Knowledge-Based Systems 67 (2014) 342–360

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Boosting instance selection algorithms ☆

Nicolás García-Pedrajas *, Aida de Haro-García

Department of Computing and Numerical Analysis of the University of Córdoba, Campus Universitario de Rabanales, 14071 Córdoba, Spain

ARTICLE INFO

Article history:
Received 5 September 2013
Received in revised form 10 March 2014
Accepted 14 April 2014
Available online 9 May 2014

Keywords:
Instance selection
Boosting
Classifier ensembles
Data mining
Large datasets

ABSTRACT

Instance selection is one of the most important preprocessing steps in many machine learning tasks. Due to the increasing size of the problems, removing useless, erroneous or noisy instances is frequently an initial step that is performed before other data mining algorithms are applied. Instance selection as part of this data reduction task is one of the most relevant problems in current data mining research.

Over the past decades, many different instance selection algorithms have been proposed, each with its own strengths and weaknesses. However, as in the case of classification, it is unlikely that a single instance selection algorithm would be able to achieve good results across many different datasets and application fields. In classification, one of the most successful ways of consistently improving the performance of a single learner is the construction of ensembles using boosting methods. In this paper, we propose a novel approach for instance selection based on boosting instance selection algorithms in the same way boosting is applied to classification.

The proposed approach opens a new field of research in which to apply the many techniques developed for boosting classifiers, for instance selection and other data reduction techniques such as feature selection and simultaneous instance and feature selection. Using 60 datasets for balanced problems and 45 datasets for class-imbalanced problems, the experiments reported show a clear improvement in several state-of-the-art instance selection algorithms using the proposed methodology.

1. Introduction

The overwhelming amount of data that are currently available in any field of research poses new problems for data mining and knowledge discovery methods. This huge amount of data makes most existing algorithms inapplicable to many real-world problems. For these very large datasets, scalability becomes an issue. One of the most common ways of dealing with huge amounts of information is data reduction. Among existing data reduction techniques, one of the most popular is instance selection, which consists of removing missing, redundant and/or erroneous instances from a training set to obtain a tractable amount of data.

Instance selection [1,2] chooses a subset of the total available data to achieve the original purpose of the data mining application as if the whole data were being used. Different variants of instance selection exist. We can identify two main models [3]: instance selection as a method for prototype selection for algorithms based on prototypes (such as *k*-nearest neighbors) and instance selection for obtaining the training set for a learning algorithm (such as decision trees or neural networks). Although the proposed model could be applied to both tasks, in this paper we are mainly concerned with the problem of prototype selection.

The problem of instance selection for instance-based learning can be defined as "the isolation of the smallest set of instances that enable us to predict the class of a query instance with the same (or higher) accuracy than the original set" [4].

Another common task in data mining is classification. A classification problem of K classes and N training observations consists of a set of instances whose class membership, label, is known. Each label is an integer from the set $Y = \{1, \ldots, K\}$. A multi-class classifier is a function that maps an instance to an element of Y. The task is to find a definition for the unknown function that correctly maps each instance to its label. In a

☆ This work has been financed in part by Project TIN-2011-22967 of the Spanish Ministry of Science and Innovation and Project P09-TIC-4623 of the Junta de Andalucía.

* Corresponding author. Tel.: +34 957211032; fax: +34 957218630.
E-mail addresses: npedrajas@uco.es (N. García-Pedrajas), adeharo@uco.es (A. de Haro-García).
URL: http://cibrg.org (N. García-Pedrajas).

http://dx.doi.org/10.1016/j.knosys.2014.04.021
0950-7051/

classifier ensemble framework, we have a set of classifiers instead of just one, each classifier mapping an instance vector to the set of labels. One of the most successful methodology for constructing ensembles of classifiers is boosting.

As a general rule, boosting methods iteratively construct an ensemble of classifiers by modifying the distribution of instances in the dataset. A weight vector, $\mathbf{w}$, is used. w_i is a measure of how difficult the accurate classification of the instance $\mathbf{x}_i$ is. Initially, all instances receive the same weight, $w_i = 1/N$. Along the boosting process weights are adapted by increasing the values of incorrectly classified instances and decreasing the values of correctly classified instances. Many different boosting methods exist, which differ in, among other aspects, the way $\mathbf{w}$ is updated. Once the process is finished and the M classifiers are constructed, a final ensemble of classifiers is obtained:

$$F(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m f_m(\mathbf{x}), \tag{1}$$

where α_m is a weight associated with the m classifier. This weight depends on the achieved classification accuracy of f_m. In some boosting methods, all classifiers are equally weighted.

In this paper, we propose that instance selection may be approached as a classification problem of two classes. Given a training instance, the instance can be classified into one of two classes: "selected" or "unselected". With this view of instance selection, we can apply the philosophy of boosting and construct *ensembles* of instance selectors. Several rounds of an instance selection procedure are performed on different samples from the training set. The samples are obtained using the skewed distribution given by boosting.

