AN EMPIRICAL STUDY ON ONLINE AGNOSTIC BOOSTING VIA REGRET MINIMIZATION

Sourav Sahoo
sourav.sahoo@smail.iitm.ac.in

Department of Electrical Engineering, Indian Institute of Technology Madras

ABSTRACT

Boosting is a family of machine learning algorithms based on the idea of aggregating weak learners. Although many boosting algorithms exist for both realizable and agnostic settings in the statistical learning framework, they existed only for the realizable case in the online setting. In [1], Brukhim *et al.* proposed the first agnostic online boosting algorithm that achieved sublinear regret. The algorithm efficiently converts an arbitrary online convex optimizer into an online booster. This algorithm also extends to statistical as well as online realizable setting. In this work, we briefly state the main results of [1] and conduct experiments on two different datasets to study the proposed algorithm's empirical performance. All the codes for this work are available here.

***Index Terms*—** Boosting, Online learning, Online convex optimization

1. INTRODUCTION

Boosting is a widely used machine learning technique used to convert (boost) numerous weak learners into a strong learner. Boosting was initially studied in the statistical realizable setting, which led to the famous Adaboost algorithm and multiple other algorithms with various applications [2]. It was further extended to the agnostic PAC setting where it has been extensively studied [3, 4, 5]. Recently, [6] studied boosting for online prediction in the realizable setting (mistake-bound setting). In [1], the paper we focus in this work, the first online agnostic boosting algorithm attaining sublinear regret was proposed.

Realizable boosting poses a constraint on the input sequence because there exists an expert that achieves near-zero loss on the input sequence. However, such a scenario is implausible in real-world situations with the application of online learning. Hence, there is a need to develop efficient online agnostic boosting algorithms.

We mention the preliminaries in Section 2 and briefly state the main results from [1] in Section 3. We detail the proposed method to conduct the empirical study in Section 4. We describe the experiments and results in Section 5 and finally conclude in Section 6.

Course project report for EE6180 Advanced Topics in Artificial Intelligence, Indian Institute of Technology Madras, Fall 2020. This work is primarily based on [1].

2. PRELIMINARIES

Let $\{(x_1, y_1), \ldots, (x_T, y_T)\} \in \mathcal{X} \times \{\pm 1\}$ be an (possibly adversarial and adaptive) sequence of training examples. We measure the performance in terms of *gains* rather than losses. The goal is to maximize the *correlation*, i.e., $\sum_{t=1}^{T} y_t \cdot \hat{y}_t$, where $\hat{y}_t$ is the prediction of the algorithm.

For a convex set $\mathcal{K} \subset \mathbb{R}^d$, an $(\mathcal{K}, N)$-OCO (online convex optimization [7]) is defined as: For $i \in [N]$, choose $p_i \in \mathcal{K}$. The adversary then reveals the cost $l_i(p_i)$ where $l_i(\cdot)$ is a bounded convex function over $\mathcal{K}$.

$$R_{\mathcal{A}}(T) = \sum_{i=1}^{N} l_i(p_i) - \min_{p \in \mathcal{K}} \sum_{i=1}^{N} l_i(p) \tag{1}$$

Let us define *randomized majority vote* function which will be required while stating the main algorithm. For $z \in \mathbb{R}$:

$$\Pi(z) = \begin{cases} \text{sign}(z) & \text{if } |z| \geq 1 \\ +1 & \text{w.p } \frac{1+z}{2} \\ -1 & \text{w.p } \frac{1-z}{2} \end{cases} \tag{2}$$

Definition 1 (Agnostic Weak Online Learning) *Let $\mathcal{H}$ be a class of functions, T be the horizon length and γ be the advantage. An online learning algorithm $\mathcal{W}$ is a (γ, T)-agnostic weak online learner (AWOL) for $\mathcal{H}$ if for any sequence of training examples, at each round t, the algorithm outputs $\mathcal{W}(x_t) \in \{\pm 1\}$ such that:*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{W}(x_t) y_t\right] \geq \gamma \max_{h \in \mathcal{H}} \mathbb{E}\left[\sum_{t=1}^{T} h(x_t) y_t\right] - R_{\mathcal{W}}(T) \tag{3}$$

where $R_{\mathcal{W}}(T)$ is the additive regret of weak learner $\mathcal{W}$ and the expectation is taken over the randomness of the weak learner and that of the possibly adaptive adversary.

