How to Boost Any Loss Function

Anonymous Author(s)
Affiliation
Address
email

Abstract

Boosting is a highly successful ML-born optimization setting in which one is required to computationally efficiently learn arbitrarily good models based on the access to a weak learner oracle, providing classifiers performing at least slightly differently from random guessing. A key difference with gradient-based optimization is that boosting's original model does not requires access to first order information about a loss, yet the decades long history of boosting has quickly evolved it into a first order optimization setting – sometimes even wrongfully *defining* it as such. Owing to recent progress extending gradient-based optimization to use only a loss' zeroth (0^{th}) order information to learn, this begs the question: what loss functions can be efficiently optimized with boosting and what is the information really needed for boosting to meet the *original* boosting blueprint's requirements?

We provide a constructive formal answer essentially showing that *any* loss function can be optimized with boosting and thus boosting can achieve a feat not yet known to be possible in the classical 0^{th} order setting, since loss functions are not required to be be convex, nor differentiable or Lipschitz – and in fact not required to be continuous either. Some tools we use are rooted in quantum calculus, the mathematical field – not to be confounded with quantum computation – that studies calculus without passing to the limit, and thus without using first order information.

1 Introduction

In ML, zeroth order optimization has been devised as an alternative to techniques that would otherwise require access to $\geqslant 1$-order information about the loss to minimize, such as gradient descent (stochastic or not, constrained or not, etc., see Section 2). Such approaches replace the access to a so-called *oracle* providing derivatives for the loss at hand, operations that can be consuming or not available in exact form in the ML world, by the access to a cheaper function value oracle, providing loss values at queried points.

Zeroth order optimization has seen a considerable boost in ML over the past years, over many settings and algorithms, yet, there is one foundational ML setting and related algorithms that, to our knowledge, have not yet been the subject of investigations: boosting [33, 32]. Such a question is very relevant: boosting has quickly evolved as a technique requiring first-order information about the loss optimized [6, Section 10.3], [42, Section 7.2.2] [56]. It is also not uncommon to find boosting reduced to this first-order setting [10]. However, originally, the boosting model did not mandate the access to any first-order information about the loss, rather requiring access to a weak learner providing classifiers at least slightly different from random guessing [32]. In the context of zeroth-order optimization gaining traction in ML, it becomes crucial to understand not just whether differentiability is necessary for boosting, but more generally what are loss functions that can be boosted with a weak learner and *in fine* where boosting stands with respect to recent formal progress on lifting gradient descent to zeroth-order optimisation.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

In this paper, we settle the question: we design a formal boosting algorithm for any loss function whose set of discontinuities has zero Lebesgue measure. With traditional floating point encoding (e.g. float64), any stored loss function would *de facto* meet this condition; mathematically speaking, we encompass losses that are not necessarily convex, nor differentiable or Lipschitz. This is a key difference with classical zeroth-order optimization results where the algorithms are zeroth-order *but* their proof of convergence makes various assumptions about the loss at hand, such as convexity, differentiability (once or twice), Lipschitzness, etc. . Our trick to avoid the use of derivatives in boosting relies on using or extending tools from quantum calculus*, some of which appear to be standard in the analysis of zeroth-order optimization. To preserve readibility and save space, all proofs and additional information are postponed to an Appendix.

2 Related work

Over the past years, ML has seen a substantial push to get the cheapest optimisation routines, in general batch [15], online [28], distributed [3], adversarial [21, 19] or bandits settings [2] or more specific settings like projection-free [27, 29, 54] or saddle-point optimisation [26, 39]. We summarize several dozen recent references in Table A1 in terms of assumptions for the analysis about the loss optimized, provided in Appendix, Section I. *Zeroth-order* optimization reduces the information available to the learner to the "cheapest" one which consists in (loss) function values, usually via a so-called function value *oracle*. However, as Table A1 shows, the loss itself is always assumed to have some form of "niceness" to study the algorithms' convergence, such as differentiability, Lipschitzness, convexity, etc. . Another quite remarkable phenomenon is that throughout all their diverse settings and frameworks, not a single one of them addresses boosting. Boosting is however a natural candidate for such investigations, for two reasons. First, the most widely used boosting algorithms are first-order information hungry [6, 42, 56]: they require access to derivatives to compute examples' weights and classifiers' leveraging coefficients. Second and perhaps most importantly, unlike other optimization techniques like gradient descent, the original boosting model *does not* mandate the access to a first-order information oracle to learn, but rather to a weak learning oracle which supplies classifiers performing slightly differently from random guessing [33, 32]. Only few approaches exist to get to "cheaper" algorithms relying on less assumptions about the loss at hand, and to our knowledge do not have boosting-compliant convergence proofs, as for example when alleviating convexity [17, 48] or access to gradients of the loss [57]. Such questions are however important given the early negative results on boosting convex potentials with first-order information [38] and the role of the classifiers in the negative results [40].

Finally, we note that a rich literature has developed in mathematics as well for derivative-free optimisation [35], yet methods would also often rely on assumptions included in the three above (*e.g.* [43]). It must be noted however that derivative-free optimisation has been implemented in computers for more than seven decades [25].

