Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Boosted adaptive filters

Dariush Kari [a,*], Ali H. Mirza [c], Farhan Khan [c], Huseyin Ozkan [b], Suleyman S. Kozat [c]

[a] *Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*
[b] *Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul 34956, Turkey*
[c] *Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Online boosting
Adaptive filtering
Boosted filter
Ensemble learning
Smooth boost
Mixture methods

ABSTRACT

We introduce the boosting notion of machine learning to the adaptive signal processing literature. In our framework, we have several adaptive filtering algorithms, i.e., the weak learners, that run in parallel on a common task such as equalization, classification, regression or filtering. We specifically provide theoretical bounds for the performance improvement of our proposed algorithms over the conventional adaptive filtering methods under some widely used statistical assumptions. We demonstrate an intrinsic relationship, in terms of boosting, between the adaptive mixture-of-experts and data reuse algorithms. Additionally, we introduce a boosting algorithm based on random updates that is significantly faster than the conventional boosting methods and other variants of our proposed algorithms while achieving an enhanced performance gain. Hence, the random updates method is specifically applicable to the fast and high dimensional streaming data. Specifically, we investigate Recursive Least Square-based and Least Mean Square-based linear and piecewise-linear regression algorithms in a mixture-of-experts setting and provide several variants of these well-known adaptation methods. Furthermore, we provide theoretical bounds for the computational complexity of our proposed algorithms. We demonstrate substantial performance gains in terms of mean squared error over the base learners through an extensive set of benchmark real data sets and simulated examples.

1. Introduction

Boosting is considered as one of the most important ensemble learning methods in the machine learning literature and it is extensively used in several different real-life applications from classification to regression [1,2]. As an ensemble learning method [3,4], boosting combines several parallel running "weakly" performing algorithms to build a final "strongly" performing algorithm [2,5]. This is accomplished by finding a linear combination of weak learning algorithms in order to minimize the total loss over a set of training data commonly using a functional gradient descent [6,7]. Boosting is successfully applied to several different problems in the machine learning literature including classification [7], regression [6,8], and prediction [9,10]. However, significantly less attention is given to the idea of boosting in adaptive signal processing [11,12] framework. To this end, our goal is (a) to introduce a new boosting approach for adaptive filtering, (b) derive several different adaptive filtering algorithms based on the boosting approach, (c) provide mathematical guarantees for the performance improvements of our algorithms, and (d) demonstrate the intrinsic connections of boosting with the adaptive mixture-of-experts algorithms [13,14] and data reuse algorithms [15]. Even though in [16] and [17], boosting has been used in limited scenarios of adaptive filtering, this paper provides mathematical guarantees along with an extensive set of experiments to illustrate the details of boosting approach in different adaptive filtering methods.

Although boosting is initially introduced in the batch setting [7], where algorithms boost themselves over a fixed set of training data, it is later extended to the online setting [18]. In the online setting, however, we neither need nor have access to a fixed set of training data, since the data samples arrive one by one as a stream [3,19]. Each newly arriving data sample is processed and then discarded without any storing. The online setting is naturally motivated by many real-life applications especially for the ones involving big data, where there may not be enough storage space available or the constraints of the problem require instant processing [20].

For adaptive filtering purposes [12], the online setting is especially important, where the sequentially arriving data is used to adjust the internal parameters of the filter, either to dynami-

* Corresponding author.
E-mail addresses: dkari2@illinois.edu (D. Kari), mirza@ee.bilkent.edu.tr (A.H. Mirza), khan@ee.bilkent.edu.tr (F. Khan), hozkan@sabanciuniv.edu (H. Ozkan), kozat@ee.bilkent.edu.tr (S.S. Kozat).

https://doi.org/10.1016/j.dsp.2018.07.012
1051-2004/

cally learn the underlying model or to track the nonstationary data statistics [13,21].

Specifically, we have m parallel running constituent filters (CF) [2] that receive the input vectors sequentially. Each CF uses an update method, such as Recursive Least Squares (RLS) or Least Mean Squares (LMS), depending on the target of the applications or problem constraints [21]. After receiving the input vector, each algorithm produces its output and then calculates its instantaneous error after the observation is revealed. In the most generic setting, this estimation/prediction error and the corresponding input vector are then used to update the internal parameters of the algorithm to minimize a priori defined loss function, e.g., instantaneous error for the LMS algorithm. These updates are performed for all of the m CFs in the mixture. However, in the online boosting approaches, these adaptations at each time proceed in rounds from top to bottom, starting from the first CF to the last one to achieve the "boosting" effect [22]. Furthermore, unlike the usual mixture approaches [13,14], the update of each CF depends on the previous CFs in the mixture. In particular, at each time t, after the kth CF calculates its error over $(\boldsymbol{x}_t, d_t)$ pair, it passes a certain weight to the next CF, the $(k+1)$th CF, quantifying how much error the constituent CFs from 1st to kth made on the current $(\boldsymbol{x}_t, d_t)$ pair. Based on the performance of the CFs from 1 to k on the current $(\boldsymbol{x}_t, d_t)$ pair, the $(k+1)$th CF may give a different emphasis (importance weight) to $(\boldsymbol{x}_t, d_t)$ pair in its adaptation in order to rectify the mistake of the previous CFs.