3. THEORETICAL RESULTS

Now, we state the main result from [1]. We omit the proof here for the sake of brevity, but an interested reader may refer [1] for the detailed proof.

Theorem 2 (Agnostic Online Boosting) *Let $\mathcal{H}$ be a class of functions, T be the horizon length and $\mathcal{W}_1, \ldots, \mathcal{W}_N$ be (γ, T) AWOL for $\mathcal{H}$ and regret $R_{\mathcal{W}}(T) = o(T)$. Then there exists an online learning algorithm which has an expected regret $\mathbb{E}[R(T)]$ which satisfies:*

$$\mathbb{E}\left[R(T)\right] \leq \frac{R_{\mathcal{W}}(T)}{\gamma} + \mathcal{O}\left(\frac{T}{\gamma\sqrt{N}}\right) \tag{4}$$

Assuming $R_{\mathcal{W}}(T) \approx \sqrt{T}$ (which is true in most cases), $T \approx N$, we get $\mathbb{E}[R(T)] \lesssim \sqrt{T}$. Furthermore, if both T and N are $\mathcal{O}(1/\gamma^2\epsilon^2)$, the average regret, i.e, $\frac{\mathbb{E}[R(T)]}{T} \lesssim \epsilon$.

The proposed boosting algorithm has black-box oracle access to two auxiliary algorithms: (1) N instances of weak learners $\mathcal{W}_1, \ldots, \mathcal{W}_N$ and (2) An online convex optimizer $\mathcal{A}$. In each round t, the algorithm observes (x_t, y_t). It sequentially updates each weak learner $\mathcal{W}_i$ by feeding it (x_t, y_t^i) where y_t^i is a *randomized label*. The role of $\mathcal{A}$ is to "determine" y_t^i. Intuitively, it guides each weak learner to correct for the mistakes made by the preceding learners. The pseudocode of the main algorithm is presented in Algorithm 1.

Algorithm 1 Online Agnostic Boosting with OCO

for $t = 1, \ldots, T$ **do**
  Receive x_t
  Predict $\hat{y}_t = \Pi(\frac{1}{\gamma N}\sum_{i=1}^{N} \mathcal{W}_i(x_t)y_t)$
  for $i = 1, \ldots, N$ **do**
    if $i > 1, p_t^i = \mathcal{A}(l_t^1, \ldots, l_t^{i-1})$ else $p_t^1 = 0$
    Set loss $l_t^i(p_t^i) = p_t^i(\frac{1}{\gamma}\mathcal{W}(x_t)y_t - 1)$
    Pass (x_t, y_t^i) to $\mathcal{W}_i$ where $P(y_t^i = y_t) = \frac{1+p_t^i}{2}$
  end for
end for

For the above algorithm, we have:

$$\mathbb{E}\left[\max_{h\in\mathcal{H}} \sum_{t=1}^{T} h(x_t)y_t - \sum_{t=1}^{T} \hat{y}_t y_t\right] \leq \frac{R_{\mathcal{W}}(T)}{\gamma} + \frac{TR_{\mathcal{A}}(N)}{N} \tag{5}$$

Using *Online Gradient Descent* as the OCO algorithm, we get the regret bound as mentioned in Theorem 2.

3.1. Statistical Agnostic and Realizable Boosting

The algorithm and analysis for the online setting extend to the statistical setting following the same structure. For the sake of brevity, we will not discuss the algorithms about the statistical setting here. However, the key takeaways are:

- The regret bounds for the realizable case are inferior as compared to state-of-the-art bounds [8]. For achieving an error of ϵ, the proposed method requires $T = \mathcal{O}(\frac{1}{\gamma^2\epsilon^2})$ whereas state-of-the-art methods need $T = \mathcal{O}(\frac{1}{\gamma^2}\log\frac{1}{\epsilon})$.
- The regret bounds for the agnostic case are the same as the state-of-the-art bounds achieved in [5], but the proposed algorithm lacks adaptivity, unlike the existing methods.

4. METHODS

As described in the earlier section, the booster algorithm can access two types of auxiliary algorithms: a weak learner and an online convex optimizer. In this section, we detail the procedure in which we choose both of them.