3 Definitions and notations

The following shorthands are used: $[n] \doteq \{1, 2, ..., n\}$ for $n \in \mathbb{N}_*$, $z \cdot [a, b] \doteq [\min\{za, zb\}, \max\{za, zb\}]$ for $z \in \mathbb{R}, a \leqslant b \in \mathbb{R}$. In the batch supervised learning setting, one is given a training set of m examples $S \doteq \{(\boldsymbol{x}_i, y_i), i \in [m]\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is an observation ($\mathcal{X}$ is called the domain: often, $\mathcal{X} \subseteq \mathbb{R}^d$) and $y_i \in \mathcal{Y} \doteq \{-1, 1\}$ is a label, or class. We study the empirical convergence of boosting, which requires fast convergence on training. We do not investigate the questions of generalization, which would entail specific design choices about the loss at hand (see *e.g.* [9]). The objective is to learn a *classifier*, *i.e.* a function $h : \mathcal{X} \to \mathbb{R}$ which belongs to a given set $\mathcal{H}$. The goodness of fit of some h on S is evaluated from a given function $F : \mathbb{R} \to \mathbb{R}$ called a loss function, whose expectation on training is sought to be minimized:

$$F(S, h) \doteq \mathsf{E}_{i \sim [m]}[F(y_i h(\boldsymbol{x}_i))]. \tag{1}$$

The set of most popular losses comprises convex functions: the exponential loss ($F_{\text{EXP}}(z) \doteq \exp(-z)$), the logistic loss ($F^{\text{LOG}}(z) \doteq \log(1 + \exp(-z))$), the square loss ($F_{\text{SQ}}(z) \doteq (1 - z)^2$), the Hinge loss ($F_{\text{H}}(z) \doteq \max\{0, 1 - z\}$). These losses have fundamental differences in terms of their relationships to *proper losses* and *surrogate losses*, two dual views of losses related to the fact that Bayes rules is an optimal predictor for the loss [46, 47, 51]. Our examples are surrogate losses because they all define upperbounds of the 0/1-loss ($F_{0/1}(z) \doteq 1_{z \leqslant 0}$, "1" being the indicator variable), furthermore *calibrated* because their derivative in $z = 0$ is negative [8].

*Calculus "without limits" [31] (thus without using derivatives), not to be confounded with calculus on quantum devices.

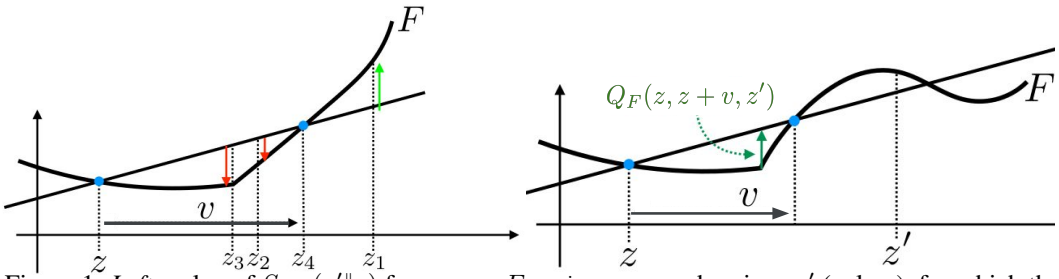


Figure 1: *Left*: value of $S_{F|v}(z'\|z)$ for convex F, $v \doteq z_4 - z$ and various z' (colors), for which the Bregman Secant distortion is positive ($z' = z_1$, green), negative ($z' = z_2$, red), minimal ($z' = z_3$) or null ($z' = z_4, z$). *Right*: depiction of $Q_F(z, z+v, z')$ for non-convex F (Definition 4.6).

4 v-derivatives, Bregman secant distortions

Unless otherwise stated, in this Section, F is a function defined over $\mathbb{R}$.

Definition 4.1. *[31] For any $z, v \in \mathbb{R}$, we let $\delta_v F(z) \doteq (F(z+v) - F(z))/v$ denote the v-derivative of F in z.*

This expression, which gives the classical derivative when the *offset* $v \to 0$, is called the h-*derivative* in quantum calculus [31, Chapter 1]. We replaced the notation for the risk of confusion with classifiers. Notice that the v-derivative is just the slope of the secant that passes through points $(z, F(z))$ and $(z+v, F(z+v))$ (Figure 1). Higher order v-derivatives can be defined [31], though we shall need a more general definition that accommodates for variable offsets.

Definition 4.2. *Let $v_1, v_2, ..., v_n \in \mathbb{R}$ and $\mathcal{V} \doteq \{v_1, v_2, ..., v_n\}$ and $z \in \mathbb{R}$. The $\mathcal{V}$-derivative $\delta_{\mathcal{V}} F$ is:*

$$\delta_{\mathcal{V}} F(z) \doteq \begin{cases} F(z) & \text{if} \quad \mathcal{V} = \varnothing \\ \delta_{v_1} F(z) & \text{if} \quad \mathcal{V} = \{v_1\} \\ \delta_{\{v_n\}}(\delta_{\mathcal{V} \setminus \{v_n\}} F)(z) & \text{otherwise} \end{cases} . \tag{2}$$

If $v_i = v, \forall i \in [n]$ then we write $\delta_v^{(n)} F(z) \doteq \delta_{\mathcal{V}} F(z)$.

