Gradient Boosting on Stochastic Data Streams

Hanzhang Hu  Wen Sun  Arun Venkatraman  J. Andrew Bagnell  Martial Hebert
School of Computer Science, Carnegie Mellon University, USA
{hanzhang, wensun, arunvenk, dbagnell, hebert}@cs.cmu.edu

Abstract

Boosting is a popular ensemble algorithm that generates more powerful learners by linearly combining base models from a simpler hypothesis class. In this work, we investigate the problem of adapting batch gradient boosting for minimizing convex loss functions to online setting where the loss at each iteration is i.i.d sampled from an unknown distribution. To generalize from batch to online, we first introduce the definition of online weak learning edge with which for strongly convex and smooth loss functions, we present an algorithm, Streaming Gradient Boosting (SGB) with exponential shrinkage guarantees in the number of weak learners. We further present an adaptation of SGB to optimize non-smooth loss functions, for which we derive a $O(\ln N/N)$ convergence rate. We also show that our analysis can extend to adversarial online learning setting under a stronger assumption that the online weak learning edge will hold in adversarial setting. We finally demonstrate experimental results showing that in practice our algorithms can achieve competitive results as classic gradient boosting while using less computation.

1 INTRODUCTION

Boosting (Freund and Schapire, 1995) is a popular method that leverages simple learning models (e.g., decision stumps) to generate powerful learners. Boosting has been used to great effect and trumps other learning algorithms in a variety of applications. In computer vision, boosting was made popular by the seminal Viola-Jones Cascade (Viola and Jones, 2001) and is still used to generate state-of-the-art results in pedestrian detection (Nam et al., 2014; Yang et al., 2015; Zhu and Peng, 2016). Boosting has also found success in domains ranging from document relevance ranking (Chapelle et al., 2011) and transportation (Zhang and Haghani, 2015) to medical inference (Atkinson et al., 2012). Finally, boosting yields an anytime property at test time, which allows it to work with varying computation budgets (Grubb and Bagnell, 2012) for use in real-time applications such as controls and robotics.

The advent of large-scale data-sets has driven the need for adapting boosting from the traditional batch setting, where the optimization is done over the whole dataset, to the online setting where the weak learners (models) can be updated with streaming data. In fact, online boosting has received tremendous attention so far. For classification, (Chen et al., 2012; Oza and Russell, 2001; Beygelzimer et al., 2015b) proposed online boosting algorithms along with theoretical justifications. Recent work by (Beygelzimer et al., 2015a), addressed the regression task through the introduction of Online Gradient Boosting (OGB). We build upon the developments in (Beygelzimer et al., 2015a) to devise a new set of algorithms presented below.

In this work, we develop streaming boosting algorithms for regression with strong theoretical guarantees under stochastic setting, where at each round the data are i.i.d sampled from some unknown fixed distribution. In particular, our algorithms are streaming extension to the classic gradient boosting (Friedman, 2001), where weak predictors are trained in a stage-wise fashion to approximate the functional gradient of the loss with respect to the previous ensemble prediction, a procedure that is shown by Mason et al. (2000) to be functional gradient descent of the loss in the space of predictors. Since the weak learners cannot match the gradients of the loss exactly, we measure the error of approximation by redefining of edge of online weak learners (Beygelzimer et al., 2015b) for online regression setting.

Assuming a non-trivial edge can be achieved by each deployed weak online learner, we develop algorithms to handle smooth or non-smooth loss functions, and theo-
retically analyze the convergence rates of our streaming boosting algorithms. Our first algorithm targets strongly convex and smooth loss functions and achieves exponential decay on the average regret with respect to the number of weak learners. We show the ratio of the decay depends on the edge and also the condition number of the loss function. The second algorithm, designed for strongly convex but non-smooth loss functions, extends from the batch residual gradient boosting algorithm from (Grubb and Bagnell, 2011). We show that the algorithm achieves $O(\ln N/N)$ convergence rate with respect to the number of weak learners $N$, which matches the online gradient descent (OGD)’s no-regret rate for strongly convex loss (Hazan et al., 2007). Both of our algorithms promise that as $T$ (the number of samples) and $N$ go to infinity, the average regret converges to zero. Our analysis leverages Online-to-Batch reduction (Cesa-Bianchi et al., 2004; Hazan and Kale, 2014), hence our results naturally extends to adversarial online learning setting as long as the weak online learning edge holds in adversarial setting, a harsher setting than stochastic setting. We conclude with some proof-of-concept experiments to support our analysis. We demonstrate that our algorithm significantly boosts the performance of weak learners and converges to the performance of classic gradient boosting with less computation.

2 RELATED WORK

Online boosting algorithms have been evolving since their batch counterparts are introduced. Oza and Russell (2001) developed some of the first online boosting algorithm, and their work are applied to online feature selection (Grabner and Bischof, 2006) and online semi-supervised learning (Grabner et al., 2008). Leistner et al. (2009) introduced online gradient boosting for the classification setting albeit without a theoretical analysis. Chen et al. (2012) developed the first convergence guarantees of online learning for classification. Then Beygelzimer et al. (2015b) presented two online classification boosting algorithms that are proved to be respectively optimal and adaptive.

Our work is most related to (Beygelzimer et al., 2015a), which extends gradient boosting for regression to the online setting: each weak online learner is trained by minimizing a linear loss, and weak learners are combined using Frank-Wolfe (Frank and Wolfe, 1956) fashioned updates. Their analysis generalizes those of batch boosting for regression (Zhang and Yu, 2005). In particular, these proofs forgo edge assumptions of the weak learners. Though Frank-Wolfe is a nice projection-free algorithm, it has relatively slow convergence and usually is restricted to smooth loss functions. In our work, each weak learner minimizes the squared loss between its prediction and the gradient, which allows us to treat weak learners as approximations of the gradients thanks to the weak learner edge assumption. Hence we can mimic classic gradient boosting and use a gradient descent approach to combine the weak learners’ predictions. These differences enable our algorithms to handle non-smooth convex losses and result in convergence bounds that is more analogous to the bounds of classic batch boosting algorithms.

Our algorithms rely on the existence of weak learner edges, which is a common assumption in the classic boosting literature (Freund and Schapire, 1995, 1999) extended to the online boosting for classification by (Chen et al., 2012; Beygelzimer et al., 2015b). This assumption enables us to analyze online gradient boosting using techniques from gradient descent for convex losses (Hazan et al., 2007).

3 PRELIMINARIES

In the classic online learning setting, at every time step $t$, the learner $A$ first makes a prediction (i.e., picks a predictor $f_t \in F$), then receives a loss $\ell_t$, which could be independently sampled from a fixed distribution $D$ (the setting we mainly focus on this work) or generated by an adversary, and suffers loss $\ell_t(f_t)$ evaluated at the predictor $f_t$ picked by the learner. The learner then updates $f_t$ to $f_{t+1}$. The regret $R_A(T)$ of the learner is defined as the difference between the total loss from the learner and the total loss from the best hypothesis in hindsight under the sequence of loss $\{\ell_t\}_t$:

$$R_A(T) = \sum_{t=1}^{T} \ell_t(f_t) - \min_{f^* \in F} \sum_{t=1}^{T} \ell_t(f^*).$$

(1)

We say the online learner is no-regret if and only if $R_A(T) = o(T)$. That is, time averaged, the online learner predictor $f_t$ is doing as well as the best hypothesis $f^*$ in hindsight. We define risk of a hypothesis $f$ as $E_{x \sim D}[\ell(f)]$. Our analysis of the risk leverages the classic Online-to-Batch reduction (Cesa-Bianchi et al., 2004; Hazan and Kale, 2014). The online-to-batch reduction first analyzes the regret without considering the stochastic assumption on the sequence of loss $\ell$, and it then relates the regret to the risk using concentration of measure.

Throughout the paper we will use the concepts of strong convexity and smoothness. A function $\ell(x)$ is said to be $\lambda$-strongly convex and $\beta$-smooth with respect to norm $\|\cdot\|$ if and only if for any pair $x_1$ and $x_2$:

$$\frac{\lambda}{2} \|x_1 - x_2\|^2 \leq \ell(x_1) - \ell(x_2) - \nabla \ell(x_2)(x_1 - x_2)$$

$$\leq \frac{\beta}{2} \|x_1 - x_2\|^2.$$

(2)
where $\nabla \ell(x)$ denotes the gradient of function $\ell$ with respect to $x$.

### 3.1 Online Boosting Setup

Our online boosting setup is similar to \cite{BeygelzimerEtAl2015d} and \cite{BeygelzimerEtAl2015b}. At each time step $t = 1, \ldots, T$, the environment picks loss $\ell_t : \mathbb{R}^m \to \mathbb{R}$. The online boosting learner makes a prediction $y_t$ without knowing $\ell_t$. Then the learner suffers loss $\ell_t(y_t)$. Throughout the paper we assume the loss is bounded as $|\ell_t(y)| \leq B, B \in \mathbb{R}^+$, $\forall t, y$. We also assume that the gradient of the loss $\nabla \ell_t(y)$ is also bounded as $\|\nabla \ell_t(y)\| \leq G, G \in \mathbb{R}^+$, $\forall t, y$.

