A Deterministic Analysis of an Online Convex Mixture of Expert Algorithms

Mehmet A. Donmez, Sait Tunc and Suleyman S. Kozat, *Senior Member*

***Abstract*—We analyze an online learning algorithm that adaptively combines outputs of two constituent algorithms (or the experts) running in parallel to model an unknown desired signal. This online learning algorithm is shown to achieve (and in some cases outperform) the mean-square error (MSE) performance of the best constituent algorithm in the mixture in the steady-state. However, the MSE analysis of this algorithm in the literature uses approximations and relies on statistical models on the underlying signals and systems. Hence, such an analysis may not be useful or valid for signals generated by various real life systems that show high degrees of nonstationarity, limit cycles and, in many cases, that are even chaotic. In this paper, we produce results in an individual sequence manner. In particular, we relate the time-accumulated squared estimation error of this online algorithm at any time over any interval to the time-accumulated squared estimation error of the optimal convex mixture of the constituent algorithms directly tuned to the underlying signal in a deterministic sense without any statistical assumptions. In this sense, our analysis provides the transient, steady-state and tracking behavior of this algorithm in a strong sense without any approximations in the derivations or statistical assumptions on the underlying signals such that our results are guaranteed to hold. We illustrate the introduced results through examples.**

***Index Terms*—Learning algorithms, mixture of experts, deterministic, convexly constrained, steady-state, transient, tracking.**

I. Introduction

The problem of estimating or learning an unknown desired signal is heavily investigated in online learning [1]–[7] and adaptive signal processing literature [8]–[11]. However, in various applications, certain difficulties arise in the estimation process due to the lack of structural and statistical information about the data model. To resolve this lack of information, mixture approaches are proposed that adaptively combine outputs of multiple constituent algorithms performing the same task in the online learning literature under the mixture of experts framework [5]–[7] and adaptive signal processing under the adaptive mixture methods framework [8]–[10]. These parallel running algorithms can be seen as alternative hypotheses for modeling, which can be exploited for both performance improvement and robustness. Along these lines, an online convexly constrained mixture method that combines outputs of two learning algorithms is introduced in [9]. In this approach, the outputs of the constituent algorithms that run in parallel on the same task are adaptively combined under a convex constraint to minimize the final MSE. This adaptive mixture is shown to be universal with respect to the input algorithms in a certain stochastic sense such that this mixture achieves (and in some cases outperforms) the MSE performance of the best constituent algorithm in the mixture in the steady-state [9]. However, the MSE analysis of this adaptive mixture for the steady-state and during the transient regions uses approximations, e.g., separation assumptions, and relies on statistical models on the signals and systems, e.g., stationary data models [9], [10]. In this paper, we study this algorithm from the perspective of online learning and produce results in an individual sequence manner such that our results are guaranteed to hold for any bounded arbitrary signal.

Nevertheless, signals produced by various real life systems, such as in underwater acoustic communication applications, show high degrees of nonstationarity, limit cycles and, in many cases, are even chaotic so that they hardly fit to assumed statistical models [12]. Hence an analysis based on certain statistical assumptions or approximations may not be useful or adequate under these conditions. To this end, we refrain from making any statistical assumptions on the underlying signals and present an analysis that is guaranteed to hold for any bounded arbitrary signal without any approximations. In particular, we relate the performance of this learning algorithm that adaptively combines outputs of two constituent algorithms to the performance of the optimal convex combination that is directly tuned to the underlying signal and outputs of the constituent algorithms in a deterministic sense. Naturally, this optimal convex combination can only be chosen in hindsight after observing the whole signal and outputs a priori (before we even start processing the data). Since we compare the performance of this algorithm with respect to the best convex combination of the constituent filters in a deterministic sense over any time interval, our analysis provides, without any assumptions, the transient, the tracking and the steady-state behaviors together [5]–[7]. In particular, if the analysis window starts from $t = 1$, then we obtain the transient behavior; if the window length goes to infinity, then we obtain the steadystate behavior; and finally if the analyze window is selected arbitrary, then we get the tracking behavior as explained in detail in Section III. The corresponding bounds may also hold for unbounded signals such as with Gaussian and Laplacian distributions, if one can define reasonable bounds such that the effect of samples of the desired signal that are outside of an interval on the cumulative loss diminishes as the data size increases as demonstrated in Section III.

After we provide a brief system description in Section II, we present a deterministic analysis of the convexly constrained

This work is supported in part by IBM Faculty Award and Outstanding Young Scientist Award Program, Turkish Academy of Sciences. Suleyman S. Kozat, Mehmet A. Donmez and Sait Tunc ({skozat,medonmez,saittunc}@ku.edu.tr) are with the Competitive Signal Processing Laboratory at Koc University, Istanbul, tel: +902123381864.

```
The Convexly Constrained Algorithm:
  Parameters:
    $\mu > 0$: learning rate.
  Inputs:
    $y_t$: desired signal.
    $\hat{y}_{1,t}, \hat{y}_{2,t}$: constituent learning algorithms.
  Outputs:
    $\hat{y}_t$: estimate of the desired signal.
  Initialization: Set the initial weights $\lambda_1 = 1/2$ and $\rho_1 = 0$.
  for $t = 1 : \ldots : n$,
    % receive the constituent algorithm outputs $\hat{y}_{1,t}$ and $\hat{y}_{2,t}$ and
    % estimate the desired signal
    $\hat{y}_t = \lambda_t \hat{y}_{1,t} + (1 - \lambda_t)\hat{y}_{2,t}$
    % Upon receiving $y_t$, update the weight according to the rule:
    $\rho_{t+1} = \rho_t + \mu e_t \lambda_t (1 - \lambda_t)[\hat{y}_{1,t} - \hat{y}_{2,t}]$
    $\lambda_{t+1} = \frac{1}{1+e^{-\rho_{t+1}}}$
  endfor
```