In the Appendix, Lemma A computes the unravelled expression of $\delta_{\mathcal{V}} F(z)$, showing that the order of the elements in $\mathcal{V}$ does not matter. n is called the order of the $\mathcal{V}$-derivative.

We can now define a generalization of Bregman divergences called *Bregman Secant distortions*.

Definition 4.3. *For any $z, z', v \in \mathbb{R}$, the Bregman Secant distortion $S_{F|v}(z'\|z)$ with generator F and offset v is:*

$$S_{F|v}(z'\|z) \doteq F(z') - F(z) - (z' - z)\delta_v F(z). \tag{3}$$

Even if F is convex, the distortion is not necessarily positive, though it is lowerbounded (Figure 1). There is an intimate relationship between the Bregman Secant distortions and Bregman divergences. We shall use a definition slightly more general than the original one when F is differentiable [12, eq. (1.4)], introduced in information geometry [5, Section 3.4] and recently reintroduced in ML [11].

Definition 4.4. *The Bregman divergence with generator F (scalar, convex) between z' and z is $D_F(z'\|z) \doteq F(z') + F^\star(z) - z'z$, where $F^\star(z) \doteq \sup_t tz - F(t)$ is the convex conjugate of F.*

We state the link between $S_{F|v}$ and D_F (proof omitted).

Lemma 4.5. *Suppose F strictly convex differentiable. Then $\lim_{v \to 0} S_{F|v}(z'\|z) = D_F(z'\|F'(z))$.*

Relaxed forms of Bregman divergences have been introduced in information geometry [44].

Definition 4.6. *For any $a, b, \alpha \in \mathbb{R}$, denote for short $\mathbb{I}_{a,b} \doteq [\min\{a,b\}, \max\{a,b\}]$ and $(ab)_\alpha \doteq \alpha a + (1-\alpha)b$. The Optimal Bregman Information (OBI) of F defined by triple $(a,b,c) \in \mathbb{R}^3$ is:*

$$Q_F(a,b,c) \doteq \max_{\alpha : (ab)_\alpha \in \mathbb{I}_{a,c}} \{\alpha F(a) + (1-\alpha)F(b) - F((ab)_\alpha)\}. \tag{4}$$

As represented in Figure 1 (right), the OBI is obtained by drawing the line passing through $(a, F(a))$ and $(b, F(b))$ and then, in the interval $\mathbb{I}_{a,c}$, look for the maximal difference between the line and F. We note that Q_F is non negative because $a \in \mathbb{I}_{a,c}$ and for the choice $\alpha = 1$, the RHS in (4) is 0. We also note that when F is convex, the RHS is indeed the maximal Bregman information of two points in [7, Definition 2], where maximality is obtained over the probability measure. The following Lemma follows from the definition of the Bregman secant divergence and the OBI. An inspection of the functions in Figure 1 provides a graphical proof.

Lemma 4.7. *For any F,*

$$\forall z, v, z' \in \mathbb{R}, S_{F|v}(z'\|z) \geqslant -Q_F(z, z+v, z'). \tag{5}$$

and if F is convex,

$$\forall z, v \in \mathbb{R}, \forall z' \notin \mathbb{I}_{z,z+v}, S_{F|v}(z'\|z) \geqslant 0, \tag{6}$$

$$\forall z, v, z' \in \mathbb{R}, S_{F|v}(z'\|z) \geqslant -Q_F(z, z+v, z+v). \tag{7}$$

We shall abbreviate the two possible forms of OBI in the RHS of (5), (7) as:

$$Q^*_F(z, z', v) \doteq \begin{cases} Q_F(z, z+v, z+v) & \text{if } F \text{ convex} \\ Q_F(z, z+v, z') & \text{otherwise} \end{cases}. \tag{8}$$

5 Boosting using only queries on the loss

We make the assumption that all training predictions of so-called "weak classifiers" are finite and non-zero.

Assumption 5.1. $\forall t > 0, \forall i \in [m]$, $|h_t(\boldsymbol{x}_i)| \in (0, +\infty)$ *(we thus let $M_t \doteq \max_i |h_t(\boldsymbol{x}_i)|$).*

Excluding 0 ensures our algorithm does not make use of derivatives. If predictions can be zero, there is a simple tweak that still avoids the use of derivatives (Appendix, Section II.2). For short, we define two *edge* quantities for $i \in [m]$ and $t = 1, 2, ...,$

$$e_{ti} \doteq \alpha_t \cdot y_i h_t(\boldsymbol{x}_i), \quad \tilde{e}_{ti} \doteq y_i H_t(\boldsymbol{x}_i), \tag{9}$$

where α_t is a leveraging coefficient for the weak classifiers in an ensemble $H_T(.) \doteq \sum_{t \in [T]} \alpha_t h_t(.)$. We observe

$$\tilde{e}_{ti} = \tilde{e}_{(t-1)i} + e_{ti}. \tag{10}$$

5.1 Algorithm: SECBOOST

5.1.1 General steps

Without further ado, Algorithm SECBOOST presents our approach to boosting without using derivatives information. The key differences with traditional boosting algorithms are red color framed. We summarize its key steps.

Step 1 This is the initialization step. Traditionally in boosting, one would pick $h_0 = 0$. Note that $\boldsymbol{w}$ is not necessarily positive. v_0 is the initial offset (Section 4).