The online boosting learner maintains a sequence of weak online learning algorithms $A_1, \ldots, A_N$. Each weak learner $A_i$ can only use hypothesis from a restricted hypothesis class $\mathcal{H}$ to produce its prediction $\hat{y}_t^i = h^i_t(x_t)$, where $h^i_t \in \mathcal{H}$. To make a prediction $y_t$ at each iteration, each $A_i$ will first make a prediction $\hat{y}_t^i \in \mathbb{R}^m$ where $\hat{y}_t^i = h^i_t(x_t)$. The online boosting learner combines all the weak learners’ predictions to produce the final prediction $y_t$ for sample $x_t$. The online learner then suffers loss $\ell_t(y_t)$ after the loss $\ell_t$ is revealed. As we will show later, with the loss $\ell_t$, the online weak learner will pass a square loss to each weak learner. Each weak learner will then use its internal no-regret online update procedure to update its own weak hypothesis from $h^i_t$ to $h^i_{t+1}$. In stochastic setting where $\ell_t$ and $x_t$ are i.i.d samples from a fixed distribution, the online boosting learner will output a combination of the hypotheses that were generated by weak learners as the final boosted hypothesis for future testing.

By leveraging linear combination of weak learners, the goal of the online boosting learner is to boost the performance of a single online learner $A_i$. Additionally, we ideally want the prediction error to decrease exponentially fast in the number $N$ of weak learners, as is the result from classic batch gradient boosting \cite{GrubbBagnell2011}.

### 4 WEAK ONLINE LEARNING

As we mentioned above, each weak online learner $A_i$ only has access to a simple hypothesis class $\mathcal{H}$. Given an example $(x, y)$ pair, we specifically consider the square loss $\|y - h(x)\|^2$ ($h \in \mathcal{H}$) for each weak online learner $A_i$. At every time step $t$, $A_i$ chooses a predictor $h_t \in \mathcal{H}$, receives $(y_t, x_t)$ and then suffers loss $\|y_t - h_t(x_t)\|^2$. With this, we now introduce the definition of Weak Online Learning Edge.

**Definition 4.1. (Weak Online Learning Edge)**

Given a restricted hypothesis class $\mathcal{H}$ and a sequence of square losses $\{\|y_t - h(x_t)\|^2\}_t$, the weak online learner predicts a sequence $\{h_t\}$ that has edge $\gamma \in (0,1]$, such that with high probability $1 - \delta$:

$$\sum_t \|y_t - h_t(x_t)\|^2 \leq (1 - \gamma) \sum_t \|y_t\|^2 + R(T),$$

where $R(T) \in o(T)$ is the excess loss.

The high probability $1 - \delta$ comes from the possible randomness of the weak online learner and the sequence of examples. Usually the dependence of the high probability bound on $\delta$ is poly-logarithmic in $1/\delta$ that is included in the term $R(T)$. We will give a concrete example on this edge definition in next section where we will show what $R(T)$ consists of. Intuitively, a larger edge implies that the hypothesis is able to better explain the variance of the learning targets $y$. Our online weak learning definition is closely related to the one from \cite{BeygelzimerEtAl2015b} in that our definition explicitly makes the following two assumptions: (1) the online learning problem is agnostic-learnable (i.e., each weak learner is no-regret) with high probability:

$$\sum_{t=1}^T \|y_t - h_t(x_t)\|^2 \leq \min_{h \in \mathcal{H}} \sum_{t=1}^T \|y_t - h(x_t)\|^2 + o(T),$$

and (2) the restricted hypothesis class $\mathcal{H}$ is rich enough such that for any sequence of $(y_t, x_t)$ with high probability:

$$\min_{h \in \mathcal{H}} \sum_{t=1}^T \|y_t - h(x_t)\|^2 \leq (1 - \gamma) \sum_{t=1}^T \|y_t\|^2 + o(T).$$

Our definition of online weak learning directly generalizes the batch weak learning definition in \cite{GrubbBagnell2011} to the online setting by the additional agnostic learnability assumption as shown in Eqn. [4].

Note that we pick square loss in our weak online learning definition. As we will show later, the goal is to enforce that the weak learners to accurately predict gradients, as was also originally used in the batch gradient boosting algorithm \cite{Friedman2001}. As the authors of \cite{Friedman2001} claimed least-squares (Eqn. 5) is a natural choice owing to the superior computational properties of many least-squares algorithms.

The above online weak learning edge definition immediately implies the following result, which is used in later proofs:

**Lemma 4.2.** Given the sequence of losses $\|y_t - h(x_t)\|^2$, $1 \leq t \leq T$, the online weak learner gener-
Gradient Boosting on Stochastic Data Streams

ates a sequence of predictors \( \{h_i\}_t \), such that:

\[
\sum_{t=1}^{T} 2y_t^T h_i(x_t) \geq \gamma \sum_{t=1}^{T} \|y_t\|^2 - R(T), \quad \gamma \in (0, 1]. \tag{6}
\]

The above lemma can be proved by expanding the square on the LHS of Eqn. [3] cancelling common terms and rearranging terms.

4.1 Why Weak Learner Edge is Reasonable?

We demonstrate here that the weak online learning edge assumption is reasonable. Let us consider the case that the hypothesis class \( \mathcal{H} \) is closed under scaling (meaning if \( h \in \mathcal{H} \), then for all \( \alpha \in \mathbb{R}, \alpha h \in \mathcal{H} \)) and let us assume \( x \sim D \), and \( y = f^*(x) \) for some unknown function \( f^* \). We define the inner product \( \langle h_1, h_2 \rangle \) of any two functions \( h_1, h_2 \) as \( \mathbb{E}_{x \sim D}[h_1(x)h_2(x)] \) and the squared norm \( \|h\|^2 \) of any function \( h \) as \( \langle h, h \rangle \). We assume \( f^* \) is bounded in a sense \( \|f^*(x)\| \leq F \in \mathbb{R}^+ \). The following proposition shows that as long as \( f^* \) is not perpendicular to the span of \( \mathcal{H} \), the span of \( \langle h, f^* \rangle \neq 0 \), then we can achieve a non-zero edge:

**Proposition 4.3.** Consider any sequence of pairs \( \{x_t, y_t\}_{t=1}^T \), where \( x_t \) is i.i.d. sampled from \( D \), \( y_t = f^*(x_t) \) and \( f^* \notin \text{span}(\mathcal{H}) \). Run any no-regret online algorithm \( A \) on sequence of losses \( \{\|y_t - h(x_t)\|^2\}_t \) and output a sequence of predictions \( \{h_i\}_t \). With probability at least \( 1 - \delta \), there exists a weak online learning edge \( \gamma \in (0, 1] \), such that:

\[
\sum_{t=1}^{T} \|h_i(x_t) - y_t\|^2 \leq (1 - \gamma) \sum_{t=1}^{T} \|y_t\|^2 + R_A(T) + (2 - \gamma)O\left(\sqrt{T \ln(1/\delta)}\right),
\]

where \( R_A(T) \) is the regret of online algorithm \( A \).

The proof of the above proposition can be found in Appendix. Matching to Eq. we have \( R(T) = R_A(T) + (2 - \gamma)O\left(\sqrt{T \ln(1/\delta)}\right) \in o(T) \). Note that when \( f^* \perp \text{span}(\mathcal{H}) \), batch gradient boosting algorithms \cite{Grubb and Bagnell 2011} will essentially fail as well.

5 ALGORITHM

5.1 Smooth Loss Functions

We first present Streaming Gradient Boosting (SGB), an algorithm (Alg. 1) that is designed for loss functions \( \ell(y) \) that are \( \lambda \)-strongly convex and \( \beta \)-smooth. Alg. 2 is the online version of the classic batch gradient boosting algorithms \cite{Friedman 2001, Grubb and Bagnell 2011}. Alg. 1 maintains \( N \) weak learners. At each time step \( t \), given example \( x_t \), the algorithm predicts \( y_t \) by linearly combining the weak learners’ predictions (Line 5). Then after receiving loss \( \ell_i \), for each weak learner, the algorithm computes the gradient of \( \ell_i \) with respect to \( y \) evaluated at the partial sum \( y_{t-1} \) (Line 11) and feeds the square loss \( \ell_i(h) \) with the computed gradient as the regression target to weak learner \( A^i \) (Line 12). The weak learner \( A^i \) then performs its own no-regret online update to compute \( h_i^{t+1} \) (Line 13).

