Information Sciences 317 (2015) 315–328

Contents lists available at ScienceDirect

Information Sciences

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

Improving over-fitting in ensemble regression by imprecise probabilities

Lev V. Utkin [a], Andrea Wiencierz [b,*]

[a] Department of Control, Automation, and System Analysis, Saint Petersburg State Forest Technical University, Institutsky pereulok 5, Saint-Petersburg 194021, Russia
[b] Department of Mathematics, University of York, York YO10 5DD, United Kingdom

ARTICLE INFO

Article history:
Received 7 November 2014
Received in revised form 24 March 2015
Accepted 17 April 2015
Available online 23 April 2015

Keywords:
Regression
AdaBoost algorithm
Over-fitting
Linear-vacuous mixture model
Kolmogorov–Smirnov bounds

ABSTRACT

In this paper, generalized versions of two ensemble methods for regression based on variants of the original AdaBoost algorithm are proposed. The generalization of these regression methods consists in restricting the unit simplex for the weights of the instances to a smaller set of weighting probabilities. Various imprecise statistical models can be used to obtain a restricted set of weighting probabilities, whose sizes each depend on a single parameter. For particular choices of this parameter, the proposed algorithms reduce to standard AdaBoost-based regression algorithms or to standard regression. The main advantage of the proposed algorithms compared to the basic AdaBoost-based regression methods is that they have less tendency to over-fitting, because the weights of the hard instances are restricted. Several simulations and applications furthermore indicate a better performance of the proposed regression methods in comparison with the corresponding standard regression methods.

1. Introduction

Regression analysis is one of the main problems in applied statistics. Roughly speaking, the aim is to estimate a function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^m$ with $m \in \mathbb{N}$, from a finite set of noisy samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} \subset \mathbb{R}$ and $n \in \mathbb{N}$. A large number of regression methods were developed in the last decades, many of which are based on the minimization of a risk functional defined by a certain loss function and by the probability distribution of the data [11,20,24]. In practice, the estimated function is obtained by minimizing the so-called empirical risk (possibly regularized) defined as the sum of the loss values for the given data points divided by n, which can be interpreted as the risk functional associated with the empirical distribution of the data. In this paper, we focus on this kind of regression methods within the discussed algorithms, because it is very easy to incorporate individual weights for the instances, which is a core element of the algorithms we generalize. The empirical distribution can be represented as the point $\hat{p} = (n^{-1}, \ldots, n^{-1})$ in the unit simplex with n vertices denoted by $S(1, n)$, which represents the set of all discrete

* Corresponding author. Tel.: +44 1904 32 3667.
E-mail addresses: lev.utkin@mail.ru (L.V. Utkin), andrea.wiencierz@york.ac.uk (A. Wiencierz).

http://dx.doi.org/10.1016/j.ins.2015.04.037
0020-0255/

probability measures on the n observations. Hence, weighted estimates can simply be interpreted as minimizers of the risk functional associated with another discrete probability distribution $p = (p_1, \ldots, p_n)$ of the data, different from the empirical distribution $\hat{p}$. A popular technique in regression estimation is the ensemble methodology. The popularity of ensemble methods for regression stems from success of boosting methods for classification, in particular, of the well-known AdaBoost (Adaptive Boosting) algorithm proposed by Freund and Schapire [6]. AdaBoost is a general purpose boosting algorithm that can be used in conjunction with many different learning algorithms to improve their performance. The basic scheme of the AdaBoost algorithm for classification is the following: Initially, a standard classifier is estimated, assigning identical weights to all examples, then, in each of a previously fixed number of iterations, the weights of all misclassified examples are increased, while the weights of correctly classified examples are decreased, before again computing a classifier accounting for the unequal weights of the instances. In this way, with each step, the classifier focuses more and more on the difficult examples of the training data set, thereby improving the classification accuracy. The final result obtained by AdaBoost is a weighted majority vote of the classifiers of each iteration, which has a better prediction performance than each of the individual classifiers alone.