Step 2.1 This step calls the weak learner, as in traditional boosting, using variable "weights" on examples (the absolute value of $\boldsymbol{w}$). The key difference with traditional boosting is that examples labels can switch between iterations as well.

Step 2.3 This step computes the leveraging coefficient α_t of the weak classifier h_t. It involves a quantity, $\overline{W}_{2,t}$, which we define as any strictly positive real satisfying

$$\mathbb{E}_{i \sim [m]} \left[\delta_{\{e_{ti}, v_{(t-1)i}\}} F(\tilde{e}_{(t-1)i}) \cdot \left(\frac{h_t(\boldsymbol{x}_i)}{M_t} \right)^2 \right] \leqslant \overline{W}_{2,t}. \tag{14}$$

For boosting rate's sake, we should find $\overline{W}_{2,t}$ as small as possible. We refer to (9) for the $e_{.}, \tilde{e}_{.}$ notations; $v_{.}$ is the current (set of) offset(s) (Section 4 for their definition). The second-order $\mathcal{V}$-derivative in the LHS plays the same role as the second-order derivative in classical boosting rates, see for example [47, Appendix, Section 4]. As offsets $\rightarrow 0$, it converges to a second-order derivative; otherwise, they still share some properties, such as the sign for convex functions.

Lemma 5.2. *Suppose F convex. For any $a \in \mathbb{R}, b, c \in \mathbb{R}_*$, $\delta_{\{b,c\}} F(a) > 0$.*

(Proof in Appendix, Section II.3) We can also see a link with weights variation since, modulo a slight abuse of notation, we have $\delta_{\{e_{ti}, v_{(t-1)i}\}} F(\tilde{e}_{(t-1)i}) = \delta_{e_{ti}} w_{ti}$. A substantial difference with traditional boosting algorithms is that we have two ways to pick the leveraging coefficient α_t; the first one can be used when a convenient $\overline{W}_{2,t}$ is directly accessible from the loss. Otherwise, there is a simple algorithm that provides parameters (including $\overline{W}_{2,t}$) such that (14) is satisfied. Section 5.3

Algorithm 1 SECBOOST$(\mathcal{S}, T)$

Input sample $\mathcal{S} = \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$, number of iterations T, initial (h_0, v_0) (constant classification and offset).

Step 1 : let $H_0 \leftarrow 1 \cdot h_0$ and $\boldsymbol{w}_1 = -\delta_{v_0} F(h_0) \cdot \mathbf{1}$; // $h_0, v_0 \neq 0$ chosen s. t. $\delta_{v_0} F(h_0) \neq 0$

Step 2 : **for** $t = 1, 2, ..., T$

Step 2.1 : let $h_t \leftarrow \times(\mathcal{S}_t, |\boldsymbol{w}_t|)$ //weak learner call, $\mathcal{S}_t \doteq \{(\boldsymbol{x}_i, y_i \cdot \text{sign}(w_{ti}))\}$

Step 2.2 : let $\eta_t \leftarrow (1/m) \cdot \sum_i w_{ti} y_i h_t(\boldsymbol{x}_i)$ //unnormalized edge

Step 2.3 :

If bound on $\overline{W}_{2,t}$ available (Section 5.3)	otherwise | general procedure
pick $\varepsilon_t > 0, \pi_t \in (0,1)$ and $\alpha_t \in \frac{\eta_t}{2(1+\varepsilon_t) M_t^2 \overline{W}_{2,t}} \cdot [1 - \pi_t, 1 + \pi_t]$; (11)	$\alpha_t \leftarrow$SOLVE$_\alpha(\mathcal{S}, \boldsymbol{w}_t, h_t)$ // $\overline{W}_{2,t} > 0, \varepsilon_t > 0, \pi_t \in (0,1)$ // Theorem 5.8

Step 2.4 : let $H_t \leftarrow H_{t-1} + \alpha_t \cdot h_t$ //classifier update

Step 2.5 : **if** $\mathbb{I}_{ti}(\varepsilon_t \cdot \alpha_t^2 M_t^2 \overline{W}_{2,t}) \neq \varnothing, \forall i \in [m]$ **then** //new offsets

for $i = 1, 2, ..., m$, let

$$v_{ti} \leftarrow \text{OO}(t, i, \varepsilon_t \cdot \alpha_t^2 M_t^2 \overline{W}_{2,t}) \ ; \quad (12)$$

else return H_t;

Step 2.6 : **for** $i = 1, 2, ..., m$, let //weight update

$$w_{(t+1)i} \leftarrow -\delta_{v_{ti}} F(y_i H_t(\boldsymbol{x}_i)) \ ; \quad (13)$$

Step 2.7 : **if** $\boldsymbol{w}_{t+1} = \mathbf{0}$ **then** break;

Return H_T.

details those two possibilities and their implementation. In the more favorable case (the former one), α_t can be chosen in an interval, furthermore defined by flexible parameters $\varepsilon_t > 0, \pi_t \in (0,1)$. Note that fixing beforehand these parameters is not mandatory: we can also pick *any*

$$\alpha_t \in \eta_t \cdot \left(0, 1/(M_t^2 \overline{W}_{2,t})\right), \quad (15)$$

and then compute choices for the corresponding ε_t and π_t. ε_t is important for the algorithm and both parameters are important for the analysis of the boosting rate. From the boosting standpoint, a smaller ε_t yields a larger α_t and a smaller π_t reduces the interval of values in which we can pick α_t; both cases tend to favor better convergence rates as seen in Theorem 5.3.