TABLE I: The learning algorithm that adaptively combines outputs of two algorithms.

mixture algorithm in Section III, where the performance bounds are given as a theorem and a lemma. We illustrate the introduced results through examples in Section IV. The paper concludes with certain remarks.

II. Problem Description

In this framework, we have a desired signal $\{y_t\}_{t\geq 1}$, where $|y_t| \leq Y < \infty$, and two constituent algorithms running in parallel producing $\{\hat{y}_{1,t}\}_{t\geq 1}$ and $\{\hat{y}_{2,t}\}_{t\geq 1}$, respectively, as the estimations (or predictions) of the desired signal $\{y_t\}_{t\geq 1}$. We assume that Y is known. Here, we have no restrictions on $\hat{y}_{1,t}$ or $\hat{y}_{2,t}$, e.g., these outputs are not required to be causal, however, without loss of generality, we assume $|\hat{y}_{1,t}| \leq Y$ and $|\hat{y}_{2,t}| \leq Y$, i.e., these outputs can be clipped to the range $[-Y, Y]$ without sacrificing performance under the squared error. As an example, the desired signal and outputs of the constituent learning algorithms can be single realizations generated under the framework of [9]. At each time t, the convexly constrained algorithm receives an input vector $\boldsymbol{x}_t \stackrel{\triangle}{=} [\hat{y}_{1,t}\ \hat{y}_{2,t}]^T$ and outputs

$$\hat{y}_t = \lambda_t \hat{y}_{1,t} + (1 - \lambda_t)\hat{y}_{2,t} = \boldsymbol{w}_t^T \boldsymbol{x}_t,$$

where $\boldsymbol{w}_t \stackrel{\triangle}{=} [\lambda_t\ (1 - \lambda_t)]^T$, $0 \leq \lambda_t \leq 1$, as the final estimate. The final estimation error is given by $e_t = y_t - \hat{y}_t$.

The combination weight λ_t is trained through an auxiliary variable using a stochastic gradient update to minimize the squared final estimation error as

$$\lambda_t = \frac{1}{1 + e^{-\rho_t}}, \tag{1}$$

$$\begin{aligned}\rho_{t+1} &= \rho_t - \mu \nabla_\rho e_t^2 \big|_{\rho=\rho_t} \\ &= \rho_t + \mu e_t \lambda_t (1 - \lambda_t)[\hat{y}_{1,t} - \hat{y}_{2,t}],\end{aligned} \tag{2}$$

where $\mu > 0$ is the learning rate. The combination parameter λ_t in (1) is constrained to lie in $[\lambda^+, (1-\lambda^+)]$, $0 < \lambda^+ < 1/2$ in [9], since the update in (2) may slow down when λ_t is too close to the boundaries. We follow the same restriction and analyze (2) under this constraint. The algorithm is presented in Table I.

Under the deterministic analysis framework, the performance of the algorithm is determined by the time-accumulated squared error [5], [7], [13]–[15]. When applied to any sequence $\{y_t\}_{t\geq 1}$, the algorithm of (1) yields the total accumulated loss

$$L_n(\hat{y}, y) = L_n(\boldsymbol{w}_t^T \boldsymbol{x}_t, y) \stackrel{\triangle}{=} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \tag{3}$$

for any n. We emphasize that for unbounded signals such as Gaussian and Laplacian distributions, we can define a suitable Y such that the samples of y_t are inside of the interval $[-Y, Y]$ with high probability and the effect of the samples that are outside of this interval on the cumulative loss (3) diminishes as n gets larger.

We next provide deterministic bounds on $L_n(\hat{y}, y)$ with respect to the best convex combination $\min\limits_{\beta\in[0,1]} L_n(\hat{y}_\beta, y)$, where

$$L_n(\hat{y}_\beta, y) = L_n(\boldsymbol{u}^T \boldsymbol{x}_t, y) = \sum_{t=1}^{n} (y_t - \hat{y}_{\beta,t})^2$$

and

$$\hat{y}_{\beta,t} \stackrel{\triangle}{=} \beta \hat{y}_{1,t} + (1 - \beta)\hat{y}_{2,t} = \boldsymbol{u}^T \boldsymbol{x}_t,$$

$\boldsymbol{u} \stackrel{\triangle}{=} [\beta\ 1{-}\beta]^T$, that holds uniformly in an individual sequence manner without any stochastic assumptions on y_t, $\hat{y}_{1,t}$, $\hat{y}_{2,t}$ or n. Note that the best fixed convex combination parameter

$$\beta_o = \arg \min_{\beta\in[0,1]} L_n(\hat{y}_\beta, y)$$

and the corresponding estimator

$$\hat{y}_{\beta_o,t} = \beta_o \hat{y}_{1,t} + (1 - \beta_o)\hat{y}_{2,t},$$

which we compare the performance against, can only be determined after observing the entire sequences, i.e., $\{y_t\}$, $\{\hat{y}_{1,t}\}$ and $\{\hat{y}_{2,t}\}$, in advance for all n.