Several detailed reviews of different kinds of boosting methods were published in the last decade [2,5,14–16]. One of the first boosting algorithms for regression is the so-called AdaBoost.R2 proposed in Drucker [4], where real-valued residuals replace the 0–1 misclassification errors in the evaluation of the estimates. However, the base regression estimates are evaluated by the weighted average of the absolute values of the residuals scaled to $[0, 1]$, which is a similar error measure to the misclassification rate. Up to the recent years, many more boosting methods for regression were developed, a recent survey is provided in Mendes-Moreira et al. [15]. In contrast to most of the ensemble-based algorithms using the weighted average of base regression estimates as their final regression functions, Kegl [13] analyzed the choice of the weighted median and proposed the corresponding algorithm called MedBoost. Another interesting boosting scheme for regression problems is proposed in Solomatine and Shrestha [19], where a threshold value for the residuals is introduced to transform the real-valued errors back to the 0–1 errors, which directly fit into the original AdaBoost scheme. This adaptation of the AdaBoost algorithm is called AdaBoost.RT and its properties were investigated in Shrestha and Solomatine [17].

A common feature of these boosting algorithms is that they iteratively search for a discrete probability distribution of the training data such that the regression error is minimized. If the algorithm searches too long or concentrates too much on a few hard-to-learn examples, the problem of over-fitting can occur. There are different approaches to deal with this problem. One possibility is to explicitly use the maximum number of iterations as a regularization parameter and select it optimally, for example, by cross-validation. Another way to regularize ensemble-based boosting algorithms can be derived following the idea of shrinkage proposed for gradient boosting by Friedman [8]. Shrinkage reduces the learning rate of the algorithm, which means that it shrinks the difference of the regression estimates between two subsequent steps. In the AdaBoost-based regression methods considered here, this idea can be transferred to shrinking the difference of the weights of each instance between two steps or to limiting the maximum size of the weights, thus preventing the algorithm from focusing too much on certain instances. In this paper, we follow this idea but we propose to use imprecise statistical models like the linear-vacuous mixture model or Kolmogorov–Smirnov bounds to restrict the set of weighting probabilities. To modify the boosting algorithms accordingly, we replace the adaption of the instances' weighting probabilities with the updating of weights in the convex linear combination of the extreme points of the restricted set. Thus, we here present a general tool for modifying available boosting algorithms and for constructing a number of new ensemble-based methods less prone to the problem of over-fitting.

In the following two sections, we propose the corresponding modifications of two popular boosting algorithms: AdaBoost.R2 introduced in Drucker [4] and AdaBoost.RT proposed in Solomatine and Shrestha [19]. Section 4 reviews suitable imprecise probability models to obtain the restricted set of weighting probabilities. Finally, we assess the performance of the modified algorithms by means of synthetic and real data.

2. AdaBoost.R2 and its modification

At first, we modify the AdaBoost.R2 algorithm proposed in Drucker [4]. The scheme of this boosting algorithm for regression is presented as Algorithm 1. Given a training data set $Z = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and a regression method which is suitable for weighted estimation, the algorithm requires a maximum number of iterations $T \in \mathbb{N}$ to be chosen a priori. Then, the iteration index t is set to one and the weighting probabilities $p_i^{(1)}$ are set to n^{-1} for all $i \in \{1, \ldots, n\}$. (Alternatively, the vector $p^{(1)}$ could be randomly selected from the unit simplex $S(1, n)$.) In each iteration step $t \in \{1, \ldots, T\}$, a regression function $\hat{f}^{(t)}$ is estimated using the weights $p^{(t)}$. In contrast to AdaBoost for classification, where the estimated classifiers are evaluated by their average misclassification error, the regression estimates are evaluated on the basis of the absolute residuals $|y_i - \hat{f}^{(t)}(x_i)|$ with $i \in \{1, \ldots, n\}$. Yet, to obtain an overall error measure similar to the misclassification rate, the absolute residuals are divided by the maximum value $D^{(t)}$ such that the weighted sum $\epsilon^{(t)}$ of the normalized residuals $\hat{e}_1^{(t)}, \ldots, \hat{e}_n^{(t)}$ lies in the interval $[0, 1]$. If $\epsilon^{(t)} > 0.5$, we exit the loop and use only the first $t - 1$ regression estimates to determine the final result. In the context