Step 2.4 is just the crafting of the final model.

Step 2.5 is new to boosting, the use of a so-called offset oracle, detailed in Section 5.1.2.

Step 2.6 The weight update does not rely on a first-order oracle as in traditional boosting, but uses only loss values through v-derivatives. The finiteness of F implies the finiteness of weights.

Step 2.7 Early stopping happens if all weights are null. While this would never happen with traditional (*e.g.* strictly convex) losses, some losses that are unusual in the context of boosting can lead to early stopping. A discussion on early stopping and how to avoid it is in Section 6.

5.1.2 The offset oracle, OO

Let us introduce notation

$$\mathbb{I}_{ti}(z) \doteq \left\{v : Q_F^*(\tilde{e}_{ti}, \tilde{e}_{(t-1)i}, v) \leqslant z\right\}, \forall i \in [m], \forall z > 0. \quad (16)$$

(see Figure 3 below to visualize $\mathbb{I}_{ti}(z)$ for a non-convex F) The offset oracle is used in Step 2.5, which is new to boosting. It requests the offsets to carry out weight update in (13) to an *offset oracle*, which achieves the following, for iteration $\#t$, example $\#i$, limit OBI z:

$$\text{OO}(t, i, z) \text{ returns some } v \in \mathbb{I}_{ti}(z) \quad (17)$$

Note that the offset oracle has the freedom to pick the offset in a whole set. Section 5.4 investigates implementations of the offset oracle, so let us make a few essentially graphical remarks here. OO does not need to build the whole $\mathbb{I}_{ti}(z)$ to return some $v \in \mathbb{I}_{ti}(z)$ for Step 2.5 in SECBOOST. In the construction steps of Figure 3, as soon as $\mathcal{O} \neq \varnothing$, one element of $\mathcal{O}$ can be returned. Figure 4 presents more examples of $\mathbb{I}_{ti}(z)$. One can remark that the sign of the offset v_{ti} in Step 2.5 of SECBOOST is the same as the sign of $\tilde{e}_{(t-1)i} - \tilde{e}_{ti} = -y_i \alpha_t h_t(\boldsymbol{x}_i)$. Hence, unless F is derivable or all edges $y_i h_t(\boldsymbol{x}_i)$ are of the same sign ($\forall i$), the set of offsets returned in Step 2.5 always contain at least two different offsets, one non-negative and one non-positive (Figure 4, (a-b)).

5.2 Convergence of SECBOOST

The offset oracle has a technical importance for boosting: $\mathbb{I}_{ti}(z)$ is the set of offsets that limit an OBI for a training example (Definition 4.6). The importance for boosting comes from Lemma 4.7: upperbounding an OBI implies lowerbounding a Bregman Secant divergence, which will also guarantee a sufficient slack between two successive boosting iterations. This is embedded in a blueprint of a proof technique to show boosting-compliant convergence which is not new, see *e.g.* [47]. We now detail this convergence.

Remark that the expected edge η_t in Step 2.2 of SECBOOST is not normalized. We define a normalized version of this edge as:

$$[-1, 1] \ni \tilde{\eta}_t \doteq \sum_i \frac{|w_{ti}|}{W_t} \cdot \tilde{y}_{ti} \cdot \frac{h_t(\boldsymbol{x}_i)}{M_t}, \tag{18}$$

with$\tilde{y}_{ti} \doteq y_i \cdot \mathrm{sign}(w_{ti})$, $W_t \doteq \sum_i |w_{ti}| = \sum_i |\delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})|$. Remark that the labels are corrected by the weight sign and thus may switch between iterations. In the particular case where the loss is non-increasing (such as with traditional convex surrogates), the labels do not switch. We need also a quantity which is, in absolute value, the expected weight:

$$\overline{W}_{1,t} \doteq \left|\mathbb{E}_{i\sim[m]}\left[\delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})\right]\right| \tag{19}$$

(we indeed observe $\overline{W}_{1,t} = |\mathbb{E}_{i\sim[m]}\left[w_{ti}\right]|$) In classical boosting for convex decreasing losses[†], weights are non-negative and converge to a minimum (typically 0) as examples get the right class with increasing confidence. Thus, $\overline{W}_{1,t}$ can be an indicator of when classification becomes "good enough" to stop boosting. In our more general setting, it shall be used in a similar indicator. We are now in a position to show a first result about SECBOOST.

Theorem 5.3. *Suppose assumption 5.1 holds. Let $F_0 \doteq F(S, h_0)$ in* SECBOOST *and z^* any real such that $F(z^*) \leqslant F_0$. Then we are guaranteed that classifier H_T output by* SECBOOST *satisfies $F(S, H_T) \leqslant F(z^*)$ when the number of boosting iterations T yields:*

$$\sum_{t=1}^{T} \frac{\overline{W}_{1,t}^2 (1 - \pi_t^2)}{\overline{W}_{2,t}(1 + \varepsilon_t)} \cdot \tilde{\eta}_t^2 \geqslant 4(F_0 - F(z^*)), \tag{20}$$

where parameters ε_t, π_t appear in Step 2.3 of SECBOOST.