Line 16 and 17 are needed for stochastic setting. We compute the average \( \bar{h}_i \) for every \( i \) in Line 16. In testing time, given \( x \sim D \), we predict \( y \) as:

\[
y = y_0 - \eta \sum_{i=1}^{N} \bar{h}_i(x). \tag{7}
\]

Since we penalize the weak learners by the squared deviation of its own prediction and the gradient from the previous partial sum, we essentially force weak learners to produce predictions that are close to the gradients (in a no-regret perspective). With this perspective, SGB can be understood as using the weak learners’ predictions as \( N \) gradient descent steps where the gradient of each step \( i \) is approximated by a weak learner’s prediction (Line 5). Let us define \( \Delta_0 = \sum_{t=1}^{T} (\ell_t(y_t^0) - \ell_t(f^*(x_t))) \), for any \( f^* \in F \). Namely \( \Delta_0 \) measures the performance of the initialization \( \{y_0^t\}_t \). Under our assumption that the loss is bounded, \( |\ell_t(x)| \leq B, \forall t, x \), we can simply upper bound
\(\Delta_0 \leq 2BT\). Alg. 1 has the following performance guarantee:

**Theorem 5.1.** Assume weak learner \(A_i, \forall i\) has weak online learning edge \(\gamma \in (0, 1]\). Let \(f^* = \arg\min_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell_t(f(x_t))\). There exists a \(\eta = \frac{\gamma}{2(\beta s - \delta)}\), for \(\lambda\)-strongly convex and \(\beta\)-smooth loss functions, such that when \(T \to \infty\), Alg. 1 generates a sequence of predictions \(\{y_t\}_i\) where:

\[
\frac{1}{T} \left| \sum_{t=1}^{T} \ell_t(y_t) - \sum_{t=1}^{T} \ell_t(f^*(x_t)) \right| \leq 2B(1 - \frac{\gamma^2 \lambda}{16\beta})^N. \tag{8}
\]

For stochastic setting where \((x_i, \ell_i) \sim D\) independently, we have when \(T \to \infty\):

\[
\mathbb{E}[\ell(y_0 - \eta \sum_{i=1}^{N} h_i(x)) - \ell(f^*(x))] \leq 2B(1 - \frac{\gamma^2 \lambda}{16\beta})^N. \tag{9}
\]

The expectation in Eqn. 9 of the above theorem is taken over the randomness of the sequence of pairs of loss and samples \(\{\ell_t, x_t\}_{t=1}^{T}\) (note that \(h_i\) is dependent on \(\ell_1, x_1, ..., \ell_T, x_T\) and \(\ell, x\)). Theorem 5.1 shows that with infinite amount samples the average regret decreases exponentially as we increase the number of weak learners. This performance guarantee is very similar to classic batch boosting algorithms [Schapire and Freund, 2012; Grubb and Bagnell, 2011], where the empirical risk decreases exponentially with the number of algorithm iterations, i.e., the number of weak learners. Theorem 5.1 mirrors that of Theorem 1 in [Beygelzimer et al. 2015a], which bounds the regret of the Frank-Wolfe-based Online Gradient Boosting algorithm. Our results utilize the additional assumptions that the loss is strongly convex and that the weak learners have edge, allowing us to shrink the average regret exponentially with respect to \(N\), while the average regret in [Beygelzimer et al. 2015a] shrinks in the order of \(1/N\) (though this dependency on \(N\) is optimal under their setting).

Proof of Theorem 5.1 detailed in Appendix B weaves our additional assumptions into the proof framework of gradient descent on smooth losses. In particular, using weak learner edge assumption, we derive Lemma 4.2 and the Lemma 3.1 to relate parts of the strong smoothness expansion of the losses to the norm-squared of the gradients \(\|\nabla \ell_t(y_t)\|^2\), which is an upper bound of \(2\lambda(\ell_t(y_t) - \ell_t(f^*(x_t)))\) due to strong convexity. Using this observation, we can relate the total regret of the ensemble of the first \(i\) learners, \(\Delta_i = \sum_{t=1}^{T} \ell_t(y_t) - \ell_t(f^*(x_t))\), with the regret from using \(i + 1\) learners, \(\Delta_{i+1}\), and show that \(\Delta_{i+1}\) shrinks \(\Delta_i\) by a constant fraction while only adding a small term \(O(R(T)) \in o(T)\). Solving the recursion on the sequence of \(\Delta_i\), we arrive at the final exponentially decaying regret bound in the number of learners.

**Remark** Due to the weak online learning edge assumption, the regret bound shown in Eqn. 8 and the risk bound shown in Eqn. 9 are stronger than typical bounds in classic online learning, in a sense that we are competing against \(f^*\) that could potentially be much more powerful than any hypothesis from \(\mathcal{H}\). For instance when the loss function is square loss \(\ell(f(x)) = \|f(x) - \hat{x}\|^2\), Theorem 5.1 essentially shows that the risk of the boosted hypothesis \(\mathbb{E}[\|y_0 - \sum_{i=1}^{N} h_i(x) - \hat{x}\|^2]\) approaches to zero as \(N\) approaches to infinity, under the assumption that \(A_i, \forall i\) have no-zero weak learning edge. Note that this is analogous to the results of classification based batch boosting [Freund and Schapire, 1995; Grubb and Bagnell, 2011] and online boosting [Beygelzimer et al., 2015b]: as number of weak learners increase, the average number of prediction mistakes approaches to zero. In other words, with the corresponding edge assumptions, these batch/online boosting classification algorithms can compete against any arbitrarily powerful classifier that always makes zero mistakes on any given training data.

### 5.2 Non-smooth Loss Functions

The regret bound shown in Theorem 5.1 only applies for strongly convex and smooth loss functions. In fact, one can show that Alg. 1 will fail for general non-smooth loss functions. We can construct a sequence of non-smooth loss functions and a special weak hypothesis class \(\mathcal{H}\) which together show that the regret of Alg. 1 grows linearly in the number of samples, regardless of the number of weak learners. We refer readers to Appendix D for more details.

Our next algorithm, Alg. 2, extends SGB (Alg. 1) to handle strongly convex but non-smooth losses. Instead of training each weak learner to fit the subgradients of non-smooth loss with respect to current prediction, we instead keep track of a residual \(\Delta_i^k\) that accumulates the difference between the subgradients, \(\nabla_k\), and the fitted prediction \(h_k(x_i)\), from \(k = 1\) up to \(i - 1\). Instead of fitting the predictor \(h_{i+1}\) to match the subgradient \(\nabla_{i+1}\), we fit it to match the sum of the subgradient and the residuals, \(\nabla_{i+1} - \Delta_i\). More specifically, in Line 13 of Alg. 2 for each weak learner \(A_i\), we feed a square loss with the sum of residual and the gradient as the regression target. We then update the residual by computing the difference between the regression target \((\Delta_i^k + \nabla_i^k)\) and the weak learner \(A_i\’s\) prediction \(h_i^k(x_i)\) (Line 14).

The last line of Alg. 2 is needed for stochastic setting.

---

Note the abusive notation. For the non-smooth loss setting (Alg. 2), \(\Delta_i\) does not refer to the regret of the ensemble’s regret with the \(i\)-th as used in the analysis of Alg. 1.
Algorithm 2 Streaming Gradient Boosting (SGB) for non-smooth loss (Residual Projection)

1: **Input:** A restricted class $\mathcal{H}$. $N$ online weak learners $\{A_i\}_{i=1}^N$. Learning rate schedule $\{\eta_i\}_{i=1}^N$.
2: \( \forall t, A_t \) initializes a hypothesis $h_t^i \in \mathcal{H}$.
3: for \( t = 1 \) to \( T \) do
4: \( \text{Receive } x_t \) and initialize $y_t^0 = y_0$ (e.g., $y_0 = 0$).
5: \( \text{for } i = 1 \) to \( N \) do
6: \( \text{Set the projected partial sum } y_t^i = \Pi_y(y_t^{i-1} - \eta_i h_t^i(x_t)). \)
7: end for
8: Predict $y_t = \frac{1}{N} \sum_{i=0}^N y_t^i$.
9: The loss $\ell_t$ is revealed and compute loss $\ell_t(y_t)$.
10: Set initial residual $\Delta_0 = 0$.
11: for \( i = 1 \) to \( N \) do
12: \( \text{Compute subgradient w.r.t. partial sum: } \nabla_t^i = \nabla_x \ell_t(y_t^{i-1}). \)
13: \( \text{Feed loss } ||(\Delta_{t-1}^i + \nabla_t^i) - h(x))||^2 \text{ to } A_t. \)
14: Update residual: $\Delta_t^i = \Delta_{t-1}^i + \nabla_t^i - h_t^i(x_t)$.
15: Weak learner $A_t$ computes $h_t^{i+1}$ using its no-regret update procedure.
16: end for
17: end for
18: **Return:** $h_t^i, 1 \leq i \leq N, 1 \leq t \leq T$.