III. A Deterministic Analysis

In this section, we first relate the accumulated loss of the mixture to the accumulated loss of the best convex combination that minimizes the accumulated loss in the following theorem. Then, we demonstrate that one cannot improve the convergence rate of this upper bound using our methodology directly and the Kullback-Leibler (KL) divergence [6] as the distance measure by providing counter examples as a lemma. The use of the KL divergence as a distance measure for obtaining worst-case loss bounds was pioneered by Littlestone [16], and later adopted extensively in the online learning literature [6], [7], [17]. We emphasize that although the steady-state and transient MSE performances of the convexly constrained mixture algorithm are analyzed with respect to the constituent learning algorithms [9], [10], we perform the steady-state, transient and tracking analysis without any stochastic assumptions or use any approximations in the following theorem.

Theorem: The algorithm given in (2), when applied to any sequence $\{y_t\}_{t\geq 1}$, with $|y_t| \leq Y < \infty$, yields, for any n and $\epsilon > 0$

$$L_n(\hat{y}, y) - \left(\frac{2\epsilon+1}{1-z^2}\right) \min_{\beta\in[0,1]} \{L_n(\hat{y}_\beta, y)\} \leq O\left(\frac{1}{\epsilon}\right), \quad (4)$$

where $O\,(.)$ is the order notation, $\hat{y}_{\beta,t} = \beta\hat{y}_{1,t} + (1-\beta)\hat{y}_{2,t}$, $z \triangleq \frac{1-4\lambda^+(1-\lambda^+)}{1+4\lambda^+(1-\lambda^+)} < 1$ and step size $\mu = \frac{4\epsilon}{2\epsilon+1}\frac{2+2z}{Y^2}$, provided that $\lambda_t \in [\lambda^+, 1-\lambda^+]$ for all t during the adaptation.

This theorem provides a regret bound for the algorithm (2) showing that the cumulative loss of the convexly constrained algorithm is close to a factor times the cumulative loss of the algorithm with the best weight chosen in hindsight. If we define the regret

$$R_n \triangleq L_n(\hat{y}, y) - \left(\frac{2\epsilon+1}{1-z^2}\right) \min_{\beta\in[0,1]} \{L_n(\hat{y}_\beta, y)\}, \quad (5)$$

then equation (4) implies that time-normalized regret

$$\frac{R_n}{n} \triangleq \frac{L_n(\hat{y}, y)}{n} - \left(\frac{2\epsilon+1}{1-z^2}\right) \min_{\beta\in[0,1]} \left\{\frac{L_n(\hat{y}_\beta, y)}{n}\right\}$$

converges to zero at a rate $O\left(\frac{1}{n\epsilon}\right)$ uniformly over the desired signal and the outputs of constituent algorithms. Moreover, (4) provides the exact trade-off between the transient and steady-state performances of the convex mixture in a deterministic sense without any assumptions or approximations. Note that (4) is guaranteed to hold independent of the initial condition of the combination weight λ_t for any time interval in an individual sequence manner. Hence, (4) also provides the tracking performance of the convexly constrained algorithm in a deterministic sense. From (4), we observe that the convergence rate of the right hand side, i.e., the bound, is $O\left(\frac{1}{n\epsilon}\right)$, and, as in the stochastic case [10], to get a tighter asymptotic bound with respect to the optimal convex combination of the learning algorithms, we require a smaller ϵ, i.e., smaller learning rate μ, which increases the right hand side of (4). Although this result is well-known in the adaptive filtering literature and appears widely in stochastic contexts, however, this trade-off is guaranteed to hold in here without any statistical assumptions or approximations. Note that the optimal convex combination in (4), i.e., minimizing β, depends on the entire signal and outputs of the constituent algorithms for all n and hence it can only be determined in hindsight.

Proof: To prove the theorem, we use the approach introduced in [7] (and later used in [6]) based on measuring progress of a mixture algorithm using certain distance measures.

We first convert (2) to a direct update on λ_t and use this direct update in the proof. Using

$$e^{-\rho_t} = \frac{1-\lambda_t}{\lambda_t}$$

from (1), the update in (2) can be written as

$$\begin{aligned}\lambda_{t+1} &= \frac{1}{1+e^{-\rho_{t+1}}} \\ &= \frac{1}{1+e^{-\rho_t - \mu e_t \lambda_t(1-\lambda_t)[\hat{y}_{1,t}-\hat{y}_{2,t}]}} \\ &= \frac{1}{1+\frac{1-\lambda_t}{\lambda_t}e^{-\mu e_t \lambda_t(1-\lambda_t)[\hat{y}_{1,t}-\hat{y}_{2,t}]}} \\ &= \frac{\lambda_t e^{\mu e_t \lambda_t(1-\lambda_t)\hat{y}_{1,t}}}{\lambda_t e^{\mu e_t \lambda_t(1-\lambda_t)\hat{y}_{1,t}} + (1-\lambda_t)e^{\mu e_t \lambda_t(1-\lambda_t)\hat{y}_{2,t}}}. \quad (6)\end{aligned}$$

Unlike [6] (Lemma 5.8), our update in (6) has, in a certain sense, an adaptive learning rate $\mu\lambda_t(1-\lambda_t)$ which requires different formulation, however, follows similar lines of [6] in certain parts.