(proof in Appendix, Section II.4) We observe the tradeoff between the freedom in picking parameters and convergence guarantee as exposed by (20): to get more freedom in picking the leveraging coefficient α_t, we typically need π_t large (Step 2.3) and to get more freedom in picking the offset $v_t \neq 0$, we typically need ε_t large (Step 2.5). However, allowing more freedom in such ways reduces the LHS and thus impairs the guarantee in (20). Therefore, there is a subtle balance between "freedom" of choice and convergence. Figure 2 pictures notable regimes for $\overline{W}_{1,t}$ and $\overline{W}_{2,t}$, leading to varying contributions in (20).

Boosting-compliant convergence We characterize convergence in the boosting framework. We define

$$\rho_t \doteq \overline{W}_{1,t}^2 / \overline{W}_{2,t}.$$

A small $\overline{W}_{1,t}^2$ is an indicator as to whether SECBOOST is close to a minimum, as seen from the left (blue) block in Figure 2. Analysis of convergence based on a minimal value of $\overline{W}_{1,t}^2$ would be standard with respect to classical non-convex optimisation. In our case, we construct a criterion involving not just $\overline{W}_{1,t}^2$, but also $\overline{W}_{2,t}$: considering for the illustration a C^2 loss function, when $\overline{W}_{2,t}$ is small, it accounts for regions with "small second-order" variations, more "regularity", which naturally offers more leeway for minimization, in particular if $\overline{W}_{1,t}$ is large (see Figure 2). This justifies the following assumption.

Assumption 5.4. *(ρ_*-**Convergence Regime**, ρ_*-CR) We assume there exists $\rho_* > 0$ such that $\forall t \geqslant 1$, $\rho_t > \rho_*$.*

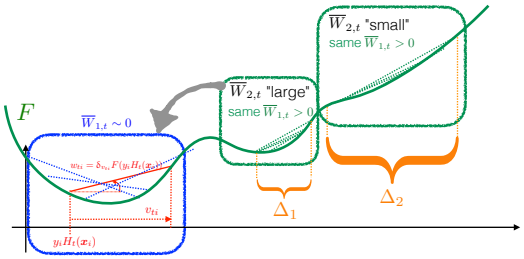


Figure 2: When SECBOOST approaches an optimum of the loss (blue + red), we can expect $\overline{W}_{1,t}$ to be small. At a fixed $\overline{W}_{1,t}$, a larger $\overline{W}_{2,t}$ indicates, at least locally near the secant points, a region of higher curvature (green blocks). Since the maximal difference of abscissae for a secant is proportional to the edge of the last weak classifier (Figure 1, right, (4) and (16)), the smaller $\overline{W}_{2,t}$ indicates a greater potential for decrease of the loss (right green block, $\Delta_2 > \Delta_1$) and leads to a greater contribution to the sum in (20). When $\overline{W}_{1,t}^2$ is small enough compared to $\overline{W}_{2,t}$, we take it as an indicator of the proximity of a local optimum, which justifies fixing a lowerbound ρ_* to ρ_t to analyze convergence "outside" this regime (best viewed in color). Bold gray arrow is discussion in Section 6.

We also rely on boosting's traditional weak learning assumption.

Assumption 5.5. *(γ-Weak Learning Assumption, γ-WLA) We assume the following on the weak learner: $\exists \gamma > 0$ such that $\forall t > 0$, $|\tilde{r}_t| \geqslant \gamma$.*

We are now in a position to state a simple corollary to Theorem 5.3.

Corollary 5.6. *Suppose assumptions 5.1, 5.4 and 5.5 hold. Let $F_0 \doteq F(S, h_0)$ in* SECBOOST *and z^* any real such that $F(z^*) \leqslant F_0$. If* SECBOOST *is run for a number T of iterations satisfying*

$$T \geqslant \frac{4(F_0 - F(z^*))}{\gamma^2 \rho_*} \cdot \frac{1 + \max_{t \in [T]} \varepsilon_t}{1 - \max_{t \in [T]} \pi_t^2}, \tag{21}$$

then $F(S, H_T) \leqslant F(z^*)$.

We remark that the dependency in γ is optimal [4].

5.3 Finding $\overline{W}_{2,t}$

There is lots of freedom in the choice of α_t in Step 2.3 of SECBOOST, and even more if we look at (15). This, however, requires access to some bound $\overline{W}_{2,t}$. In the general case, the quantity it upperbounds in (14) also depends on α_t because $e_{ti} \doteq \alpha_t \cdot y_i h_t(\boldsymbol{x}_i)$. So unless we can obtain such a "simple" $\overline{W}_{2,t}$ that does *not* depend on α_t, (11) – and (15) – provide a *system* to solve for α_t.

$\overline{W}_{2,t}$ via properties of F Classical assumptions on loss functions for zeroth-order optimization can provide simple expressions for $\overline{W}_{2,t}$ (Table A1). Consider smoothness: we say that F is β-smooth if it is derivable and its derivative satisfies the Lipschitz condition $|F'(z') - F'(z)| \leqslant \beta |z' - z|, \forall z, z'$ [13]. Notice that this implies the condition on the v-derivative of the derivative: $|\delta_v F'(z)| \leqslant \beta, \forall z, v$. This also provides a straightforward useful expression for $\overline{W}_{2,t}$.