**Algorithm 3 SGB (Residual Projection) for testing**

1: **Input:** Test sample $x$ and $h_t^i, 1 \leq i \leq N, 1 \leq t \leq T$ from the output of Alg. 2.
2: for $t = 1$ to $T$ do
3: \( \text{for } i = 1 \) to \( N \) do
4: \( y_t^i = \Pi_y(y_t^{i-1} - \eta_i h_t^i(x_t)). \)
5: end for
6: \( y_t = \frac{1}{N} \sum_{i=0}^N y_t^i. \)
7: end for
8: **Predict:** $y = \mathcal{T}(x) = \frac{1}{T} \sum_{t=1}^T y_t$.

where $(\ell_t, x_t) \sim D$ i.i.d. In test time, given sample $x \sim D$, we predict $y$ using $h_t^i, \forall i, t$ in procedure shown in Alg. 3. For notation simplicity, we denote the testing procedure shown in Alg. 3 as $\mathcal{T}(x)$, which $\mathcal{T}$ explicitly depends on the returns $h_t^i, 1 \leq i \leq N, 1 \leq t \leq T$ from SGB (Residual Projection). Since it’s impractical to store and apply all $TN$ models, we follow previous stochastic learning techniques by Johnson and Zhang (2013) and use the final predictor at time $T$ for testing in the experiment section. Alternatively, one can record predictors from the last few time-steps, and average their predictions.

Intuitively, this approach prevents the weak learners from consistently failing to match a certain direction of the subgradient as the net error in the direction is stored in residual. By the assumption of weak learner edge, the directions will be approximated. We also note that if we assume the subgradients are bounded, then the residual magnitudes increase at most linearly in the number of weak learners. Simultaneously, each weak learner shrinks the residual by at least a constant factor due to the assumption of edge. Hence, we expect the residual to shrink exponentially in the number of learners. Utilizing this observation, we arrive at the following performance guarantee:

**Theorem 5.2.** Assume the loss $\ell_t$ is $\lambda$-strongly convex for all $t$ with bounded gradients, $||\nabla \ell_t(y)|| \leq G$ for all $y$, and each weak learner $A_t$ has edge $\gamma \in (0, 1)$. Let $F$ be a function space, and $\mathcal{H} \subset F$ be a restriction of $F$. Let $f^* = \arg \min_{f \in F} \frac{1}{T} \sum_{t=1}^T \ell_t(f(x_t))$ be the optimal predictor in $F$ in hindsight. Let $c = \frac{\gamma}{2} - 1$. Let step size be $\eta_t = \frac{1}{N}$. When $T \to \infty$, we have:

$$\frac{1}{T} \sum_{t=1}^T (\ell_t(y_t) - \ell_t(f^*(x_t))) \leq \frac{4c^2G^2}{\lambda N}(1 + \ln N + \frac{1}{8N}).$$

For stochastic setting where $(x_t, \ell_t) \sim D$ independently, when $T \to \infty$ we have:

$$E[\ell(\mathcal{T}(x)) - \ell(f^*(x))] \leq \frac{4c^2G^2}{\lambda N}(1 + \ln N + \frac{1}{8N}).$$

The above theorem shows that the average regret of Alg. 2 is $O(\ln N/N)$ with respect to the number $N$ of weak learners, which matches the regret bounds of Online Gradient Descent for strongly convex loss. The key idea for proving Theorem 5.2 is to combine our online weak learning edge definition with the proof framework of Online Gradient Descent for strongly convex loss functions from Hazan et al. (2007). The detailed proof can be found in Appendix C.

## 6 Experiments

We demonstrate the performance of our Streaming Gradient Boosting using the following UCI datasets (Lichman, 2013): YEAR, ABALONE, SLICE, and A9A (Kohavi and Becker) as well as the MNIST (LeCun et al., 1998) dataset. If available, we use the given train-test split of each data-set. Otherwise, we create a random 90%-10% train-test split.

### 6.1 Experimental Analysis of Regret Bounds

We first demonstrate the relationships between the regret bounds shown in Eqn. S and the parameters including the number of weak learners, the number of samples and edge $\gamma$. We compute the regret of SGB with respect to a separate powerful learner which plays the $f^*$ in Eqn. S. We use regression trees as the weak
learners and show how the regret varies as a function of the trees’ depth (depth empirically relates to edge $\gamma$) as well as over the number $N$ of weak learners (Fig. 1a) and the number of samples $T$ (Fig. 1b).

For the experimental results shown in Fig. 1 we used smooth loss functions with $L_2$ regularization (see Appendix E for more details). For binary classification (A9A), we use logistic loss and for regression task (SLICE), we used square loss. For each regression tree weak learner, Follow The Regularized Leader (FTRL) (Shalev-Shwartz 2011) was used as the no-regret online update algorithm with regularization posed as the depth of the tree. Fig. 1a shows the relationship between the number of weak learners and the average regret given a fixed total number of samples. The average regret decreases as we increase the number of weak learners. We note that the curves are close to linear at the beginning, matching our theoretical analysis that the average regret decays exponentially (note the y-axis is log scale) with respect to the number of weak learners. This shows that SGB can significantly boost the performance of a single weak learner.

To investigate the effect of the edge parameter $\gamma$, we additionally compute the average regret in Fig. 1 as the depth of the regression tree is increased. The tree depth increases the model complexity of the base learner and should relate to a larger $\gamma$ edge parameter. From this experiment, we see that the average regret shrinks as the depth of the trees increases.

Finally, Fig. 1b shows the convergence of the average regret with respect to the number of samples. We again see that more powerful weak learners (deeper regression trees with potentially larger $\gamma$) results faster convergence of our algorithm. We ran Alg. 2 on A9A with hinge loss and SLICE with $L_1$ (least absolute deviation) loss and observed very similar results as shown in Fig. 1b.

6.2 Batch Boosting vs. Streaming Boosting

We next compare batch boosting to SGB using two-layer neural networks as weak learners\(^3\) and see that SGB reaches similar final performance as the batch boosting algorithm albeit with less training computation. Since computing the average predictors $\hat{h}_i$ that appear in Eqn. 7 is impractical, we simply use the final predictor $\hat{h}_T$ to replace $\hat{h}_i$ in Eqn. 7 for prediction. This is a common trick used in practice for stochastic optimization (e.g., (Johnson and Zhang 2013)). Our baseline, the classic batch gradient boosting (GB) (Friedman 2001), is trained using a naive online implementation: each weak learner is trained until convergence before moving on to the next weak learner. In both GB and SGB, we train weak learners using ADAM (Kingma and Ba 2015) optimization.

Since the training run-time complexity almost equates the total complexity of weak learner predictions and updates, we analyze the complexity of training SGB and GB, using the prediction complexity of one weak learner as the unit cost. Our choice of weak learner and update method (two-layer networks and ADAM) determines that updating a weak learner is about two units cost. In training using SGB, each of the $T$ data samples triggers predictions and updates with all $N$ of the weak learners. This results in a training computational complexity of $3TN = O(TN)$. For GB, let $TB$ be the samples needed for each weak learner to converge. Then the complexity of training GB is $TB \sum_{i=1}^{N} i + 2TBN \simeq \frac{1}{2}TN^2 = O(TN^2)$. This is because when training weak learner $i$, all previous $i-1$ weak learners must predict for each data point in order to update the learner $i$. Hence, SGB and GB will have the same training complexity if $TB \simeq \frac{6T}{N} = \Theta(\frac{N}{T})$. In our experiments we observe weak learners typically converge less than $\frac{N}{T}$ samples, but our following experiment shows that SGB still can converge faster overall.

\(^3\)The number of hidden units by data-set: ABALONE, A9A: 1; YEAR, SLICE: 10; MNIST: 5x5 convolution with stride of 2 and 5 output channels. Sigmoid is used as the activation for all except SLICE, which uses leaky ReLU.
Fig. 2 plots the test-time loss versus training computation, measured by the number of prediction operations done by the weak learners. Blue dots show the performances of GB when the weak learners are added. We first note that SGB successfully converges to the results of GB in all cases – evidence that SGB is a truly a streaming conversion of GB. As it takes many weak learners to achieve good performance on ABALONE and YEAR, we observe that SGB converges with less computation than GB. On A9A, however, GB is more computationally efficient than SGB. This is because the first weak learner trained by GB already performs well. Learning a single weak learner for GB is faster than simultaneously optimizing all N = 8 weak learners as SGB. This suggests that if we initially set N too big, SGB could be less computationally efficient. In fact Fig. 2 shows that very larger N causes slower convergence to the same final error plateau. On the other hand too small N (N = 3) will result worse performance. We specify the chosen N for SGB in Fig. 2, and they are around the number of weak learners that GB requires to converge and achieve good performance. We also note that SGB has slower initial progress compared to GB on SLICE in Fig. 2c and MNIST in Fig. 2e. This is an understandable result as SGB has a much larger pool of parameters to optimize. Despite this initial disadvantage, SGB surpasses GB and converges faster overall, suggesting the advantage of updating all the weak learners together. Table 1 records the test error (square error for regression and error ratio for classification) of the neural network base learner, GB, and SGB. We observe that SGB achieves test errors that are competitive with GB in all cases.

