Computational Statistics and Data Analysis 93 (2016) 483–497

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

Noise peeling methods to improve boosting algorithms

Waldyn Martinez [a,*], J. Brian Gray [b]

[a] Department of Information Systems and Analytics, Miami University, Oxford, OH, USA

[b] Department of Information Systems, Statistics and Management Science, University of Alabama, Tuscaloosa, AL, USA

ARTICLE INFO

Article history:
Received 29 April 2014
Received in revised form 26 June 2015
Accepted 27 June 2015
Available online 10 July 2015

Keywords:
AdaBoost
Ensembles
Generalization error
Noise detection
Noise filtering
Outliers

ABSTRACT

Boosting refers to a family of methods that combine sequences of individual classifiers into highly accurate ensemble models through weighted voting. AdaBoost, short for "Adaptive Boosting", is the most well-known boosting algorithm. AdaBoost has many strengths. Among them, there is sufficient empirical evidence pointing to its performance being generally superior to that of individual classifiers. In addition, even when combining a large number of weak learners, AdaBoost can be very robust to overfitting usually with lower generalization error than other competing ensemble methodologies, such as bagging and random forests. However, AdaBoost, as most hard margin classifiers, tends to be sensitive to outliers and noisy data, since it assigns observations that have been misclassified a higher weight in subsequent iterations. It has recently been proven that for any booster with a potential convex loss function, and any nonzero random classification noise rate, there is a data set, which can be efficiently learnable by the booster if there is no noise, but cannot be learned with accuracy better than 1/2 with random classification noise present. Several techniques to identify and potentially delete (peel) noisy samples in binary classification are proposed in order to improve the performance of AdaBoost. It is found that peeling methods generally perform better than AdaBoost and other noise resistant boosters, especially when high levels of noise are present in the data.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Boosting algorithms combine sets of weak classifiers (learners) $h_t(\boldsymbol{x})$, $t = 1, 2, \ldots, T$, which are typically decision trees, into ensemble predictions in the form of weighted voting. The development of boosting algorithms has its roots in the Probably Approximately Correct (PAC) learning theory (Valiant, 1984). The PAC learning theory states that a concept class is strongly learnable if a learning algorithm is able to output a hypothesis with arbitrary accuracy in polynomial time, when sufficient training samples are readily available. A weakly learnable concept class drops the "arbitrary accuracy" to performing only slightly better than random. Schapire (1990) provided the first instance of an algorithm able to "boost" a weak classifier to a strong PAC learner. Boosting refers to a family of methods able to improve the accuracy of the weak classifiers by obtaining useful information from all the different learners combined. AdaBoost (Freund and Schapire, 1997), the most well-known boosting algorithm, possesses many desirable properties, including the strong PAC learning property. Generally, AdaBoost outperforms not only individual classifiers, but also most competing ensemble methodologies in many aspects. For example, under certain conditions, the test set error rates for subsequent iterations of AdaBoost continue

* Correspondence to: Department of Information Systems and Analytics, FSB 2020, Miami University, Oxford, OH 45056, USA. Tel.: +1 513 529 2154; fax: +1 513 529 9689.

E-mail addresses: martinwg@miamioh.edu (W. Martinez), bgray@cba.ua.edu (J.B. Gray).

http://dx.doi.org/10.1016/j.csda.2015.06.010
0167-9473/© 2015 Elsevier B.V. All rights reserved.

to decrease even when the training error has hit zero (Schapire et al., 1998; Opitz and Maclin, 1999). This property is remarkable, since it is expected that a more complex classifier would overfit the data easily. Additionally, the flexibility of using any weak classifier (with the ability to be trained with weighted data) has made AdaBoost a popular method in data mining and machine learning. On the other hand, AdaBoost tends to be sensitive to outliers and noise. Grove and Schuurmans (1998), Mason et al. (2000) and Dietterich (2000) have provided evidence that AdaBoost does overfit and the generalization error deteriorates rapidly when the data contain noise. Long and Servedio (2010) proved that for any booster with a potential convex loss function, and any nonzero random classification noise rate, there is a data set, which can be efficiently learnable by the booster if there is no noise, but cannot be learned with accuracy better than 1/2 with random classification noise present. Boosters with potential convex loss functions include the most common boosting algorithms to date.