We present the framework for adapting boosting to instance selection and constructing ensembles of instance selectors and show how this framework can be used with several of the most common boosting methods. In classifier boosting, an ensemble of classifiers is constructed iteratively. Each new classifier focuses on the instances that previous classifiers have found more difficult. Our approach constructs a combination of instance selection steps in a similar way. An instance selection algorithm is repeatedly applied. Each round focuses on the instances found more difficult by the previous instance selection algorithms. The different steps are combined by voting, which is common in boosting.

This approach opens a new field of research in which all of the methods developed for constructing ensembles of classifiers can be applied. Furthermore, it allows for the incorporation of one of the most successful classification technique, boosting, into another relevant task, data reduction. Although the proposed methodology can be used with other methods of constructing ensembles, we will restrict ourselves to boosting because this is a more successful technique.

This paper is organized as follows: Section 2 explains the proposed methodology; Section 3 reviews some related work; Section 4 details the experimental setup; Section 5 presents and discusses the results, and, finally, Section 6 presents the conclusions of our work and some future lines of research.

2. Boosting instance selection approach

A classification problem of K classes and N training observations consists of a set of instances whose class membership is known. Let $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_N, y_N)\}$ be a set of N training samples, where each instance $\mathbf{x}_i$ belongs to a domain X. Each label is an integer from the set $Y = \{1, \ldots, K\}$. A multi-class classifier is a function $f : X \to Y$ that maps an instance $\mathbf{x} \in X \subset \mathbb{R}^D$ to an element of Y.

The task is to find a definition for the unknown function $f(\mathbf{x})$ given the set of training instances. In a classifier ensemble framework, we have a set of classifiers $\mathbb{F} = \{f_1, f_2, \ldots, f_M\}$, each classifier performing a mapping of an instance vector $\mathbf{x} \in \mathbb{R}^D$ to the set of labels $Y = \{1, \ldots, K\}$. The design of classifier ensembles involves two main tasks: constructing the individual classifiers, f_m, and developing a combination rule that finds a class label for $\mathbf{x}$ based on the outputs of the classifiers $\{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\}$. For more-detailed descriptions of ensembles, the reader is referred to other reviews: [5–8] or [9].

Most techniques focus on obtaining a group of classifiers that are as accurate as possible but that disagree as much as possible. These two objectives are somewhat conflicting because if the classifiers are more accurate, it is obvious that they must agree more frequently. Many methods have been developed to enforce diversity on the classifiers that form an ensemble [5]. Kuncheva [10] identifies four fundamental approaches: (i) using different combination schemes, (ii) using different classifier models, (iii) using different feature subsets and (iv) using different training sets. The last one is perhaps the most commonly used. The algorithms in this last approach can be divided into two groups: algorithms that adaptively change the distribution of the training set based on the performance of the previous classifiers and algorithms that do not adapt the distribution. Boosting methods are the most representative of the first group. The most widely used boosting methods are AdaBoost [11] and its variants.

Bagging [12] is the most representative algorithm of the second group. Bagging (after *B*ootstrap *agg*regat*ing*) simply generates different bootstrap samples from the training set. Several empirical studies have shown that AdaBoost is able to reduce both the bias and variance components of errors [13–15]. However, Bagging seems to be more efficient in reducing bias than AdaBoost [15].

As stated above, boosting methods iteratively construct an ensemble of classifiers by modifying the distribution of instances in the dataset. Our approach to boosting instance selection is based on considering instance selection as a two-class classification problem and constructing an ensemble of these "classifiers" to perform the instance selection process. Proceeding as in classifier boosting, weights are initialized to a uniform distribution. Then, the instance selection algorithm is run with this uniform distribution, and the set of selected instances is recorded.

In each round of boosting, a new classifier is trained using a non-uniform distribution of the instances, where difficult instances receive more attention. After the classifier is trained, the distribution of the instances is updated by considering the error of the last classifier added, or the last few in some boosting methods, and a new round is performed. In our method, in each round, an instance selection algorithm is implemented using the non-uniform distribution given by the boosting weights. After the selection process, the algorithm must update the distribution of instances. To update the distribution, the method uses the subset of selected instances obtained by the last instance selection process and classifies all of the training instances using this subset and a k-NN rule.[1] With the error obtained by this classification, the weights are updated, and a new round is performed.

When ensembles of classifiers are constructed using boosting, the weights of the instances can be used in two ways. If the classifier learning algorithm accepts instance weights, they can be directly fed to the training process. The weights can also be used to obtain a sample from the training set using weighted sampling, where the instances with higher weights will be more frequently sampled. The former method is usually referred to as reweighting and the latter as resampling. Most instance selection algorithms do not accept weights for the instances; thus, we have chosen

[1] In fact, any other classifier can be used; however, to provide the needed focus, we will restrict ourselves to the case of a k-NN rule.