Here, for a fixed $\beta \in [0,1]$, we define an estimator

$$\hat{y}_{\beta,t} \triangleq \beta\hat{y}_{1,t} + (1-\beta)\hat{y}_{2,t} = \boldsymbol{u}^T\boldsymbol{x}_t,$$

where $\beta \in [0,1]$ and $\boldsymbol{u} \triangleq [\beta \;\; 1-\beta]^T$. Defining

$$\zeta_t = e^{\mu e_t \lambda_t(1-\lambda_t)},$$

we have from (6)

$$\begin{aligned}&\beta \ln\left(\frac{\lambda_{t+1}}{\lambda_t}\right) + (1-\beta)\ln\left(\frac{1-\lambda_{t+1}}{1-\lambda_t}\right) \\ &= \hat{y}_{\beta,t}\ln\zeta_t - \ln\left(\lambda_t\zeta_t^{\hat{y}_{1,t}} + (1-\lambda_t)\zeta_t^{\hat{y}_{2,t}}\right). \quad (7)\end{aligned}$$

Using the inequality

$$\alpha^x \leq 1 - x(1-\alpha)$$

for $\alpha \geq 0$ and $x \in [0,1]$ from [7], we have

$$\begin{aligned}\zeta_t^{\hat{y}_{1,t}} &= (\zeta_t^{2Y})^{\frac{\hat{y}_{1,t}+Y}{2Y}}\zeta_t^{-Y} \\ &\leq \zeta_t^{-Y}\left(1 - \frac{\hat{y}_{1,t}+Y}{2Y}(1-\zeta_t^{2Y})\right),\end{aligned}$$

which implies in (7)

$$\begin{aligned}&\ln\left(\lambda_t\zeta_t^{\hat{y}_{1,t}} + (1-\lambda_t)\zeta_t^{\hat{y}_{2,t}}\right) \\ &\leq \ln\left(\zeta_t^{-Y}(1 - \frac{\lambda\hat{y}_{1,t} + (1-\lambda_t)\hat{y}_{2,t} + Y}{2Y}(1-\zeta_t^{2Y}))\right) \\ &= -Y\ln\zeta_t + \ln\left(1 - \frac{\hat{y}_t + Y}{2Y}(1-\zeta_t^{2Y})\right), \quad (8)\end{aligned}$$

where $\hat{y}_t = \lambda_t\hat{y}_{1,t} + (1-\lambda_t)\hat{y}_{2,t}$. As in [6], one can further bound (8) using

$$\ln(1 - q(1-e^p)) \leq pq + \frac{p^2}{8}$$

for $0 \leq q < 1$ (originally from [7])

$$\begin{aligned}&\ln\left(\lambda_t\zeta_t^{\hat{y}_{1,t}} + (1-\lambda_t)\zeta_t^{\hat{y}_{2,t}}\right) \\ &\leq -Y\ln\zeta_t + (\hat{y}_t + Y)\ln\zeta_t + \frac{Y^2(\ln\zeta_t)^2}{2}. \quad (9)\end{aligned}$$

Using (9) in (7) yields

$$\begin{aligned}&\beta\ln\left(\frac{\lambda_{t+1}}{\lambda_t}\right) + (1-\beta)\ln\left(\frac{1-\lambda_{t+1}}{1-\lambda_t}\right) \geq \quad (10) \\ &(\hat{y}_{\beta,t} + Y)\ln\zeta_t - (\hat{y}_t + Y)\ln\zeta_t - \frac{Y^2(\ln\zeta_t)^2}{2}.\end{aligned}$$

At each adaptation, the progress made by the algorithm towards $\boldsymbol{u}$ at time t is measured as $D(\boldsymbol{u}||\boldsymbol{w}_t) - D(\boldsymbol{u}||\boldsymbol{w}_{t+1})$, where $\boldsymbol{w}_t \overset{\triangle}{=} [\lambda_t \; (1-\lambda_t)]^T$ and

$$D(\boldsymbol{u}||\boldsymbol{w}) \overset{\triangle}{=} \sum_{i=1}^{2} u_i \ln(u_i/w_i)$$

is the KL divergence [7], [18], $\boldsymbol{u} \in [0,1]^2$, $\boldsymbol{w} \in [0,1]^2$. We require that this progress is at least $a(y_t - \hat{y}_t)^2 - b(y_t - \hat{y}_{\beta,t})^2$ for certain a, b, μ [6], [7], i.e.,

$$\begin{aligned} & a(y_t - \hat{y}_t)^2 - b(y_t - \hat{y}_{\beta,t})^2 \\ & \leq D(\boldsymbol{u}||\boldsymbol{w}_t) - D(\boldsymbol{u}||\boldsymbol{w}_{t+1}) \\ & = \beta \ln\left(\frac{\lambda_{t+1}}{\lambda_t}\right) + (1-\beta)\ln\left(\frac{1-\lambda_{t+1}}{1-\lambda_t}\right), \end{aligned} \tag{11}$$

which yields the desired deterministic bound in (4) after telescoping.

