An Empirical Study on Online Agnostic Boosting via Regret Minimization

Sourav Sahoo[1]

[1]Department of Electrical Engineering, Indian Institute of Technology Madras

EE6180 Course Project

Online Agnostic Boosting via Regret Minimization

Nataly Brukhim[12], Xinyi Chen[12], Elad Hazan[12] and Shay Moran[3]

[1]Google AI Princeton, [2]Princeton University

[3]Technion - Israel Institute of Technology

NeurIPS 2020

In a nutshell..

- Brukhim *et al.*[1] proposed a novel online agnostic boosting algorithm that attains sublinear regret.
- It efficiently converts an arbitrary online convex optimizer into an online booster.
- The proposed algorithm unifies the four cases of (statistical/online) and (realizable/agnostic) setting.

[1] Nataly Brukhim et al. "Online Agnostic Boosting via Regret Minimization". In: *arXiv preprint arXiv:2003.01150* (2020).

Motivation

Why do we need *online agnostic boosting*?

Motivation

Why do we need *online agnostic boosting*?

Realizable setting poses a restriction on input sequences because it assumes that there exists an expert that achieves near-zero mistake bound.

Motivation

Why do we need *online agnostic boosting*?

Realizable setting poses a restriction on input sequences because it assumes that there exists an expert that achieves near-zero mistake bound.

In contrast, in agnostic settings, there is no restriction on input sequences and hence it can be chosen adversarially.

Motivation

Why do we need *online agnostic boosting*?

Realizable setting poses a restriction on input sequences because it assumes that there exists an expert that achieves near-zero mistake bound.

In contrast, in agnostic settings, there is no restriction on input sequences and hence it can be chosen adversarially.

Applications of Online Agnostic Boosting

- Differential Privacy
- Time Series Prediction
- Online Control

Background

- Let $\{(x_1, y_1), \ldots, (x_T, y_T)\} \in \mathcal{X} \times \{-1, +1\}$ 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.

Background

Online Convex Optimization

For a convex set $\mathcal{K} \subset \mathbb{R}^d$, an $(\mathcal{K}, N)$-OCO is defined as follows:
For $i = 1, \ldots, N$:

- Choose $p_i \in \mathcal{K}$.
- Adversary reveals cost $l_i(p_i)$ where $l_i(\cdot)$ is a bounded convex function over $\mathcal{K}$.
- $$R_{\mathcal{A}}(T) = \sum_{i=1}^{N} l_i(p_i) - \min_{p \in \mathcal{K}} \sum_{i=1}^{N} l_i(p) \tag{1}$$

Randomized Majority Vote

For $z \in \mathbb{R}$:

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

Background

Definition (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:

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

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.

Theoretical Results

Theorem (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:

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

Note:

- 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 Algorithm

- The booster has black-box oracle access to two auxiliary algorithms:
  - N instances of weak learners $\mathcal{W}_1, \ldots, \mathcal{W}_N$
  - An online convex optimizer $\mathcal{A}$

The Proposed Algorithm

- The booster has black-box oracle access to two auxiliary algorithms:
  - N instances of weak learners $\mathcal{W}_1, \ldots, \mathcal{W}_N$
  - An online convex optimizer $\mathcal{A}$
- At each round t,
  - The algorithm observes (x_t, y_t).

The Proposed Algorithm

- The booster has black-box oracle access to two auxiliary algorithms:
  - N instances of weak learners $\mathcal{W}_1, \ldots, \mathcal{W}_N$
  - An online convex optimizer $\mathcal{A}$
- At each round t,
  - The algorithm observes (x_t, y_t).
  - 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 Proposed Algorithm

- The booster has black-box oracle access to two auxiliary algorithms:
  - N instances of weak learners $\mathcal{W}_1, \ldots, \mathcal{W}_N$
  - An online convex optimizer $\mathcal{A}$
- At each round t,
  - The algorithm observes (x_t, y_t).
  - 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.

Pseudo Code

Algorithm 1 Online Agnostic Boosting with OCO

for $t = 1, \ldots, T$ **do**
  Get x_t, predict: $\hat{y}_t = \Pi\left(\frac{1}{\gamma N} \sum_{i=1}^{N} \mathcal{W}_i(x_t)\right)$.
  for $i = 1, \ldots, N$ **do**
    If $i > 1$, set $p_t^i = \mathcal{A}(\ell_t^1, ..., \ell_t^{i-1})$. \\Note that $\mathcal{A}$ is restarted at each time step t.
    Else, set $p_t^1 = 0$
    Set next loss: $\ell_t^i(p) = p(\frac{1}{\gamma}\mathcal{W}_i(x_t)y_t - 1)$.
    Pass (x_t, y_t^i) to $\mathcal{W}_i$, where y_t^i is a random label s.t. $\mathbb{P}[y_t^i = y_t] = \frac{1+p_t^i}{2}$.
  end for
end for

Figure 1: Pseudocode for the proposed algorithm.

For the above algorithm, we have:

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

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

Statistical Agnostic and Realizable Boosting

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

[2] Robert E Schapire and Yoav Freund. "Boosting: Foundations and algorithms". In: *Kybernetes* (2013).

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

Statistical Agnostic and Realizable Boosting

The algorithm and analysis for the online setting extends to the statistical setting following the same structure. For sake of brevity, we will not discuss the algorithms pertaining 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[2]. For achieving an error of ϵ, the proposed method requires $T = \mathcal{O}(\frac{1}{\gamma^2 \epsilon^2})$ whereas SOTA methods need $T = \mathcal{O}(\frac{1}{\gamma^2} \log \frac{1}{\epsilon})$.

