19] is commonly used as an evaluation criterion. For a binary classification, we can obtain a confusion matrix, as shown in Table 1, and based on which, four metrics can be calculated.

The ROC graphic depicts the trade-off between benefits (TPrate) and costs (FPrate); in other words, one classifier cannot increase the number of true positives without increasing the false positives. In a ROC curve, the x-axis represents the FPrate and the y-axis represents the TPrate. The points in the curve are obtained by sweeping the classification threshold from the most positive classification value to the most negative. The area under the ROC curve (AUC) [20] is a useful metric for classifying performance because it gives the probability that a randomly selected pair of samples (one

☆ This article belongs to Analytics and Applications.

E-mail address: yju@xmu.edu.cn (Y. Ju).

http://dx.doi.org/10.1016/j.bdr.2015.12.001
2214-5796/

positive and one negative) would have their predicted probabilities correctly ordered.

In imbalanced classification domains, ROCs are considered the "gold standard" of a classifier's ability. However, using only the ROC to select a potentially optimal classifier is not enough. In fact, the ROC curve and the AUC values reflect only the ranking power of positive prediction probability. Furthermore, a high AUC does not insure a high prediction accuracy. For example, in a dataset containing only 1% positive samples, the AUC value can reach more than 0.9 only if all the positive samples rank in the top 10% according to prediction probability. Even if the probabilities of positive samples are predicted to be less than 0.5, as long as the positive probabilities exceed the negative ones, the ROC will exhibit good performance. This phenomenon is typical in imbalanced datasets. Therefore, finding an appropriate prediction probability threshold is as important as a perfect ROC curve for the accurate prediction of testing and unknown data. In most classifiers, the default prediction probability threshold is 0.5. However, this threshold does not work well for imbalanced classification prediction.

Although researchers have attempted to raise the AUC value in previous works, these investigations disregarded the prediction probability thresholds for testing and unknown data. Consequently, classification performance, including recall, precision, and F-scores, remains imperfect even if the AUC value could become rather high. Few tools or Web servers are available for finding the classification threshold. In this paper, we propose a sampling-based threshold auto-tuning method to address this problem. This method can obtain perfect performance on the AUC criteria in addition to very good precision, recall, and F-scores.

2. Methods

2.1. F-score should be another metric aside from the AUC

The AUC is often considered a reliable performance metric for imbalanced binary classification problems [21–24]. However, when the dataset is imbalanced and the AUC has reached a high score, the classification performance may not be as perfect as the AUC value reflects because plenty of "trash" negative samples exist in the imbalanced dataset. "Trash" negative samples raise the AUC value, but a few other negative samples remain mixed with the positive samples, which are difficult to distinguish. These few remaining negative samples diminish performance, including precision and recall, while very slightly influencing the AUC value. In the testing dataset, the values of precision and recall may be less than 0.5, whereas the AUC value can exceed 0.9. AUC50 was proposed to address this problem and to measure the performance of protein remote homology detection [25] (Fig. 1). The AUC50 refers to the AUC up to the first 50 false positive samples. Although the AUC50 can avoid the influence of "trash" true negative samples, 50 is overly arbitrary for various datasets. If less than 50 true negative samples exist in the dataset, then the AUC50 is equal to the AUC. Furthermore, if the training samples are massive, and 50 false positive samples account for only a very small portion of the training set, the AUC50 would be meaningless. Therefore, even though the AUC50 can often better describe classification performance than the AUC, it cannot alleviate the problem inherent to massive data. Thus, we need a different metric altogether along with the AUC to measure classification performance.

We attempt to employ the F-score together with the AUC as a classification measurement for protein remote homology detection. The F-score is a trade-off between precision (P) and recall (R) and is described as follows:

$$P = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}; \tag{1}$$

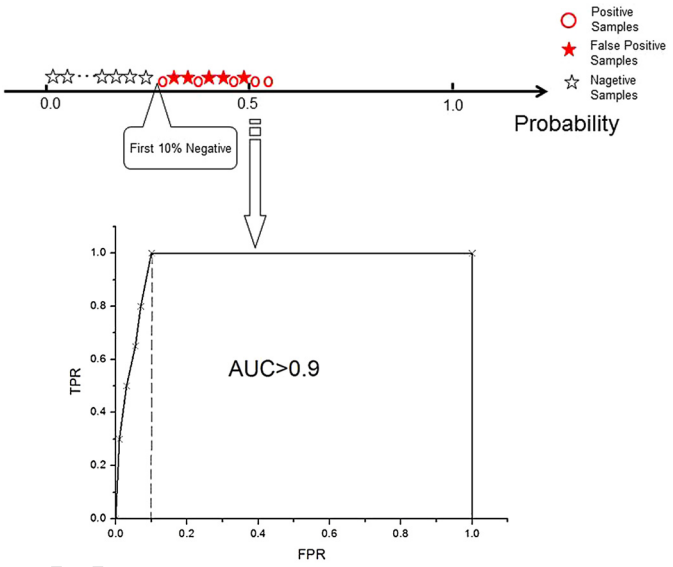

Fig. 1. The ROC of an imbalanced dataset.

$$R = \frac{\text{TruePositive}}{\text{Truepositive} + \text{FalseNegative}}; \tag{2}$$

and

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{3}$$

where β is a parameter used to adjust the weight between P and R.

2.2. How to set the classification threshold for the testing set

Prediction results are ultimately determined according to prediction probabilities. The threshold is typically set to 0.5. If the prediction probability exceeds 0.5, the sample is predicted to be positive; otherwise, negative. However, 0.5 is not ideal for some cases, particularly for imbalanced datasets.

The probability threshold for classification will not interfere with the AUC value. In other words, the AUC is influenced by the probability ranking result only, and it is not related to the setting of the classification threshold. Therefore, we only need to tune the threshold to obtain the best F-score.

The threshold for the best F-score can be easily obtained if the training set is not massive. We can test all of the probabilities for every positive sample with a brute-force attack. Then, the threshold with the best F-score for the training set can be calculated by using cross-validation [26]. We then determine if the calculated threshold can be used for the test data.

We observed that Liao's protein remote homology detection dataset was not massive enough. The prediction probabilities are distributed differently between the training and testing sets. Moreover, the probability ranges are considerably different, as shown in Fig. 2. We posit that the best threshold position in the training set should be mapped to the corresponding position in the testing set.

We denote the maximum prediction probability in the training set as Maxtrain. In this paper, prediction probability refers to the probability of positive predictions by the classifier. If the prediction probability is less than the threshold, the sample is predicted to be negative. Similarly, we also denote Mintrain, Maxtest, Mintest. Thresholdtrain, and Thresholdtest. Thus, the mapping rule should satisfy the following equation, from which we can compute the Thresholdtest: