Knowledge-Based Systems 25 (2012) 13–21

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

On the effectiveness of preprocessing methods when dealing with different levels of class imbalance

V. García, J.S. Sánchez *, R.A. Mollineda

Institute of New Imaging Technologies, Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain

ARTICLE INFO

Article history:
Available online 26 June 2011

Keywords:
Imbalance
Resampling
Classification
Performance measures
Multi-dimensional scaling

ABSTRACT

The present paper investigates the influence of both the imbalance ratio and the classifier on the performance of several resampling strategies to deal with imbalanced data sets. The study focuses on evaluating how learning is affected when different resampling algorithms transform the originally imbalanced data into artificially balanced class distributions. Experiments over 17 real data sets using eight different classifiers, four resampling algorithms and four performance evaluation measures show that over-sampling the minority class consistently outperforms under-sampling the majority class when data sets are strongly imbalanced, whereas there are not significant differences for databases with a low imbalance. Results also indicate that the classifier has a very poor influence on the effectiveness of the resampling strategies.

1. Introduction

Class imbalance constitutes one of the problems that has recently received most attention in research areas such as Machine Learning, Pattern Recognition, Data Mining, and Knowledge Discovery. A two-class data set is said to be imbalanced if one of the classes (the minority one) is represented by a very small number of instances in comparison to the other (majority) class [1]. Besides, the minority class is usually the most important one from the point of view of the learning task. It has been observed that class imbalance may cause a significant deterioration in the performance attainable by standard learners because these are often biased towards the majority class [2,3]. These classifiers attempt to reduce global measures such as the error rate, not taking the data distribution into consideration. This issue is especially important in real-world applications where it is often costly to misclassify examples of the minority class, such as diagnosis of infrequent diseases [4], fraud detection in mobile telephone communications [5,6] or credit cards [7], detection of oil spills in satellite radar images [8], text categorization [9,10], credit assessment [11], prediction of customer insolvency [12], and translation initiation site recognition in DNA sequences [13]. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases respectively, they are also known as positive and negative examples.

Main research on this topic can be categorized into three groups. One has primarily focused on the implementation of solutions for handling the imbalance, both at the data and algorithmic levels [14–16]. Another group has addressed the problem of measuring the classifier performance in imbalanced domains [17,18]. The third has been to analyse what data complexity characteristics aggravate the problem and even, to study whether there exist other factors that lead to a loss of classifier performance or it is the imbalance problem per se that causes the performance decrease [19,20].

Among the most investigated issues, one can find both algorithmic and data level solutions. Examples of the former are approaches to internally biasing the discriminating process [14] and multi-experts systems [21], whereas the data level solutions consist of artificially resampling the original data set until the problem classes are approximately equally represented. Conclusions about what is the best data level solution for the class imbalance problem are divergent. In this sense, Hulse and Khoshgoftaar [22] suggest that the utility of each particular resampling technique depends on various factors, including the ratio between positive and negative examples, the characteristics of data, and the nature of the classifier. Other papers [2,23–25] have also studied this dependence during the last decade. Nevertheless, their conclusions should be carefully interpreted because most of them are based on narrow learning frameworks.

In many ways, this paper significantly extends previous works by increasing the scope and detail at which it is studied the influence of the imbalance ratio and the classifier on the effectiveness of the most popular resampling strategies (under and over-sampling). To this end, we will carry out a collection of experiments over 17 real databases with two different levels of imbalance, employing

* Corresponding author.
E-mail addresses: jimenezv@uji.es (V. García), sanchez@uji.es (J.S. Sánchez), mollined@uji.es (R.A. Mollineda).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.06.013

eight classifiers, four resampling techniques and four performance metrics.

The rest of the paper is organized as follows. Section 2 reviews several resampling techniques for problems with imbalanced data sets. Section 3 surveys a number of common performance evaluation measures, which can be especially useful for class imbalance. Next, in Section 4 the experimental set-up is described. Section 5 reports the results and discusses the most important findings. Finally, Section 6 remarks our conclusions and outlines possible directions for future research.

2. Data-driven methods for balancing the class distributions

Resampling techniques aim at correcting problems with the distribution of a data set [26]. Weiss and Provost [27] noted that in many real applications the original distribution of samples is not always the best distribution to use for a given classifier, and different resampling approaches try to modify the "natural" distribution to another that is closer to the optimal one. This can be accomplished either by over-sampling the minority class, by under-sampling the majority class, or by combining simple over and under-sampling techniques in a systematic manner [25,28]. All these strategies can be applied to any learning system, since they act as a preprocessing phase, allowing the learning system to receive the training instances as if they belonged to a well-balanced data set. Thus any bias of the system towards the majority class due to the different proportion of examples per class would be expected to be eliminated.