In information theory and probability theory, the KL divergence, which is also known as the relative entropy, is empirically shown to be an efficient measure of the distance between two probability vectors [6], [7], [18]. Here, the vectors $\boldsymbol{u}$ and $\boldsymbol{w}_t$ are probability vectors, i.e., $\boldsymbol{u}, \boldsymbol{w}_t \in [0,1]^2$ and $\boldsymbol{u}^T\mathbf{1} = \boldsymbol{w}_t^T\mathbf{1} = 1$, where $\mathbf{1} \overset{\triangle}{=} [1 \; 1]^T$. This use of KL divergence as a distance measure between weight vectors is widespread in the online learning literature [6], [13], [17].

We observe from (11) and (10) that to prove the theorem, it is sufficient to show that $G(y_t, \hat{y}_t, \hat{y}_{\beta,t}, \zeta_t) \leq 0$, where

$$\begin{aligned} G(y_t, \hat{y}_t, \hat{y}_{\beta,t}, \zeta_t) & \overset{\triangle}{=} -(\hat{y}_{\beta,t} + Y)\ln\zeta_t + (\hat{y}_t + Y)\ln\zeta_t \\ & + \frac{Y^2(\ln\zeta_t)^2}{2} + a(y_t - \hat{y}_t)^2 - b(y_t - \hat{y}_{\beta,t})^2. \end{aligned} \tag{12}$$

For fixed $y_t, \hat{y}_t, \zeta_t$, $G(y_t, \hat{y}_t, \hat{y}_{\beta,t}, \zeta_t)$ is maximized when $\frac{\partial G}{\partial \hat{y}_{\beta,t}} = 0$, i.e.,

$$\hat{y}_{\beta,t} - y_t + \frac{\ln\zeta_t}{2b} = 0$$

since $\frac{\partial^2 G}{\partial \hat{y}_{\beta,t}^2} = -2b < 0$, yielding $\hat{y}^*_{\beta,t} = y_t - \frac{\ln\zeta_t}{2b}$. Note that while taking the partial derivative of $G(\cdot)$ with respect to $\hat{y}_{\beta,t}$ and finding $\hat{y}^*_{\beta,t}$, we assume that all $y_t, \hat{y}_t, \zeta_t$ are fixed, i.e., their partial derivatives with respect to $\hat{y}_{\beta,t}$ is zero. This yields an upper bound on $G(\cdot)$ in terms of $\hat{y}_{\beta,t}$. Hence, it is sufficient to show that $G(y_t, \hat{y}_t, \hat{y}^*_{\beta,t}, \zeta_t) \leq 0$ such that [6]

$$\begin{aligned} & G(y_t, \hat{y}, \hat{y}^*_{\beta,t}, \zeta_t) \\ & = -\left(y_t + Y - \frac{\ln\zeta_t}{2b}\right)\ln\zeta_t + (\hat{y}_t + Y)\ln\zeta_t \\ & + \frac{Y^2(\ln\zeta_t)^2}{2} + a(y_t - \hat{y}_t)^2 - \frac{(\ln\zeta_t)^2}{4b} \qquad (13) \\ & = a(y_t - \hat{y}_t)^2 - (y_t - \hat{y}_t)\ln\zeta_t + \frac{(\ln\zeta_t)^2}{4b} \\ & + \frac{Y^2(\ln\zeta_t)^2}{2} \\ & = (y_t - \hat{y}_t)^2 \times \Bigg[a - \mu\lambda_t(1-\lambda_t) \\ & + \frac{\mu^2{\lambda_t}^2(1-\lambda_t)^2}{4b} + \frac{Y^2\mu^2{\lambda_t}^2(1-\lambda_t)^2}{2}\Bigg]. \qquad (14) \end{aligned}$$

For (14) to be negative, defining $k \overset{\triangle}{=} \lambda_t(1-\lambda_t)$ and

$$H(k) \overset{\triangle}{=} k^2\mu^2\left(\frac{Y^2}{2} + \frac{1}{4b}\right) - \mu k + a,$$

it is sufficient to show that $H(k) \leq 0$ for $k \in [\lambda^+(1-\lambda^+), \frac{1}{4}]$, i.e., $k \in [\lambda^+(1-\lambda^+), \frac{1}{4}]$ when $\lambda_t \in [\lambda^+, (1-\lambda^+)]$, since $H(k)$ is a convex quadratic function of k, i.e., $\frac{\partial^2 H}{\partial k^2} > 0$. Hence, we require the interval where the function $H(\cdot)$ is negative should include $[\lambda^+(1-\lambda^+), \frac{1}{4}]$, i.e., the roots k_1 and k_2 (where $k_2 \leq k_1$) of $H(\cdot)$ should satisfy

$$k_1 \geq \frac{1}{4}, \; k_2 \leq \lambda^+(1-\lambda^+),$$

where

$$k_1 = \frac{\mu + \sqrt{\mu^2 - 4\mu^2 a\left(\frac{Y^2}{2} + \frac{1}{4b}\right)}}{2\mu^2(\frac{Y^2}{2} + \frac{1}{4b})} = \frac{1 + \sqrt{1 - 4as}}{2\mu s}, \tag{15}$$

$$k_2 = \frac{\mu - \sqrt{\mu^2 - 4\mu^2 a\left(\frac{Y^2}{2} + \frac{1}{4b}\right)}}{2\mu^2(\frac{Y^2}{2} + \frac{1}{4b})} = \frac{1 - \sqrt{1 - 4as}}{2\mu s} \tag{16}$$

and

$$s \overset{\triangle}{=} \left(\frac{Y^2}{2} + \frac{1}{4b}\right).$$

To satisfy $k_1 \geq 1/4$, we straightforwardly require from (15)

$$\frac{2 + 2\sqrt{1 - 4as}}{s} \geq \mu.$$

To get the tightest upper bound for (15), we set

$$\mu = \frac{2 + 2\sqrt{1 - 4as}}{s},$$

i.e., the largest allowable learning rate.