4.1. Finding a (γ, T)-AWOL

We intend to use Hedge algorithm as the (γ, T)-AWOL. The interval $[0, 1]$ is divided to intervals of length $1/b$ each. So, we get b intervals of form $\left[\frac{i-1}{b}, \frac{i}{b}\right], i \in [b]$. Consider the midpoints of these intervals along with $1 + \frac{1}{2b}$. These $b + 1$ points serve as boundaries for each expert. For any input $x \in \mathbb{R}$, if $x < \frac{i}{b} - \frac{1}{2b}$, then experts $[0, 1, \ldots, i]$ predict $+1$ and rest predict -1. Each weak learner $\mathcal{W}_i$ observes (x_t^i, y_t) in round t, where $x_t^i = \langle x_t, v_i\rangle, i \in [N], t \in [T]$ where $v_i \in \{e_j | e_j$'s are unit vectors in $\mathbb{R}^d\}$. Finally, using the regret bound for Hedge, we get:

$$R_{\mathcal{W}}(T) = \sqrt{2T\log(b+1)} = \mathcal{O}(\sqrt{T\log b}) \tag{6}$$

4.1.1. Estimating γ

Once we have a weak learner, we need to estimate the value of γ. From Definition 1, for some K large enough, we have:

$$\begin{aligned}\gamma_i &\leq \frac{\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{W}_i(x_t)y_t\right] + R_{\mathcal{W}_i}(T)}{\max_{h\in\mathcal{H}} \mathbb{E}\left[\sum_{t=1}^{T} h(x_t)y_t\right]} \\ &\approx \frac{\frac{1}{K}\sum_{j=1}^{K}\sum_{t=1}^{T} \mathcal{W}_i(x_t^j)y_t^j + R_{\mathcal{W}_i}(T)}{\frac{1}{K}\sum_{j=1}^{K}\max_{h\in\mathcal{H}}\sum_{t=1}^{T} h(x_t^j)y_t^j}\end{aligned}$$

Hence, $\gamma = \min\{\gamma_1, \ldots, \gamma_N\}$

4.2. Online Convex Optimizer

We use online gradient descent (OGD) as the online convex optimizer. We follow the OGD algorithm with adaptive step-size as described in [7]. In all the experiments, the convex set $\mathcal{K}$ is fixed to be $[-1, +1]$. So, the regret bound for OGD as:

$$R_{\mathcal{A}}(N) = (3/2)\cdot GD\sqrt{N} = \mathcal{O}(GD\sqrt{N}) \tag{7}$$

where G is upper bound on the gradient of losses, D is the diameter of $\mathcal{K}$.

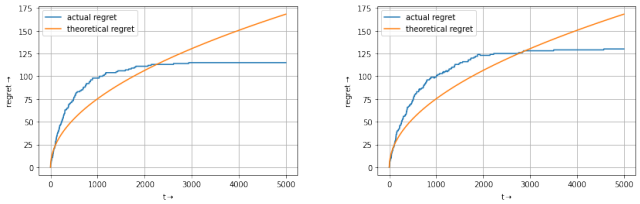
5. EXPERIMENTS AND RESULTS

5.1. Datasets

We experiment on two different datasets from UCI Machine Learning repository [9]. The first dataset, D1, is the *Optical Recognition of Handwritten Digits Dataset* and the second one, D2, is the *ISOLET Dataset*. D1 contains 5620 data points and D2 contains 7797 data points. Each data point is normalized so that $x \in [0,1]^d$. For D1, $d = 64$ whereas for D2, $d = 617$. Although, the actual datasets dealt with multi-class classification problems, we converted them into two-class classification by clustering multiple classes.

5.2. Implementing the (γ, T)-AWOL

We test the proposed candidate for the (γ, T)-AWOL on a toy problem as a proof-of-concept. We construct a dataset containing $T = 5000$ data points $(x_t, y_t) \in [0,1] \times \{\pm 1\}, t \in [T]$. We fix a limit $c^* \in [0,1]$ and the *true label* $y_t = +1$ if $x_t < c^*$ else $y_t = -1$. We randomly flip the true labels of $\sigma \in [0,1]$ fraction of the total data points to simulate the non-realizable setting. We use $b+1$ experts, where $b = 16$. In Figure 1, we compare the actual and theoretical regrets for both the noiseless case ($\sigma = 0$) and the noisy case.