While these methods can result in greatly improved results over the original data set, they have also shown several important drawbacks. Under-sampling techniques may throw out potentially valuable data, whereas over-sampling artificially increases the size of the data set and consequently, worsens the computational burden of the learning algorithm. On the other hand, both under and over-sampling modify the prior probability of classes, and both lead to a decrease in the accuracy of the negative class.

Effectiveness of these resampling approaches has been analysed in previous studies with respect to different sources of data complexity and classification models. However, most of them have focused on some particular learning factors (classifiers, data sets, performance metrics, resampling strategies), but disregarding the effect of others.

- Japkowicz and Stephen [2] discussed the performance of basic resampling methods when using a C5.0 decision tree induction system over a reduced number of artificial and real-world data sets. The error rate on each class was recorded to carry out this study.
- Barandela et al. [24] presented an empirical comparison of several under and over-sampling techniques based on intelligent heuristics. The experiments were constrained to five real data sets using the nearest neighbour rule for classification and the geometric mean as the performance evaluation metric.
- Estabrooks et al. [25] studied the behaviour of random strategies at different resampling rates with C4.5 classifiers. They evaluated the performance on seven artificial and five real data sets by means of the overall error rate and the error on each class.
- Batista et al. [23] conducted a broad experimental analysis with 13 databases and 10 resampling methods, but conclusions were limited to the C4.5 decision tree and the use of the area under the ROC curve for assessing the results.

2.1. Over-sampling

The simplest method to increase the size of the minority class corresponds to random over-sampling, that is, a non-heuristic method that balances the class distribution through the random replication of positive examples. This contributes to balance the class distribution without adding new information to the data set. Nevertheless, since this method replicates existing positive examples, overfitting is more likely to occur.

Instead of simply duplicating original examples, Chawla et al. [15] proposed an over-sampling technique that generates new synthetic minority instances by interpolating between preexisting positive examples that lie close together. This method, called SMOTE (Synthetic Minority Over-sampling TEchnique), allows the classifier to build larger decision regions that contain nearby instances from the minority class.

From the original SMOTE algorithm, several modifications have further been proposed in the literature. For example, SMOTEBoost is an approach introduced by Chawla et al. [29] that combines SMOTE with the standard boosting procedure. García et al. [30] developed three variants based upon the concept of surrounding neighbourhood with the aim of taking both proximity and spatial distribution of the instances into consideration. Han et al. [31] presented the Borderline-SMOTE algorithm, which only creates new minority examples based on existing instances that are near the decision border. On the other hand, MSMOTE [32] not only considers the distribution of minority instances but also rejects latent noise based on the k-nearest neighbour classifier. Hongyu and Herna [33] introduced the DataBoost-IM method, which combines boosting and data generation.

2.2. Under-sampling

Random under-sampling [2,34] aims at balancing the data set through the random removal of negative examples. Despite important information can be lost when examples are discarded at random, it has empirically been shown to be one of the most effective resampling methods.

Unlike the random approach, many other proposals are based on a more intelligent selection of the negative examples to be eliminated. For example, Kubat and Matwin [35] proposed the one-sided selection (OSS) technique, which selectively removes only those negative instances that either are redundant or that border the minority class examples (the authors assume that these bordering cases are noise). The border examples were detected using the concept of Tomek links [36], while the redundant ones were eliminated by means of Hart's condensing [37].

In contrast to the one-sided selection technique, the so-called neighbourhood cleaning rule [38] emphasizes more data cleaning than data reduction. To this end, Wilson's editing [39] is used to identify and remove noisy negative instances. Similarly, Barandela et al. [14] introduced a method that eliminates not only noisy instances of the majority class by means of Wilson's editing (WE), but also redundant examples through the modified selective subset (MSS) condensing algorithm [40].

On the other hand, Yen et al. [41] presented a cluster-based under-sampling algorithm. It first clusters all the original examples into some clusters, and then selects an appropriate number of majority class samples from each cluster by considering the ratio of the number of majority class samples to the number of minority class samples in the cluster. García and Herrera [42] proposed the use of evolutionary computation algorithms to under-sample the majority class. Chen et al. [43] introduced a method based on pruning support vectors of the majority class.

3. Performance metrics for imbalanced class distributions

Evaluation of classification performance plays a critical role in the design of a learning system and therefore, the use of an appropriate measure becomes as important as the selection of a good algorithm to successfully tackle a given problem. Traditionally,