The proposed idea for online boosting is clearly related to the adaptive mixture-of-experts algorithms widely used in the machine learning literature, where several parallel running adaptive algorithms are combined to improve the performance. In the mixture methods, the performance improvement is achieved due to the diversity provided by using several different adaptive algorithms each having a different view or advantage [14]. This diversity is exploited to yield a final combined algorithm, which achieves a performance better than any of the algorithms in the mixture. Although the online boosting approach is similar to mixture approaches [14], there are significant differences. In the online boosting notion, the parallel running algorithms are not independent, i.e., one deliberately introduces the diversity by updating the CFs one by one from the first CF to the kth CF for each new sample based on the performance of all the previous CFs on this sample. In this sense, each adaptive algorithm, say the $(k+1)$th CF, receives feedback from the previous CFs, i.e., 1st to kth, and updates its inner parameters accordingly. As an example, if the current $(\boldsymbol{x}_t, d_t)$ is well modeled by the previous CFs, then the $(k+1)$th CF performs minor update using $(\boldsymbol{x}_t, d_t)$ and may give more emphasis (importance weight) to the later arriving samples that may be worse modeled by the previous CFs. Thus, by boosting, each adaptive algorithm in the mixture can concentrate on different parts of the input and output pairs achieving diversity and significantly improve the gain.

The linear adaptive filters, such as LMS or RLS, are among the simplest as well as the most widely used regression algorithms in the real-life applications [21]. Therefore, we use such algorithms as the CFs in our boosting algorithms. To this end, we first apply the boosting notion to several parallel running linear RLS-based CFs and introduce three different approaches to use the importance weights [22], namely "weighted updates", "data reuse", and "random updates". In the first approach, we use the importance weights directly to produce certain weighted RLS algorithms. In the second approach, we use the importance weights to construct data reuse adaptive algorithms [18]. However, data reuse in boosting, such as [18], is significantly different from the usual data reusing approaches in adaptive filtering [15]. As an example, in boosting, the importance weight coming from the kth CF determines the data reuse amount in the $(k+1)$th CF, i.e., it is not used for the kth filter, hence, achieving the diversity. The third approach uses the importance weights to decide whether to update the constituent CFs or not, based on a random number generated from a Bernoulli distribution with the parameter equal to the weight. The latter method can be effectively used for big data processing [23] due to the reduced complexity. The output of the constituent CFs is also combined using a linear mixture algorithm to construct the final output. We then update the final combination algorithm using the LMS algorithm [14]. Furthermore, we extend the boosting idea to parallel running linear LMS-based algorithms similar to the RLS case.

We start our discussion by investigating the related work in Section 2 and continue by introducing the problem setup and background in Section 3. We introduce our generic boosted adaptive filter algorithm in Section 4 and provide the mathematical justifications for its performance. Then, in Section 5, three different variants of the proposed boosting algorithm are derived, using the RLS and LMS, which are extended to piecewise linear filters in Section 6. Then, in Section 7 we provide the mathematical analysis for the computational complexity of the proposed algorithms. The paper concludes with an extensive set of experiments over the well-known benchmark data sets and simulation models widely used in the machine learning literature to demonstrate the significant gains achieved by the boosting notion.

2. Related work

AdaBoost is one of the earliest and most popular boosting methods, which has been used for binary and multiclass classifications as well as regression [7]. This algorithm has been well studied and has clear theoretical guarantees, and its excellent performance is explained rigorously [24]. However, AdaBoost cannot perform well on noisy data sets [25], therefore, other boosting methods have been suggested that are more robust against noise.

In order to reduce the effect of noise, SmoothBoost was introduced in [25] in a batch setting, which avoids overemphasizing the noisy samples, hence, provides robustness against noise. In [18], the authors extend bagging and boosting methods to an online setting, where they use a Poisson sampling process to approximate the reweighting algorithm. However, the online boosting method in [18] corresponds to AdaBoost, which is susceptible to noise. In [26], the authors use a greedy optimization approach to develop the boosting notion to the online setting and introduce stochastic boosting. Nevertheless, while most of the online boosting algorithms in the literature seek to approximate AdaBoost, [22] investigates the inherent difference between batch and online learning, extends the SmoothBoost algorithm to an online setting, and provides the mathematical guarantees for their algorithm. [22] points out that the online weak learners do not need to perform well on all possible distributions of data, instead, they have to perform well only with respect to smoother distributions. Recently, in [27], the authors have developed two online boosting algorithms for classification, an optimal algorithm in terms of the number of weak learners, and also an adaptive algorithm using the potential functions and boost-by-majority [28].

In addition to the classification task, the boosting approach has also been developed for the regression [6]. In [29], a boosting algorithm for regression is proposed, which is an extension of Adaboost.R [29]. Moreover, in [6], several gradient descent algorithms are presented, and some bounds on their performances are provided. In [26], the authors present a family of boosting algorithms for online regression through greedy minimization of a loss function. Also, in [30], the authors propose an online gradient boosting algorithm for regression.

In this paper, we propose a novel family of boosted adaptive filters (as an ensemble or combination technique) using the "online