Lemma 5.7. *Suppose that the loss F is β-smooth. Then we can fix $\overline{W}_{2,t} = 2\beta$.*

(Proof in Appendix, Section II.5) What the Lemma shows is that a bound on the v-derivative of the derivative implies a bound on order-2 $\mathcal{V}$-derivatives (in the quantity that $\overline{W}_{2,t}$ bounds (14)). Such a condition on v-derivatives is thus weaker than a condition on derivatives, and it is strictly weaker if we impose a strictly positive lowerbound on the offset's absolute value, which would be sufficient to characterize the boosting convergence of SECBOOST.

A general algorithm for $\overline{W}_{2,t}$ If we cannot make any assumption on F, there is a simple way to *first* obtain α_t and then $\overline{W}_{2,t}$, from which all other parameters of Step 2.3 can be computed. We first need a few definitions. We first generalize the edge notation appearing in Step 2.2:

$$\eta(\boldsymbol{w}, h) \doteq \mathbb{E}_{i \sim [m]} \left[w_i y_i h(\boldsymbol{x}_i) \right],$$

[†]This is an important class of losses since it encompasses the convex surrogates of symmetric proper losses [45, 52]

Algorithm 2 SOLVE$_\alpha(\mathcal{S}, \boldsymbol{w}, h)$

Input sample $\mathcal{S} = \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$, $\boldsymbol{w} \in \mathbb{R}^m$, $h : \mathcal{X} \to \mathbb{R}$.
Step 1 : find any $a > 0$ such that

$$\frac{|\eta(\boldsymbol{w}, h) - \eta(\tilde{\boldsymbol{w}}(\text{sign}(\eta(\boldsymbol{w}, h)) \cdot a), h)|}{|\eta(\boldsymbol{w}, h)|} < 1. \tag{22}$$

Return $\text{sign}(\eta(\boldsymbol{w}, h)) \cdot a$.

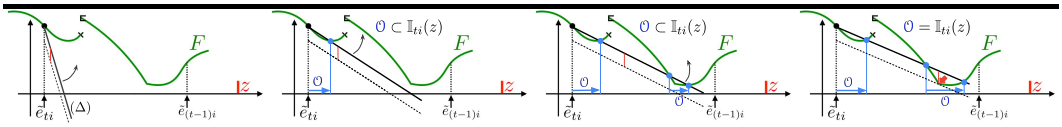


Figure 3: A simple way to build $\mathbb{I}_{ti}(z)$ for a discontinuous loss F ($\tilde{e}_{ti} < \tilde{e}_{(t-1)i}$ and z are represented), $\mathcal{O}$ being the set of solutions as it is built. We rotate two half-lines, one passing through $(\tilde{e}_{ti}, F(\tilde{e}_{ti}))$ (thick line, (Δ)) and a parallel one translated by $-z$ (dashed line) (a). As soon as (Δ) crosses F on any point $(z', F(z'))$ with $z \neq \tilde{e}_{ti}$ while the dashed line stays below F, we obtain a candidate offset v for OO, namely $v = z' - \tilde{e}_{ti}$. In (b), we obtain an interval of values. We keep on rotating (Δ), eventually making appear several intervals for the choice of v if F is not convex (c). Finally, when we reach an angle such that the maximal difference between (Δ) and F in $[\tilde{e}_{ti}, \tilde{e}_{(t-1)i}]$ is z (z can be located at an intersection between F and the dashed line), we stop and obtain the full $\mathbb{I}_{ti}(z)$ (d).

so that $\eta_t \doteq \eta(\boldsymbol{w}_t, h_t)$. Remind the weight update, $w_{ti} \doteq -\delta_{v_{(t-1)i}} F(y_i H_{t-1}(\boldsymbol{x}_i))$. We define a "partial" weight update,

$$\tilde{w}_{ti}(\alpha) \doteq -\delta_{v_{(t-1)i}} F(\alpha y_i h_t(\boldsymbol{x}_i) + y_i H_{t-1}(\boldsymbol{x}_i)) \tag{23}$$

(if we were to replace $v_{(t-1)i}$ by v_{ti} and let $\alpha \doteq \alpha_t$, then $\tilde{w}_{ti}(\alpha)$ would be $w_{(t+1)i}$, hence the partial weight update). Algorithm 2 presents the simple procedure to find α_t. Notice that we use $\tilde{\boldsymbol{w}}$ with sole dependency on the prospective leveraging coefficient; we omit for clarity the dependences in the current ensemble ($H_.$), weak classifier ($h_.$) and offsets ($v_{.i}$) needed to compute (23).