To have $k_2 \leq \lambda^+(1-\lambda^+)$ with $\mu = \frac{2+2\sqrt{1-4as}}{s}$, from (16) we require

$$\frac{1 - \sqrt{1 - 4as}}{4(1 + \sqrt{1 - 4as})} \leq \lambda^+(1-\lambda^+). \tag{17}$$

Equation (17) yields

$$as = a\left(\frac{Y^2}{2} + \frac{1}{4b}\right) \leq \frac{1 - z^2}{4}, \tag{18}$$

where

$$z \overset{\triangle}{=} \frac{1 - 4\lambda^+(1-\lambda^+)}{1 + 4\lambda^+(1-\lambda^+)}$$

and $z < 1$ after some algebra.

To satisfy (18), we set $b = \frac{\epsilon}{Y^2}$ for any (or arbitrarily small) $\epsilon > 0$ that results

$$a \leq \frac{(1 - z^2)\epsilon}{Y^2(2\epsilon + 1)}. \tag{19}$$

To get the tightest bound in (11), we select

$$a = \frac{(1 - z^2)\epsilon}{Y^2(2\epsilon + 1)}$$

in (19). Such selection of a, b and μ results in (11)

$$\begin{aligned} & \left(\frac{(1 - z^2)\epsilon}{Y^2(2\epsilon + 1)}\right)(y_t - \hat{y}_t)^2 - \left(\frac{\epsilon}{Y^2}\right)(y_t - \hat{y}_{\beta,t})^2 \\ & \leq \beta \ln\left(\frac{\lambda_{t+1}}{\lambda_t}\right) + (1-\beta)\ln\left(\frac{1-\lambda_{t+1}}{1-\lambda_t}\right). \end{aligned} \tag{20}$$

After telescoping, i.e., summation over t, $\sum_{t=1}^{n}$, (20) yields

$$\begin{aligned}
&aL_n(\hat{y},y) - b\min_{\beta\in[0,1]}\{L_n(\hat{y}_\beta,y)\} \\
&\leq \beta\ln\left(\frac{\lambda_{n+1}}{\lambda_1}\right) + (1-\beta)\ln\left(\frac{1-\lambda_{n+1}}{1-\lambda_1}\right) \leq O(1), \quad (21)
\end{aligned}$$

so that

$$\begin{aligned}
&\left(\frac{(1-z^2)\epsilon}{Y^2(2\epsilon+1)}\right)L_n(\hat{y},y) - \left(\frac{\epsilon}{Y^2}\right)\min_{\beta\in[0,1]}\{L_n(\hat{y}_\beta,y)\} \\
&\leq O(1). \quad (22)
\end{aligned}$$

Hence, it follows that

$$\begin{aligned}
&L_n(\hat{y},y) - \left(\frac{2\epsilon+1}{1-z^2}\right)\min_{\beta\in[0,1]}\{L_n(\hat{y}_\beta,y)\} \quad (23) \\
&\leq \frac{(2\epsilon+1)Y^2}{n\epsilon(1-z^2)}O(1) \leq O\left(\frac{1}{\epsilon}\right), \quad (24)
\end{aligned}$$

which is the desired bound.

Note that using

$$b = \frac{\epsilon}{Y^2},\ a = \frac{(1-z^2)\epsilon}{Y^2(2\epsilon+1)},\ s = \left(\frac{Y^2}{2} + \frac{1}{4b}\right),$$

we get

$$\mu = \frac{2+2\sqrt{1-4as}}{s} = \frac{4\epsilon}{2\epsilon+1}\frac{2+2z}{Y^2},$$

after some algebra, as in the statement of the theorem. This concludes the proof of the theorem. □

In the following lemma, we show that the order of the upper bound using the KL divergence as the distance measure under the same methodology cannot be improved by presenting an example in which the bound on b is of the same order as that given in the theorem.