Many methods that automatically handle noisy data and outliers have been proposed to alleviate the limitations of AdaBoost. These methods fall into the category of accommodating noisy observations by reducing the influence they have on the fit. Algorithms such as BrownBoost (Freund, 2001), GentleBoost (Friedman et al., 2000), LogitBoost (Friedman et al., 2000), MadaBoost (Domingo and Watanabe, 2000), LP_{Reg}-AdaBoost (Rätsch et al., 2001), ν-LP and ν-ARC (Rätsch et al., 1999) mostly attempt to accommodate noise by allowing unusual observations to fall on the wrong side of the prediction in subsequent iterations of the algorithm. Most of these methods rely on convex loss functions, and are therefore susceptible to the shortcomings presented in Long and Servedio (2010), that is, the methods cannot guarantee an accuracy better than 1/2 with random classification noise present. In addition, these algorithms do not generally possess the strong PAC learning property as the original AdaBoost does. On the other hand, using a detection/deletion (also called noise filtering or noise peeling) approach by pre-processing the data might be preferable under high noise circumstances (Gamberger et al., 2000). In this approach outliers are identified and deleted, and AdaBoost is refit to the remaining data, hopefully reducing the noise to levels where AdaBoost performs best.

In the next section, we give some preliminaries on the random classification noise framework as well as AdaBoost and ensembles. In Section 3, we study the performance of boosting methods under noise. In Section 4, we propose several techniques to identify and potentially delete or reduce noise in the training sample, in order to improve the performance of AdaBoost. Finally in Section 5, we compare the proposed techniques with the most common noise resistant boosting algorithms under different noise settings and scenarios.

2. Preliminaries

We assume we are given a training sample $S = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)\}$ of data pairs that are independently and identically distributed (i.i.d.) according to an unknown distribution P_{XY} with joint density $p(\boldsymbol{x}, y)$, where $Y \in \{-1, +1\}$ is a binary response variable and $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{p}}$ are the predictors. The training examples are also assumed to be initially generated by a sampling oracle $EX()$ without noise. The general goal of learning is to estimate a function $H : \mathbb{X} \to Y$ such that H will correctly classify unseen examples $(\boldsymbol{x}, y)$. The function is selected such that the generalization error $R[H]$ (also called the expected risk of the function) is minimized:

$$R[H] = E_{\boldsymbol{x},y} g(y, H(\boldsymbol{x})) = \int g(y, H(\boldsymbol{x}))\, dp(\boldsymbol{x}, y), \tag{1}$$

where $g(y, H(\boldsymbol{x}))$ is a suitable loss function. For binary classification the loss function $I(y_i \neq H(\boldsymbol{x}))$ is typically used, where $I(y_i \neq H(\boldsymbol{x})) = 1$ if $(y_i \neq H(\boldsymbol{x}))$, 0 otherwise. The generalization error cannot be minimized directly because the underlying distribution P_{XY} is unknown. The minimum theoretical value of $R[H]$ is often referred to as Bayes' minimum risk (Breiman, 2000). We refer to $P[.]$ as probabilities with respect to P_{XY} and $\hat{P}[.]$ as the probability with respect to the empirical distribution over the sample S.

For boosting methods, we assume a set of T classifiers $h_t(\boldsymbol{x})$, $t = 1, 2, \ldots, T$, is created from the space of classifiers $\mathcal{H}$. The classifiers are usually called base learners, individual learners or individual classifiers, and they are generated from the training data by a base-learning algorithm $\boldsymbol{\mathcal{B}}$. Each classifier takes a $p \times 1$ input vector $\boldsymbol{x}$ and produces a prediction $h_t(\boldsymbol{x}) \in \{-1, +1\}$ for a binary response variable Y. The combined classifier of the prediction is given by:

$$H(\boldsymbol{x}) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right), \tag{2}$$

where $sign : \mathbb{R} \to \{-1, +1\}$, such that $sign(x) = -1$ when $x < 0$, $+1$ otherwise; α_t is the weight associated with the tth weak classifier, $0 \leq \alpha_t \leq 1$ and $\sum_{t=1}^{T} \alpha_t = 1$. The AdaBoost algorithm is arguably the best-known boosting method. AdaBoost is described in Algorithm 1. Unlike earlier boosting methods (Schapire, 1990; Freund, 1995), AdaBoost adjusts adaptively to the errors of the weak hypotheses (hence the name Adaptive Boosting). For example, the step 3(e) in Algorithm 1 ensures more weight is given to instances that are misclassified by the base learning algorithm in previous rounds. The task of AdaBoost is to create a set of weak learners and determine their associated weights $\{\alpha_1, \ldots, \alpha_T\}$ based on a training sample of data S, to produce a combined prediction with small generalization error $R[H]$. Many researchers have pointed out that the margins of the observations may hold the answer as to why the AdaBoost algorithm and most ensemble

Download English Version:

https://daneshyari.com/en/article/415344

Download Persian Version:

https://daneshyari.com/article/415344

Daneshyari.com