[2] Schapire and Freund, "Boosting: Foundations and algorithms".

[3] Kalai and Kanade, "Potential-based agnostic boosting".

Statistical Agnostic and Realizable Boosting

The algorithm and analysis for the online setting extends to the statistical setting following the same structure. For sake of brevity, we will not discuss the algorithms pertaining 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[2]. For achieving an error of ϵ, the proposed method requires $T = \mathcal{O}(\frac{1}{\gamma^2 \epsilon^2})$ whereas SOTA methods need $T = \mathcal{O}(\frac{1}{\gamma^2} \log \frac{1}{\epsilon})$.
- The regret bounds for the agnostic case are same as the state-of-the-art bounds achieved earlier[3] but the proposed algorithm lacks adaptivity unlike the existing ones.

[2] Schapire and Freund, "Boosting: Foundations and algorithms".
[3] Kalai and Kanade, "Potential-based agnostic boosting".

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.

Finding a (γ, T)-AWOL

- Each weak learner $\mathcal{W}_i$ observes (x_t^i, y_t) in round t, where $x_t^i = \langle x_t, v_i \rangle, i \in [N], t \in [T]$ and $v_i \in \{e_j | e_j\text{'s are unit vectors in } \mathbb{R}^d\}$.
- Finally, using the regret bound for Hedge, we get:

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

Finding a (γ, T)-AWOL

- Each weak learner $\mathcal{W}_i$ observes (x_t^i, y_t) in round t, where $x_t^i = \langle x_t, v_i \rangle, i \in [N], t \in [T]$ and $v_i \in \{e_j | e_j\text{'s are unit vectors in } \mathbb{R}^d\}$.
- Finally, using the regret bound for Hedge, we get:

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

How do we get γ?

Estimating γ

From Definition 1, for some K large enough, we have:

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

$$\approx \frac{\frac{1}{K}\sum_{i=1}^{K}\sum_{t=1}^{T} \mathcal{W}_i(x_t)y_t + R_{\mathcal{W}_i}(T)}{\frac{1}{K}\sum_{i=1}^{K}\max_{h\in\mathcal{H}}\sum_{t=1}^{T} h(x_t)y_t}$$

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

Online Convex Optimizer

- We follow the OGD algorithm with adaptive stepsize as described in Hazan[4].
- Using regret bound for OGD, we get:

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

  where G is upper bound on the gradient of losses, D is the diameter of $\mathcal{K}$.
- In our case, $D = 2$. Furthermore, in this setting, we get $G \leq \frac{2}{\gamma}$ from the proposed algorithm.
- However, for calculating the exact regret, we compute G by taking the maximum of the norm of the gradients in the OCO procedure.

[4] Elad Hazan. "Introduction to online convex optimization". In: *arXiv preprint arXiv:1909.05207* (2019).

Datasets

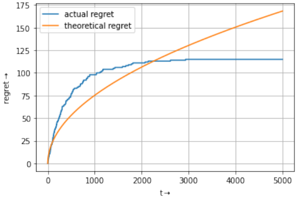
- We use two datasets from the UCI Machine Learning repository[5].
  - Dataset 1: Optical Recognition of Handwritten Digits Dataset
  - Dataset 2: ISOLET Dataset
- The datasets contain 5K and 7.5K data points respectively.
- Each data point in normalized so that $x \in [0, 1]^d$.
- For Dataset 1, $d = 64$ and for Dataset 2, $d = 617$.

[5] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

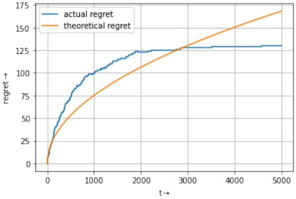
Implementing the (γ, T)-AWOL

- We construct a toy 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 $b = 16$. The results are depicted in Figure 2.

Implementing the (γ, T)-AWOL



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



(c) $\sigma = 0.1$

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

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 $\mathrm{sign}(\langle x, -e_k \rangle + c^*)$.

The Main Algorithm

- 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* 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, $\sum_{i=1}^{d} N_i = N$.

The Main Algorithm

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

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

So, the main regret bound can be approximated as:

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

Varying T while keeping N fixed

- 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.

Varying T while keeping N fixed

- 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.

Varying T while keeping N fixed

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).

$\boldsymbol{T}$	**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

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.

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 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.

Varying N while keeping T fixed

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 the main algorithm 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

Conclusion and Future Work

- In this work, we conducted experiments on two different datasets to practically verify the theoretical results obtained by Brukhim *et al.*[6].
- 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.

All the codes for this work are available at https://github.com/sourav22899/ee6180-theoretical-ml/.

[6] Brukhim et al., "Online Agnostic Boosting via Regret Minimization".

References

- Brukhim, Nataly et al. "Online Agnostic Boosting via Regret Minimization". In: *arXiv preprint arXiv:2003.01150* (2020).
- Dua, Dheeru and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.
- Hazan, Elad. "Introduction to online convex optimization". In: *arXiv preprint arXiv:1909.05207* (2019).
- Kalai, Adam Tauman and Varun Kanade. "Potential-based agnostic boosting". In: *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference* (2009), pp. 880–888.
- Schapire, Robert E and Yoav Freund. "Boosting: Foundations and algorithms". In: *Kybernetes* (2013).