Lemma: For positive real constants a, b and μ which satisfies (11) for all $|y_t| \leq Y$, $|\hat{y}_{1,t}| \leq Y$ and $|\hat{y}_{2,t}| \leq Y$ and $\lambda_t \in [\lambda^+,(1-\lambda^+)]$, we require

$$b \geq 4a + \frac{a}{4\lambda^+(1-\lambda^+)}.$$

Proof: Since the inequality in (11) should be satisfied for all possible y_t, $\hat{y}_{1,t}$, $\hat{y}_{2,t}$, β and λ_t, the proper values of a, b and μ should satisfy (11) for any particular selection of y_t, $\hat{y}_{1,t}$, $\hat{y}_{2,t}$, β and λ_t. First we consider

$$y_t = \hat{y}_{1,t} = Y,\ \hat{y}_{2,t} = 0,\ \beta = 1,\ \lambda_t = \lambda^+,$$

(or,similarly, $y_t = \hat{y}_{1,t} = Y$, $\hat{y}_{2,t} = -Y$ and $\lambda_t = \lambda^+$). In this case, we have

$$\begin{aligned}
&a(Y-\lambda^+Y)^2 \\
&\leq -\ln(\lambda^+ + (1-\lambda^+)e^{\mu(Y-\lambda^+Y)\lambda^+(1-\lambda^+)(-Y)}) \\
&\leq -\lambda^+\ln 1 - \mu(1-\lambda^+)^2\lambda^+Y(1-\lambda^+)(-Y) \quad (25) \\
&= \mu(1-\lambda^+)^3\lambda^+Y^2, \quad (26)
\end{aligned}$$

where (25) follows from the Jensen's Inequality for concave function $\ln(\cdot)$. By (26), we have

$$\mu \geq \frac{a}{\lambda^+(1-\lambda^+)}. \quad (27)$$

For another particular case where

$$y_t = -Y/2,\ \hat{y}_{1,t} = 0,\ \hat{y}_{2,t} = Y,\ \beta = 1,\ \lambda_t = 1/2,$$

we have

$$\begin{aligned}
&a(-Y)^2 - b(-\frac{Y}{2})^2 \\
&\leq -\ln(\frac{1}{2} + \frac{1}{2}e^{\mu(-Y)\frac{1}{4}(-\frac{Y}{2})}) \\
&\leq -\frac{1}{2}\mu\frac{Y^2}{8}, \quad (28)
\end{aligned}$$

where (28) also follows from the Jensen's Inequality. By (28), we have

$$b \geq 4a + \frac{\mu}{4} \geq 4a + \frac{a}{4\lambda^+(1-\lambda^+)}, \quad (29)$$

where (29) follows from (27), which finalizes the proof. □

IV. Simulations

In this section, we illustrate the performance of the learning algorithm (2) and the introduced results through examples. We demonstrate that the upper bound given in (4) is asymptotically tight by providing specific sequences for the desired signal y_t and the outputs of constituent algorithms $\hat{y}_{1,t}$ and $\hat{y}_{2,t}$. We also demonstrate that to get a tighter asymptotic bound, we require a smaller learning rate μ, as suggested by our theoretical analysis.

In the first case, we present the regret of the learning algorithm (2) defined in (5) and the corresponding upper bound given in (4). We first set $Y = 0.5$, $\lambda^+ = 0.08$ and $\mu = 0.08$. Here, the desired signal is given by

$$y_t = Y$$

for $t = 1,\ldots,10000$. For this specific example, the parallel running constituent algorithms produce the sequences

$$\hat{y}_{1,t} = Y,\ \hat{y}_{2,t} = (-1)^tY$$

for $t = 1,\ldots,10000$. Note that, in this case, the best convex combination weight is $\beta_o = 1$ and the cumulative loss of the best convex combination is 0 since y_t and $\hat{y}_{1,t}$ are identical. In Fig. 1a, we plot the time-normalized regret of the learning algorithm (2) "Time-normalized regret, $\mu_1 = 0.08$" and the upper bound given in (4) "$O(1/(n\epsilon_1))$". From Fig. 1a, we observe that the bound introduced in (4) is asymptotically tight, i.e., as n gets larger, the gap between the upper bound and the time-normalized regret gets smaller.

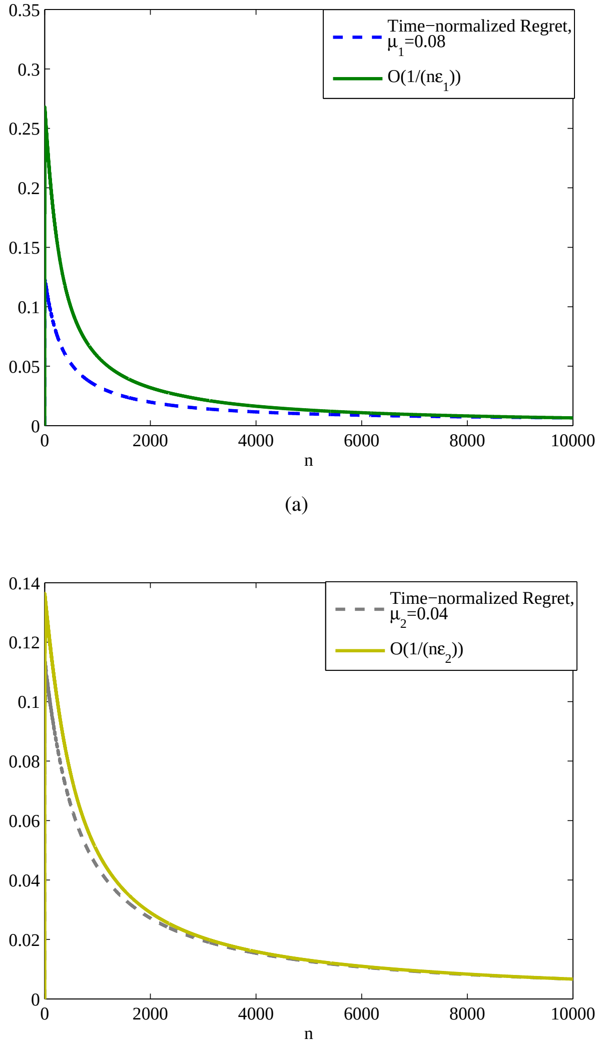
In the second case, we set $Y = 0.54$, $\lambda^+ = 0.08$ and $\mu = 0.04$. Here, the desired signal is given by