Table 1: Average test-time loss: square error for regression, and error rate for classification.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Base (regression)</th>
<th>GB (regression)</th>
<th>SGB (regression)</th>
<th>Base (classification)</th>
<th>GB (classification)</th>
<th>SGB (classification)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABALONE</td>
<td>8.2848</td>
<td>2.1411</td>
<td>2.1532</td>
<td>0.163280</td>
<td>0.019320</td>
<td>0.016320</td>
</tr>
<tr>
<td>YEAR</td>
<td>4.99 x 10^7</td>
<td>42.8976</td>
<td>43.0573</td>
<td>0.1547</td>
<td>0.1579</td>
<td>0.1523</td>
</tr>
<tr>
<td>SLICE</td>
<td>0.036045</td>
<td>0.000755</td>
<td>0.000713</td>
<td>0.163280</td>
<td>0.019320</td>
<td>0.016320</td>
</tr>
<tr>
<td>A9A (classification)</td>
<td>0.1547</td>
<td>0.1579</td>
<td>0.1523</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MNIST</td>
<td>0.163280</td>
<td>0.019320</td>
<td>0.016320</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7 CONCLUSION

In this paper, we present SGB for online convex programming. By introducing an online weak learning edge definition that naturally extends the edge definition from batch boosting to the online setting and by using square loss, we are able to boost the predictions from weak learners in a gradient descent fashion. Our SGB algorithm guarantees exponential regret shrinkage in the number N of weak learners for strongly convex and smooth loss functions. We additionally extend SGB for optimizing non-smooth loss function, which achieves O(ln N/N) no-regret rate. Finally, experimental results support the theoretical analysis.

Though our SGB algorithm currently utilizes the procedure of gradient descent to combine the weak learners predictions, our online weak learning definition and the design of square loss for weak learners leave open the possibility to leverage other gradient-based update procedures such as accelerated gradient descent, mirror descent, and adaptive gradient descent for combining the weak learners’ predictions.
References


Supplementary Material for Gradient Boosting on Stochastic Data Streams

A Proof of Proposition 4.3

Proof. Given that a no-regret online learning algorithm $\mathcal{A}$ running on sequence of loss $\|h(x_t) - y_t\|^2$, we have can easily see that Eqn. 4 holds as:

$$\sum_{t=1}^{T} \|h_t(x_t) - y_t\|^2 \leq \min_{h \in \mathcal{H}} \|h(x_t) - y_t\|^2 + R_\mathcal{A}(T),$$  \hspace{1cm} (11)

where $R_\mathcal{A}(T)$ is the regret of $\mathcal{A}$ and is $o(T)$. To prove Proposition 4.3, we only need to show that Eqn. 5 holds for some $\gamma \in (0,1]$. This is equivalent to showing that there exist a hypothesis $\hat{h} \in \mathcal{H}$ such that $\langle h, f^* \rangle > 0$. To see this equivalence, let us assume that $\langle h, f^* \rangle > 0$. Using Pythagorean theorem, we can see that $\|h^* - f^*\|^2 = (1 - \epsilon^2)\|f^*\|^2$. Hence we get $\gamma$ is at least $\epsilon^2$, which is in $(0,1]$.

Now since we assume that $f^* \not\perp \text{span}(\mathcal{H})$, then there must exist $h' \in \mathcal{H}$, such that $\langle f^*, h' \rangle \neq 0$, otherwise $f^* \perp \mathcal{H}$. Consider the hypothesis $h'/\|h'\|$ and $-h'/\|h'\|$ (we assume $\mathcal{H}$ is closed under scale), we have that either $\langle h', f^* \rangle > 0$ or $\langle -h', f^* \rangle > 0$. Namely, we find at least one hypothesis $h$ such that $\langle h, f^* \rangle > 0$ and $\|h\| = 1$. Hence if we pick $\hat{h} = \arg \max_{h \in \mathcal{H}, \|h\|=1} \langle h, f^*/\|f^*\| \rangle$, we must have $\langle \hat{h}, f^*/\|f^*\| \rangle = \epsilon > 0$. Namely we can find a hypothesis $h^* \in \mathcal{H}$, which is $\epsilon\|f^*/\hat{h}\|$, such that there is non-zero $\gamma \in (0,1]$:

$$\|h^* - f^*\|^2 \leq (1 - \gamma)\|f^*\|^2.$$  \hspace{1cm} (12)

To show that we can extend this $\gamma$ to the finite sample case, we are going to use Hoeffding inequality to relate the norm $\|\cdot\|$ to its finite sample approximation.

Applying Hoeffding inequality, we get with probability at least $1 - \delta/2$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} y_t \|^2 - \langle f^* \rangle \right| \leq O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right),$$  \hspace{1cm} (13)

where based on assumption that $f^*(\cdot)$ is bounded as $\|f^*(\cdot)\| \leq F$. Similarly, we have with probability at least $1 - \delta/2$:

$$\left| \frac{1}{T} \sum_{t=1}^{T} (h^*(x_t) - f^*(x_t))^2 - \|h^* - f^*\|^2 \right| \leq O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right),$$  \hspace{1cm} (14)

Apply union bound for the above two high probability statements, we get with probability at least $1 - \delta$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} y_t^2 - \langle f^*, f^* \rangle \right| \leq O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right), \text{ and},$$

$$\left| \frac{1}{T} \sum_{t=1}^{T} (h^*(x_t) - f^*(x_t))^2 - \|h^* - f^*\|^2 \right| \leq O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right).$$  \hspace{1cm} (15)

Now to prove the theorem, we proceed as follows:

$$\frac{1}{T} \sum_{t=1}^{T} \|h^*(x_t) - f^*(x_t)\|^2$$

$$\leq \|h^* - f^*\| + O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right)$$

$$\leq (1 - \gamma)\|f^*\|^2 + O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right)$$

$$\leq (1 - \gamma) \frac{1}{T} \sum_{t=1}^{T} y_t^2 + (1 - \gamma)O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right) + O\left( \sqrt{\frac{F^2}{T} \ln(4/\delta)} \right).$$  \hspace{1cm} (16)
Hence we get with probability at least $1 - \delta$:
\[
\sum_{t=1}^{T} \|h^{*}(x_{t}) - f^{*}(x_{t})\|^2 \leq \sum_{t=1}^{T} \|y_{t}\|^2 + (2 - \gamma)O(\sqrt{T \ln(1/\delta)}).
\] (17)

Set $R(T) = R_{A}(T) + (2 - \gamma)O(\sqrt{T \ln(1/\delta)}$, we prove the proposition. \hfill \Box

\section{Proof of Theorem 5.1}

An important property of $\lambda$-strong convexity that we will use later in the proof is that for any $x$ and $x^{*} = \arg \min_{x} l(x)$, we have:
\[
\|\nabla l(x)\|^2 \geq 2\lambda(l(x) - l(x^{*})).
\] (18)

We prove Eqn. [18] below.

From the $\lambda$-strong convexity of $l(x)$, we have:
\[
l(y) \geq l(x) + \nabla l(x)(y - x) + \frac{\lambda}{2}\|y - x\|^2.
\] (19)

Replace $y$ by $x^{*}$ in the above equation, we have:
\[
l(x^{*}) \geq l(x) + \nabla l(x)(x^{*} - x) + \frac{\lambda}{2}\|x^{*} - x\|^2
\]
\[
\Rightarrow 2\lambda(x^{*}) \geq 2\lambda(x) + 2\lambda\nabla l(x)(x^{*} - x) + \lambda^{2}\|x^{*} - x\|^2
\]
\[
\Rightarrow -2\lambda\nabla l(x)(x^{*} - x) - \lambda^{2}\|x^{*} - x\|^2 \geq 2\lambda(l(x) - l(x^{*}))
\]
\[
\Rightarrow \|\nabla l(x)\|^2 - \|\nabla l(x)\|^2 - 2\lambda\nabla l(x)(x^{*} - x) - \lambda^{2}\|x^{*} - x\|^2 \geq 2\lambda(l(x) - l(x^{*}))
\]
\[
\Rightarrow \|\nabla l(x)\|^2 - \|\nabla l(x)\|^2 + 2\lambda(l(x) - l(x^{*}))
\]
\[
\Rightarrow \|\nabla l(x)\|^2 \geq 2\lambda(l(x) - l(x^{*})).
\] (20)