(a) $\sigma = 0.0$ (Noiseless case) (c) $\sigma = 0.1$

Fig. 1: Comparison of actual and theoretical regret bound for the weak learner on the toy dataset as described in Section 5.2 for two different values of σ. Here, theoretical regret $R_{\mathcal{W}}(t) = \sqrt{2t\log(b+1)}$. Best viewed in colour.

5.3. Online Gradient Descent

As discussed in Section 4.2, we use adaptive step size in OGD to attain optimal regret bound. The step size for time t for OGD, $\eta_t = \frac{D}{G\sqrt{t}}$ [7]. In our case, $D = 2$. Now, we need to estimate G. By definition, $G = \max_{t\in[T]} \|\nabla_x l_t(x)\|$. From Algorithm 1, we have $\|\nabla_x l_t(x)\| = \left\|\frac{1}{\gamma}\mathcal{W}(x_t)y_t - 1\right\|$. It is to be noted that $\mathcal{W}(x_t)y_t \in \{\pm 1\}$. So, $\|\nabla_x l_t(x)\| = \max\{|\frac{1}{\gamma}+1|, |\frac{1}{\gamma}-1|\} \leq 2/\gamma$ as $\gamma \leq 1$. We use $G = 2/\gamma$ for calculating the step sizes. However, for calculating the exact regret of OGD, we compute G by taking the maximum of the norm of the gradients in the OCO procedure.

5.4. The main algorithm

We consider the hypothesis class $\mathcal{H}$ to be linear. Precisely, for some data point $x \in \mathbb{R}^d$ and $c \in \mathbb{R}$, we have:

$$h(x; w) = \begin{cases} +1 & \text{if } \langle w, x\rangle + c \geq 0 \\ -1 & \text{if } \langle w, x\rangle + c < 0 \end{cases} \tag{8}$$

It is to be noted that the decision boundaries of the weak learners are also linear classifiers of the form $\text{sign}(\langle x, -e_k\rangle + c^*)$, using the notations from Section 5.2. In the training process, as the weak learner sees the training examples, it tries to learn the optimal value of c^*. While estimating γ, we set the parameter $K = 100$ for D1 and $K = 10$[1] for D2. We have d different weak learners, each corresponding one coordinate of the input vector. We need to decide the number of weak learners N and how many instances of each "unique" weak learner should be considered in N. It is to be noted that even if we choose multiple identical copies of the same weak learner, randomized relabelling (line 7, Algorithm 1) ensures that each weak learner updates differently. We use $N_i, i \in [d]$ copies of each weak learner to initialize the online boosting algorithm. So, $\sum_{i=1}^{d} N_i = N$. We implement the linear classifier using scikit-learn [10].

Using the method similar to the one employed to estimate γ, for some large enough M, we have:

$$\mathbb{E}\left[\max_{h\in\mathcal{H}} \sum_{t=1}^{T} h(x_t)y_t - \sum_{t=1}^{T} \hat{y}_t y_t\right] \approx \frac{1}{M}\sum_{i=1}^{M}\left[\max_{h\in\mathcal{H}} \sum_{t=1}^{T} h(x_t^i)y_t^i - \sum_{t=1}^{T} \hat{y}_t^i y_t^i\right] \tag{9}$$

So, (5) can be approximated as:

$$\frac{1}{M}\sum_{i=1}^{M}\left[\max_{h\in\mathcal{H}} \sum_{t=1}^{T} h(x_t^i)y_t^i - \sum_{t=1}^{T} \hat{y}_t^i y_t^i\right] \leq \frac{R_{\mathcal{W}}(T)}{\gamma} + \frac{TR_{\mathcal{A}}(N)}{N} \tag{10}$$

5.4.1. *Varying T while keeping N fixed*

We fix the number of weak learners, N, and observe the empirical regret and theoretical upper bounds for different values of time horizon, T. For D1, best regret bound is obtained when $d = N = 64$ weak learners are sampled randomly from a categorical distribution proportional to their respective γ, i.e, $N_i \propto \gamma_i, i \in [d]$. For D2, we obtain the best regret bound when $d = N = 617$ and $N_i = 1, \forall i \in [d]$. The observations of the experiment are detailed in Table 1. It is to be noted that as T increases, the mean empirical regret decreases for both D1 and D2. Furthermore, the standard deviation of the regret also reduces with an increase in T, indicating that for longer sequences, the learning procedure yields less noisy results.