$$y_t = 0.5$$

for $t = 1,\ldots,10000$. For this example, the constituent algorithms produce the sequences

$$\hat{y}_{1,t} = Y,\ \hat{y}_{2,t} = (-1)^t0.5$$

for $t = 1,\ldots,10000$. In this case, the best convex combination weight is $\beta_o = 0.96$, however, unlike the first case, the cumulative loss of the best convex combination is nonzero. In Fig. 1b, we plot the time-normalized regret of the learning



(a)

(b)

Fig. 1: Tightness of the regret bound. (a) $\mu_1 = 0.08$. (b) $\mu_2 = 0.04$.

algorithm (2) “Time-normalized regret, $\mu_2 = 0.04$” and the corresponding upper bound given in (4) “$O(1/(n\epsilon_2))$” for this example. We observe from Fig. 1b that the bound introduced in (4) is asymptotically tight. We also observe that, in this case, the upper bound is tighter compared to the first case since the learning rate, and consequently the parameter ϵ is smaller, as suggested by our theoretical results.

In this section, we illustrated our theoretical results and the performance of the learning algorithm (2) through examples. We observed that the upper bound given in (4) is asymptotically tight by presenting two different examples, i.e., two different cases for the desired signal y_t and the outputs of constituent algorithms $\hat{y}_{1,t}$ and $\hat{y}_{2,t}$. We also observed that to get a tighter asymptotic bound, we require a smaller learning rate μ, as suggested by the results introduced in Section III.

V. Conclusion

In this paper, we analyze a learning algorithm [9] that adaptively combines outputs of two constituent algorithms running in parallel to model an unknown desired signal from the perspective of online learning theory and produce results in an individual sequence manner such that our results are guaranteed to hold for any bounded arbitrary signal. We relate the time-accumulated squared estimation error of this algorithm at any time to the time-accumulated squared estimation error of the optimal convex combination of the constituent algorithms that can only be chosen in hindsight. We refrain from making statistical assumptions on the underlying signals and our results are guaranteed to hold in an individual sequence manner. We also demonstrate that the proof methodology cannot be changed directly to obtain a better bound, in the convergence rate, on the performance by providing counter examples. To this end, we provide the transient, steady state and tracking analysis of this mixture in a deterministic sense without any assumptions on the underlying signals or without any approximations in the derivations. We illustrate the introduced results through examples.

References

[1] S. Wan, “Parameter incremental learning algorithm for neural networks,” *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1424–1438, November 2006.

[2] N. Liang, “A fast and accurate online sequential learning algorithm for feedforward networks,” *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, November 2006.

[3] S. Bhama, “Single layer neural networks for linear system identification using gradient descent technique,” *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 884–888, September 1993.

[4] T. C. Silva and L. Zhao, “Stochastic competitive learning in complex networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 385–398, March 2012.

[5] N. Cesa-Bianchi and L. Lugosi, *Prediction, Learning and Games*. Cambridge University Press, 2006.

[6] J. Kivinen and M. K. Warmuth, “Exponentiated gradient versus gradient descent for linear predictors,” *Journal of Information and Computation*, vol. 132, no. 1, pp. 1–62, January 1997.

[7] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, “How to use expert advice,” *Journal of the ACM*, vol. 44, no. 3, pp. 427–485, May 1997.

[8] A. C. Singer and M. Feder, “Universal linear prediction by model order weighting,” *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2685–2699, October 1999.

[9] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, “Mean-square performance of a convex combination of two adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, March 2006.

[10] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, “Steady state MSE performance analysis of mixture approaches to adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4050–4063, August 2010.

[11] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley and Sons, 2003.

[12] S. S. Kozat, “Competitive signal processing,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2004.

[13] N. Cesa-Bianchi and G. Lugosi, “On prediction of individual sequences relative to a set of experts,” in *Proceedings of the eleventh annual conference on Computational learning theory*, ser. COLT’ 98. ACM, 1998, pp. 1–11.

[14] V. Vovk, “A game of prediction with expert advice,” *Journal of Computer and System Sciences*, vol. 56, no. 2, pp. 153–173, April 1998.

[15] M. Herbster and M. K. Warmuth, “Tracking the best expert,” vol. 32, no. 2, pp. 286–294, August 1995.

[16] N. Littlestone, “Mistake bounds and logarithmic linear-threshold learning algorithms,” Ph.D. dissertation, University of California, Santa Cruz, Computer Research Laboratory, 1989.

[17] N. Cesa-Bianchi, P. M. Long, and M. Warmuth, “Worst-case quadratic loss bounds for prediction using linear functions and gradient descent,” *IEEE Transaction on Neural Networks*, vol. 7, no. 3, pp. 604–619, May 1996.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley and Sons, 1991.