\subsection{Proofs for Lemma 4.2}

\textit{Proof.} Complete the square on the left hand side (LHS) of Eqn. [3], we have:
\[
\sum t \|y_{t}\|^2 - 2y_{t}^{T}h_{t}(x_{t}) + \|h_{t}(x_{t})\|^2 \leq (1 - \gamma) \sum t \|y_{t}\|^2 + R(T).
\] (21)

Now let us cancel the $\sum t y_{t}^{2}$ from both side of the above inequality, we have:
\[
\sum t -2y_{t}^{T}h_{t}(x_{t}) \leq \sum t -2y_{t}^{T}h_{t}(x_{t}) + \|h_{t}(x_{t})\|^2 \leq -\gamma \sum t \|y_{t}\|^2 + R(T).
\] (22)

Rearrange, we have:
\[
\sum t 2y_{t}^{T}h_{t}(x_{t}) \geq \gamma \sum t \|y_{t}\|^2 - R(T).
\] (23)

\subsection{Proof of Theorem 5.1}

We need another lemma for proving theorem 5.1

\textbf{Lemma B.1.} For each weak learner $A_{t}$, we have:
\[
\sum t \|h_{t}^{*}(x_{t})\|^2 \leq (4 - 2\gamma) \sum t \|\nabla \ell_{t}(y_{t}^{-1})\|^2 + 2R(T).
\] (24)
Gradient Boosting on Stochastic Data Streams

Proof of Lemma [B.1] For $\sum_i (h_i^t(x_i))^2$, we have:

$$\sum_i \|h_i^t(x_i)\|^2 = \sum_i \|h_i^t(x_i) - \nabla \ell_i(y_i^{t-1}) + \nabla \ell_i(y_i^{t-1})\|^2$$

$$\leq \sum_i \|h_i^t(x_i) - \nabla \ell_i(y_i^{t-1})\|^2 + \sum_i \|\nabla \ell_i(y_i^{t-1})\|^2 + \sum_i 2(h_i^t(x_i) - \nabla \ell_i(y_i^{t-1}))^T \nabla \ell_i(y_i^{t-1})$$

$$\leq \sum_i 2\|h_i^t(x_i) - \nabla \ell_i(y_i^{t-1})\|^2 + \sum_i 2\|\nabla \ell_i(y_i^{t-1})\|^2$$

$$\leq 2(1 - \gamma) \sum_i \|\nabla \ell_i(y_i^{t-1})\|^2 + 2R(T) + 2 \sum_i \|\nabla \ell_i(y_i^{t-1})\|^2$$

(By Weak Online Learning Definition)

$$\leq (4 - 2\gamma) \sum_i \|\nabla \ell_i(y_i^{t-1})\|^2 + 2R(T). \tag{25}$$

Proof of Theorem [5.1] For $1 \leq i \leq N$, let us define $\Delta_i = \sum_{t=1}^T (\ell_i(y_i^t) - \ell_i(f^*(x_i)))$. Following similar proof strategy as shown in [Beygelzimer et al. 2015a], we will link $\Delta_i$ to $\Delta_{i-1}$. For $\Delta_i$, we have:

$$\Delta_i = \sum_{t=1}^T (\ell_i(y_i^t) - \ell_i(f^*(x_i))) = \sum_{t} \ell_i(y_i^{t-1} - \eta h_i^t(x_i)) - \sum_{t} \ell_i(f^*(x_i))$$

$$\leq \sum_{t} [\ell_i(y_i^{t-1}) - \eta \nabla \ell_i(y_i^{t-1})^T h_i^t(x_i) + \frac{\beta \eta^2}{2} \|h_i^t(x_i)\|^2] - \sum_{t} \ell_i(f^*(x_i))$$

(By $\beta$-smoothness of $\ell_i$)

$$\leq \sum_{t} \left[\ell_i(y_i^{t-1}) - \frac{\eta}{2} \|\nabla \ell_i(y_i^{t-1})\|^2 + \frac{\eta R(T)}{2} + \frac{\beta \eta^2}{2} \|h_i^t(x_i)\|^2\right] - \sum_{t} \ell_i(f^*(x_i))$$

(By Lemma [4.2])

$$\leq \sum_{t} \left[\ell_i(y_i^{t-1}) - \frac{\eta}{2} \|\nabla \ell_i(y_i^{t-1})\|^2 + \frac{\eta R(T)}{2} + \beta \eta^2(2 - \gamma) \|\nabla \ell_i(y_i^{t-1})\|^2 + \beta \eta^2 R(T) - \ell_i(f^*(x_i))\right]$$

(By Lemma [B.1])

$$= \Delta_{i-1} - \left(\frac{\eta}{2} - \beta \eta^2(2 - \gamma)\right) \sum_{t} \|\nabla \ell_i(y_i^{t-1})\|^2 + \left(\frac{\eta}{2} + \beta \eta^2\right) R(T)$$

$$\leq \Delta_{i-1} - (\eta \gamma \lambda - \beta \eta^2 \lambda(4 - 2\gamma)) \sum_{t} (\ell_i(y_i^{t-1}) - \ell_i(f^*(x_i))) + \left(\frac{\eta}{2} + \beta \eta^2\right) R(T)$$

(By Eqn. [18])

$$= \Delta_{i-1} \left[1 - (\eta \gamma \lambda - \beta \eta^2 \lambda(4 - 2\gamma))\right] + \left(\frac{\eta}{2} + \beta \eta^2\right) R(T) \tag{26}$$

Due to the setting of $\eta$, we know that $0 < (1 - (\eta \gamma \lambda - \beta \eta^2 \lambda(4 - 2\gamma))) < 1$. For notation simplicity, let us first define $C = 1 - (\eta \gamma \lambda - \beta \eta^2 \lambda(4 - 2\gamma))$. Starting from $\Delta_0$, keep applying the relationship between $\Delta_i$ and $\Delta_{i-1}$ $N$ times, we have:

$$\Delta_N = C^N \Delta_0 + \left(\frac{\eta}{2} + \beta \eta^2\right) R(T) \sum_{i=1}^N C^{i-1}$$

$$= C^N \Delta_0 + \left(\frac{\eta}{2} + \beta \eta^2\right) R(T) \frac{C^N}{1 - C}$$

$$\leq C^N \Delta_0 + \left(\frac{\eta}{2} + \beta \eta^2\right) R(T) \frac{1}{1 - C}.$$ 

Now divide both sides by $T$, and take $T$ to infinity, we have:

$$\frac{1}{T} \Delta_N = C^N \frac{1}{T} \Delta_0 \leq C^N 2B, \tag{27}$$
where we simply assume that $\ell_t(y) \in [-B, B], B \in \mathbb{R}^+$ for any $t$ and $y$. Now let us go back to $C$, to minimize $C$, we can take the derivative of $C$ with respect to $\eta$, set it to zero and solve for $\eta$, we will have:

$$
\eta = \frac{\gamma}{\beta(8 - 4\gamma)}. 
$$

(28)

Substitute this $\eta$ back to $C$, we have:

$$
C = 1 - \frac{\gamma^2 \lambda}{\beta(16 - 8\gamma)} \geq 1 - \frac{\lambda}{8\beta} \geq 1 - \frac{1}{8} = \frac{7}{8}.
$$

(29)

Hence, we can see that there exist a $\eta = \frac{\gamma}{\beta(8 - 4\gamma)}$, such that:

$$
\frac{1}{T} \Delta_N \leq 2B(1 - \frac{\gamma^2 \lambda}{\beta(16 - 8\gamma)})^N \leq 2B(1 - \frac{\gamma^2 \lambda}{16\beta})^N.
$$

(30)

Hence we prove the first part of the theorem regarding the regret. For the second part of the theorem where $\ell_t$ and $x_t$ are i.i.d sampled from a fixed distribution, we proceed as follows.

Let us take expectation on both sides of the inequality (30). The left hand side of inequality (30) becomes:

$$
\frac{1}{T} \mathbb{E} \Delta_N = \frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^{T} (\ell_t(y^*_t) - \ell_t(f^*(x_t))) \right] = \frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^{T} \ell_t(-\mu \sum_{i=1}^{N} h_t^i(x_t)) \right] - \frac{1}{T} \mathbb{E} \mathbb{E}_{(\ell_t, x_t) \sim D} [\ell_t(f^*(x_t))]
$$

where the expectation is taken over the randomness of $x_t$ and $\ell_t$. Note that $h_t^i$ only depends on $x_1, \ell_t, ..., x_{t-1}, \ell_{t-1}$. We also define $\mathbb{E}_t$ as the expectation over the randomness of $x_t$ and $\ell_t$ at step $t$ conditioned on $x_1, \ell_1, ..., x_{t-1}, \ell_{t-1}$.