[1] Due to constraint in computing resources, we had to limit $K = 10$ in the higher dimensional dataset.

Table 1: Variation of empirical regret with T for a fixed N. E.R. = Empirical Regret obtained from (9) with $M = 20$ and T.U.B. = Theoretical Upper Bound is found from (6) and (7).

T	**Dataset 1** ($N = 64$)		**Dataset 2** ($N = 617$)	
	E.R.	T.U.B.	E.R.	T.U.B
100	0.847 ± 0.113	32.045	0.812 ± 0.078	20.063
200	0.784 ± 0.074	28.444	0.806 ± 0.069	16.189
500	0.683 ± 0.041	25.248	0.802 ± 0.031	12.750
1000	0.609 ± 0.026	23.637	0.735 ± 0.026	11.017
2000	0.575 ± 0.017	22.498	0.694 ± 0.012	9.792
5000	0.527 ± 0.007	21.487	0.653 ± 0.011	8.704
7500	N/A	N/A	0.642 ± 0.007	8.361

5.4.2. *Varying N while keeping T fixed*

We fix $T = 5000$ for D1 and vary the number of weak learners, N. Like the previous experiment, we sample weak learners from a categorical distribution proportional to their respective γ. The observations are described in Table 2. The results verify the fact discussed in [1] that with an increase in N, the regret bound improves (becomes small); however, the cost of computation rapidly increases, which is evident from the third column of Table 2.

Table 2: Variation of empirical regret with N for a fixed T for Dataset 1. E.R. = Empirical Regret obtained from (9) with $M = 20$ and T.U.B. = Theoretical Upper Bound is found from (6) and (7). t is the mean time taken to run Algorithm 1 for fixed N and T averaged over M runs and rounded-off to nearest integer.

N	**E.R.**	**T.U.B**	t (in s)
64	0.527 ± 0.007	21.487	35
200	0.463 ± 0.006	15.464	58
500	0.455 ± 0.006	12.718	144
1000	0.484 ± 0.006	9.732	277
2000	0.473 ± 0.005	7.621	532
5000	0.482 ± 0.006	5.748	1296

6. CONCLUSION AND FUTURE WORK

Brukhim *et al.* [1] proposed the first boosting algorithm for online agnostic boosting. In this work, we conducted experiments on two different datasets to practically verify the paper's theoretical results mentioned above. As evident from the experiments, the choice of a good weak learner is essential for obtaining an optimal regret bound. In the future, we will focus on selecting a good weak learner. From a theoretical perspective, further research can focus on obtaining stronger upper bounds.

7. REFERENCES

[1] Nataly Brukhim, Xinyi Chen, Elad Hazan, and Shay Moran, "Online agnostic boosting via regret minimization," *arXiv preprint arXiv:2003.01150*, 2020.

[2] Yoav Freund and Robert E Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[3] Dmitry Gavinsky, "Optimally-smooth adaptive boosting and application to agnostic learning," *Journal of Machine Learning Research*, vol. 4, no. May, pp. 101–117, 2003.

[4] Shai Ben-David, Philip M Long, and Yishay Mansour, "Agnostic boosting," in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 507–516.

[5] Adam Tauman Kalai and Varun Kanade, "Potential-based agnostic boosting," *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pp. 880–888, 2009.

[6] Alina Beygelzimer, Satyen Kale, and Haipeng Luo, "Optimal and adaptive algorithms for online boosting," in *International Conference on Machine Learning*, 2015, pp. 2323–2331.

[7] Elad Hazan, "Introduction to online convex optimization," *arXiv preprint arXiv:1909.05207*, 2019.

[8] Robert E Schapire and Yoav Freund, "Boosting: Foundations and algorithms," *Kybernetes*, 2013.

[9] Dheeru Dua and Casey Graff, "UCI machine learning repository," 2017.

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.