Since $\ell_t, x_t$ are sampled i.i.d from $D$, we can simply write $\mathbb{E}_t[\ell_t(-\mu \sum_{i=1}^{N} h_t^i(x_t)) = \mathbb{E}_t[\ell(-\mu \sum_{i=1}^{N} h_t^i(x))]$. Now the above inequality can be simplified as:

$$
\frac{1}{T} \mathbb{E} \Delta_N = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_t[\ell_t(-\mu \sum_{i=1}^{N} h_t^i(x))] - \mathbb{E}_{(\ell, x) \sim D} \ell^*(f(x))
$$

(31)

Now use the fact that $1/T \mathbb{E} \Delta_N \leq 2B(1 - \frac{\gamma^2 \lambda}{16\beta})^N$, we prove the theorem.

\[\square\]

C Proof of Theorem 5.2

Lemma C.1. In Alg. 2, if we assume the 2-norm of gradients of the loss w.r.t. partial sums by $G$ (i.e., $\|\nabla_i \| = \| \nabla \ell_t(y^t_{i-1}) \| \leq \frac{G}{\gamma}$), and assume that each weak learner $A_i$ has regret $R(T) = o(T)$, then there exists a constant $c = \frac{1 - \gamma}{\sqrt{1 - \gamma (1 - \frac{1 + \gamma T}{\gamma})}} < \frac{2}{e} - 1$ such that

$$
\sum_{t=1}^{T} \|\Delta_t^i\|^2 \leq c^2 G^2 T \quad \text{and} \quad \sum_{t=1}^{T} \|h_t^i(x_t)\|^2 \leq (4 - 2\gamma)(1 + c)^2 G^2 T + 2R(T) \leq 4c^2 G^2 T.
$$

(33)

Proof. We prove the first inequality by induction on the weak learner index $i$. When $i = 0$, the claim is clearly true since $\Delta_0^i = 0$ for all $t$. Now we assume the claim is true for some $i \geq 0$, and prove it for $i + 1$. We first note that by the inequality $\frac{1}{T} \sum_{t=1}^{T} a_t \leq \sqrt{\frac{\sum_{t=1}^{T} a_t^2}{T}}$ for all sequence $\{a_t\}_t$, we have

$$
\frac{1}{T} \left( \sum_t \|\Delta_t^i\|^2 \right) \leq \sum_t \|\Delta_t^i\|^2 \leq c^2 G^2 T
$$

(34)
\begin{equation}
\Rightarrow (\sum_t \|\Delta^t_i\|^2) \leq c^2 G^2 T^2
\end{equation}
\begin{equation}
\Rightarrow \sum_t \|\Delta^t_i\| \leq cG T
\end{equation}

Then by the assumption that weak learner \( A_i \) has an edge \( \gamma \) with regret \( R(T) \), we have from step 14 of Alg. 2.

\begin{equation}
\sum_t \|\Delta^t_{i+1}\|^2 = \sum_t \|\Delta^t_i + \nabla^t_i + h^t_i(x_i)\|^2 \leq (1 - \gamma) \sum_t \|\Delta^t_i + \nabla^t_i\|^2 + R(T)
\end{equation}
\begin{equation}
\leq (1 - \gamma) \left( \sum_t \|\Delta^t_i\|^2 + 2G \sum_t \|\Delta^t_i\|^2 + G^2 T \right) + R(T)
\end{equation}
\begin{equation}
\leq (1 - \gamma)(1 + c)G^2 T + R(T)
\end{equation}
\begin{equation}
= c^2 G^2 T
\end{equation}

We have the last equality because \( c \) is chosen as the positive root of the quadratic equation: \( \gamma c^2 + (2\gamma - 2)c + (\gamma - 1 - \frac{R(T)}{cG^2}) = 0 \), which is equivalent to \( c^2 G^2 T = (1 - \gamma)(c + 1)^2 G^2 T + R(T) \).

The second inequality of the lemma can be derived from a similar argument of Lemma B.1 by expanding \( \| (\Delta^t_{i-1} + \nabla^t_i - h^t_i(x_i)) - (\Delta^t_{i-1} + \nabla^t_i) \|^2 \) and then applying edge assumption.

We now use the above lemma to prove the performance guarantee of Alg. 2 as follows.

**Proof of Theorem 5.2.** We first define the intermediate predictors as: \( f^t_0(x) := h_0(x), f^t_i(x) := f^{t-1}(x) - \eta_i h^t_i(x), \) and \( f^t_i(x) := P(f^t_i(x)) \). Then for all \( i = 1, \ldots, N \) we have:

\begin{equation}
\|f^t_i(x_i) - f^*(x_i)\|^2 \leq \|f^t_i(x_i) - f^*(x_i)\|^2 = \|f^t_{i-1}(x_i) - \eta_i h^t_i(x_i) - f^*(x_i)\|^2
\end{equation}
\begin{equation}
= \|f^t_{i-1}(x_i) - f^*(x_i)\|^2 + \eta_i^2 \|h^t_i(x_i)\|^2 - 2\eta_i \langle f^t_{i-1}(x_i) - f^*(x_i), h^t_i(x_i) - \Delta^t_{i-1} - \nabla^t_i \rangle
\end{equation}
\begin{equation}
- 2\eta_i \langle f^t_{i-1}(x_i) - f^*(x_i), \Delta^t_{i-1} - \nabla^t_i \rangle
\end{equation}
Rearranging terms we have:

\begin{equation}
\langle f^*(x_i) - f^t_{i-1}(x_i), \nabla^t_i \rangle \geq \frac{1}{2 \eta_i} \|f^t_i(x_i) - f^*(x_i)\|^2 - \frac{1}{2 \eta_i} \|f^t_{i-1}(x_i) - f^*(x_i)\|^2 - \frac{\eta_i}{2} \|h^t_i(x_i)\|^2
\end{equation}
\begin{equation}
- \langle f^*(x_i) - f^t_{i-1}(x_i), h^t_i(x_i) - \Delta^t_{i-1} - \nabla^t_i \rangle - \langle f^*(x_i) - f^t_{i-1}(x_i), \Delta^t_{i-1} \rangle
\end{equation}

Using \( \lambda \)-strongly convex of \( \ell_i \) and applying the above equality and \( \Delta^t_i = \Delta^t_{i-1} + \nabla^t_i - h^t_i(x_i) \), we have:

\begin{equation}
\ell_i(f^*(x_i)) \geq \ell_i(f^t_{i-1}(x_i)) + \langle f^*(x_i) - f^t_{i-1}(x_i), \nabla^t_i \rangle + \frac{\lambda}{2} \|f^*(x_i) - f^t_{i-1}(x_i)\|^2
\end{equation}
\begin{equation}
\geq \ell_i(f^t_{i-1}(x_i)) + \frac{1}{2 \eta_i} \|f^t_i(x_i) - f^*(x_i)\|^2 - \frac{1}{2 \eta_i} \|f^t_{i-1}(x_i) - f^*(x_i)\|^2 - \frac{\eta_i}{2} \|h^t_i(x_i)\|^2
\end{equation}
\begin{equation}
+ \langle f^*(x_i) - f^t_{i-1}(x_i), \Delta^t_i \rangle - \langle f^*(x_i) - f^t_{i-1}(x_i), \Delta^t_{i-1} \rangle + \frac{\lambda}{2} \|f^*(x_i) - f^t_{i-1}(x_i)\|^2
\end{equation}
Summing over \( t = 1, \ldots, T \) and \( i = 1, \ldots, N \) we have:

\begin{equation}
N \sum_{t=1}^T \ell_i(f^t(x_i))
\end{equation}
\[
\geq \sum_{i=1}^{N} \sum_{t=1}^{T} \left[ \ell_t(f_{i-1}^t(x_t)) + \langle f^*(x_t) - f_{i-1}^t(x_t), \nabla f_i^t \rangle + \frac{\lambda}{2} \|f^*(x_t) - f_{i-1}^t(x_t)\|^2 \right] \\
= \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^t(x_t)) - \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\eta_i}{2} \|h_t^i(x_t)\|^2 \\
+ \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{1}{2\eta_i} \|f_t^i(x_t) - f^*(x_t)\|^2 - \sum_{i=1}^{N} \sum_{t=1}^{T} \left( \frac{1}{2\eta_i} - \frac{\lambda}{2} \right) \|f_{i-1}^t(x_t) - f^*(x_t)\|^2 \\
+ \sum_{i=1}^{N} \sum_{t=1}^{T} \langle f^*(x_t) - f_{i-1}^t(x_t), \Delta_t^i \rangle - \sum_{i=1}^{N} \sum_{t=1}^{T} \langle f^*(x_t) - f_{i-1}^t(x_t), \Delta_t^i \rangle \\
\geq \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^t(x_t)) - \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\eta_i}{2} \|h_t^i(x_t)\|^2 \\
+ \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{1}{2\eta_i} \|f_t^i(x_t) - f^*(x_t)\|^2 - \sum_{i=1}^{N-1} \sum_{t=1}^{T} \left( \frac{1}{2\eta_i} - \frac{\lambda}{2} \right) \|f_{i-1}^t(x_t) - f^*(x_t)\|^2 \\
+ \sum_{i=1}^{N-1} \sum_{t=1}^{T} \left[ \langle f^*(x_t) - f_{i-1}^t(x_t), \Delta_{i-N}^t \rangle + \frac{1}{2\eta_N} \|f_{i-N}^t(x_t) - \eta_N h_{i-N}^t(x_t) - f^*(x_t)\|^2 \right] \\
(\text{We next apply } \eta_i = \frac{1}{\lambda_i} \text{ and complete the squares for the last sum.}) \\
= \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^t(x_t)) - \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\eta_i}{2} \|h_t^i(x_t)\|^2 - \sum_{i=1}^{N} \sum_{t=1}^{T} \langle \eta_i h_t^i(x_t), \Delta_t^i \rangle \\
+ \frac{1}{2\eta_N} \sum_{t=1}^{T} \| (f_{i-N}^t(x_t) - f^*(x_t)) + \eta_N (\Delta_{i-N} - h_{i-N}^t(x_t)) \|^2 \\
- \frac{\eta N}{2} \sum_{t=1}^{T} (\|\Delta_{i-N} - h_{i-N}^t(x_t)\|^2 - \|h_{i-N}^t(x_t)\|^2) \\
(\text{We next drop the completed square, and apply Cauchy-Schwarz}) \\
\geq \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^t(x_t)) - \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\eta_i}{2} \|h_t^i(x_t)\|^2 - \sum_{i=1}^{N} \sum_{t=1}^{T} \|h_t^i(x_t)\|\|\Delta_t^i\| - \frac{\eta N}{2} \sum_{t=1}^{T} \|\Delta_{i-N}\|^2 \\
(\text{We next apply Cauchy-Schwarz again.}) \\
\geq \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^t(x_t)) - \sum_{i=1}^{N} \frac{\eta_i}{2} \sum_{t=1}^{T} \|h_t^i(x_t)\|^2 - \frac{\eta N}{2} \sum_{t=1}^{T} \|\Delta_{i-N}\|^2 \\
- \sum_{i=1}^{N} \eta_i \sqrt{\sum_{t=1}^{T} \|h_t^i(x_t)\|^2 \sum_{t=1}^{T} \|\Delta_t^i\|^2} \\
\]
Now we apply Lemma C.1 and replace the remaining η_t = \frac{1}{N\lambda}. Using \sum_{i=1}^{N} \frac{1}{\eta_i} \leq 1 + \ln N, we have:

\[
N \sum_{t=1}^{T} \ell_t(f^*(x_t)) 
\geq \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^*(x_t)) - \sum_{i=1}^{N} \frac{1}{2\lambda} 4c^2G^2T - \frac{1}{2N\lambda} c^2G^2T - \sum_{i=1}^{N} \frac{1}{\lambda} 2c^2G^2T 
\geq \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t(f_{i-1}^*(x_t)) - \frac{4c^2G^2T}{\lambda} (1 + \ln N) - \frac{c^2G^2T}{2N\lambda} 
\tag{55}
\]

Dividing both sides by NT and rearrange terms, we get:

\[
\frac{1}{TN} \sum_{i=1}^{N} \sum_{t=1}^{T} [\ell_t(y_t^i) - \ell_t(f^*(x_t))] \leq \frac{4c^2G^2}{N\lambda} (1 + \ln N) + \frac{c^2G^2}{2N^2\lambda}. 
\]

Using Jensen’s inequality for the LHS of the above inequality, we get:

\[
\frac{1}{T} \sum_{t=1}^{T} \ell_t \left( \frac{1}{N} \sum_{i=1}^{N} y_t^i \right) - \ell_t(f^*(x_t)) \leq \frac{4c^2G^2}{N\lambda} (1 + \ln N) + \frac{c^2G^2}{2N^2\lambda}, 
\]

which proves the first part of the theorem.

For stochastic setting, we can prove it by using similar proof techniques (e.g., take expectation on both sides of Eqn. 57 and use Jensen inequality) that we used for proving theorem 5.1.

\[\square\]

\section{Counter Example for Alg. 1}

In this section, we provide a counter example where we show that Alg. 1 cannot guarantee to work for non-smooth loss. We set y \in \mathbb{R}^2, and design a loss function \ell_t(y) = 2|y_{[1]}| + |y_{[2]}|, where y_{[i]} stands for the i'th entry of the vector y, for all time step t. The subgradient of this non-smooth loss is \([2, 1]^T\), \([2, -1]^T\), \([-2, 1]^T\), or \([-2, -1]^T\), depending on the position of y. We restricted the weak hypothesis class \mathcal{H} to consist of only two types of hypothesis: hypothesis h(x) = [\alpha, 0]^T, or hypothesis h(x) = [0, \alpha]^T, where \alpha \in [-2, 2]. We can show that given a sequence of training examples \{(x_t, g_t)\}_{t=1}^{T}, where g_t is the one of the gradient from the total four possible subgradient of \ell_t, the hypothesis that minimizes the accumulated square loss \sum_{\tau=1}^{T} (h(x_\tau) - g_\tau)^2 is going to be the type of h(x) = [\alpha, 0]^T.

Now we consider using Follow the Leader (FTL) as a no-regret online learning algorithm for each weak learner. Based on the above analysis, we know that no matter what the sequence of training examples each weak learner has received as far, the weak leaners always choose the hypothesis with type \(h(x) = [\alpha, 0]^T\) from \mathcal{H}. So, for every time step t, if we initialize \(y_{t}^{0} = [a, b]^T\), where \(a > 0\) and \(b > 0\), then the output \(y_{t}^{N}\) (computed from Line 8 in Alg.1) always have the form of \(y_{t}^{N} = [\eta, b]\), where \(\eta \in \mathbb{R}\). Namely, all weak learners’ prediction only moves \(y_t\) horizontally and it will never be moved vertically. But note that the optimal solution is located at \([0, 0]^T\). Since for all t, \(y_{t}^{[2]}\) is also b constant away from 0, the total regret accumulates linearly as \(bT\), regardless of how many weak learners we have.

\section{Details of Implementation}

\subsection{Binary Classification}

For binary classification, following \cite{Friedman2001}, let us define feature \(x \in \mathbb{R}^n\), label \(u \in \{-1, 1\}\). With \(x_t\) and \(u_t\), the loss function \(\ell_t\) is defined as:

\[
\ell_t(y) = \ln(1 + \exp(-u_t y)) + 4y^2. 
\tag{57}
\]

where \(y \in \mathbb{R}\). In this setting, we have \(\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}\). The regularization is to avoid overfitting: we can set \(y = \infty \ast \text{sign}(u_t)\) to make the loss close to zero.
The loss function \( \ell_t(y) \) is twice differentiable with respect to \( y \), and the second derivative is:

\[
\nabla^2 \ell_t(y) = \frac{\exp(u_t y)}{(1 + \exp(u_t y))^2}
\]

(58)

Note that we have:

\[
\nabla^2 \ell_t(y) \leq \frac{1}{1/\exp(u_t y) + 2 + \exp(u_t y)} \leq \frac{1}{4}.
\]

(59)

Hence, \( \ell_t(y) \) is 1/4-smooth.

Under the assumption that the output from hypothesis from \( \mathcal{H} \) is bounded as \( |y| \leq Y \in \mathbb{R}^+ \), we also have:

\[
\nabla^2 \ell_t(y) \geq \frac{1}{2 + 2\exp(Y)}
\]

(60)

Hence, with boundness assumption, we can see that \( \ell_t(y) \) is \( 1/(2 + 2\exp(Y)) \)-strongly convex and (1/4)-smooth.

The another loss we tried is the hinge loss:

\[
\ell_t(y) = \max(0, 1 - u_t y) + \lambda y^2.
\]

(61)

With the regularization, the loss \( \ell_t(y) \) is still strongly convex, but no longer smooth.

### E.2 Multi-class Classification

Follow the settings in [Friedman 2001], for multi-class classification problem, let us define feature \( x \in \mathbb{R}^n \), and label information \( u \in \mathbb{R}^k \), as a one-hot representation, where \( u[i] = 1 \) (\( u[i] \) is the i-th element of \( u \)), if the example is labelled by \( i \), and \( u[i] = 0 \) otherwise. The loss function \( \ell_t \) is defined as:

\[
\ell_t(y) = -\sum_{i=1}^{k} u_t[i] \ln \frac{\exp(y[i])}{\sum_{j=1}^{k} \exp(y[j])},
\]

(62)

where \( y \in \mathbb{R}^k \). In this setting, we let weak learner \( i \) pick hypothesis \( h \) from \( \mathcal{H} \) that takes feature \( x_i \) as input, and output \( \hat{y}_i \in \mathbb{R}^k \). The online boosting algorithm then linearly combines the weak learners’ prediction to predict \( y \).