Theorem 5.8. *Suppose Assumptions 5.1 and 5.5 hold and F is continuous at all abscissae $\{\tilde{e}_{(t-1)i} \doteq y_i H_{t-1}(\boldsymbol{x}_i), i \in [m]\}$. Then there are always solutions to Step 1 of* SOLVE$_\alpha$ *and if we let $\alpha_t \leftarrow$SOLVE$_\alpha(\mathcal{S}, \boldsymbol{w}_t, h_t)$ and then compute*

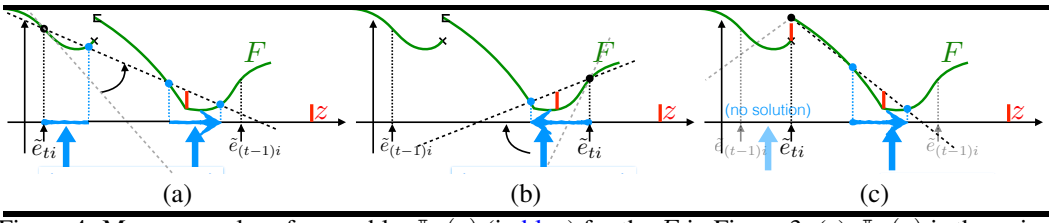
$$\overline{W}_{2,t} \doteq \left| \mathbb{E}_{i \sim [m]} \left[\frac{h_t^2(\boldsymbol{x}_i)}{M_t^2} \cdot \delta_{\{\alpha_t y_i h_t(\boldsymbol{x}_i), v_{(t-1)i}\}} F(\tilde{e}_{(t-1)i}) \right] \right|,$$

then $\overline{W}_{2,t}$ satisfies (14) *and α_t satisfies* (11) *for some $\varepsilon_t > 0, \pi_t \in (0, 1)$.*

The proof, in Section II.6, proceeds by reducing condition (15) to (22). The Weak Learning Assumption (5.5) is important for the denominator in the LHS of (22) to be non zero. The continuity assumption *at all abscissae* is important to have $\lim_{a \to 0} \eta(\tilde{\boldsymbol{w}}_t(a), h_t) = \eta_t$, which ensures the existence of solutions to (22), also easy to find, *e.g.* by a simple dichotomic search starting from an initial guess for a. Note the necessity of being continuous only at abscissae defined by the training sample, which is finite in size. Hence, if this condition is not satisfied but discontinuities of F are of Lebesgue measure 0, it is easy to add an infinitesimal constant to the current weak classifier, ensuring the conditions of Theorem 5.8 and keeping the boosting rates.

5.4 Implementation of the offset oracle

Figure 3 explains how to build graphically $\mathbb{I}_{ti}(z)$ for a general F. While it is not hard to implement a general procedure following the blueprint (*i.e.* accepting the loss function as input), it would be far from achieving computational optimality: a much better choice consists in specializing it to the (set of) loss(es) at hand via hardcoding specific optimization features of the desired loss(es). This would not prevent "loss oddities" to get absolutely trivial oracles (see Appendix, Section II.7).



(a) (b) (c)

Figure 4: More examples of ensembles $\mathbb{I}_{ti}(z)$ (in blue) for the F in Figure 3. (a): $\mathbb{I}_{ti}(z)$ is the union of two intervals with all candidate offsets non negative. (b): it is a single interval with non-positive offsets. (c): at a discontinuity, if z is smaller than the discontinuity, we have no direct solution for $\mathbb{I}_{ti}(z)$ for at least one positioning of the edges, but a simple trick bypasses the difficulty (see text).

6 Discussion

For an efficient implementation, boosting requires specific design choices to make sure the weak learning assumption stands for as long as necessary; experimentally, it is thus a good idea to adapt the weak learner to build more complex models as iterations increase (*e.g.* learning deeper trees), keeping Assumption 5.5 valid with its advantage over random guessing parameter $\gamma > 0$. In our more general setting, our algorithm SECBOOST pinpoints two more locations that can make use of specific design choices to keep assumptions stand for a larger number of iterations.

The first is related to handling local minima. When Assumption 5.4 breaks, it means we are close to a local optimum of the loss. One possible way of escaping those local minima is to adapt the offset oracle to output larger offsets (Step 2.5) that get weights computed outside the domain of the local minimum. As an illustration, in Figure 2, the middle green box shows a local minimum in which SECBOOST can get trapped if offsets are small enough (slopes of mixed signs, just like in the blue area; $\overline{W}_{1,t}$ is small, Assumption 5.4 breaks). However, for larger offsets, many slopes will tip to being positive with one of their intersection with F in the blue area, signalling a better optimum for prediction in this blue area that SECBOOST can reach with its model update, schematized with a gray arrow ($\overline{W}_{1,t}$ is large, Assumption 5.4 does not break). Section 5.4 has presented a general blueprint for the offset oracle but more specific implementation designs can be used; some are discussed in the Appendix, Section II.7.

The second is related to handling losses that take on constant values over parts of their domain. To prevent early stopping in Step 2.7 of SECBOOST, one needs $\boldsymbol{w}_{t+1} \neq \mathbf{0}$. The update rule of $\boldsymbol{w}_t$ imposes that the loss must then have non-zero *variation* for some examples between two successive edges (9). If the loss F is constant, then clearly the algorithm obviously stops without learning anything. If F is piecewise-constant, this constrain the design of the weak learner to make sure that some examples receive a different loss with the new model update $H_\cdot$. As explained in Appendix, Section II.11, this can be efficiently addressed by specific designs on SOLVE_α.

In the same way as there is no "1 size fits all" weak learner for all domains in traditional boosting, we expect specific design choices to be instrumental in better handling specific losses in our more general setting. Our theory points two locations further work can focus on.

7 Conclusion

Boosting has rapidly moved to an optimization setting involving first-order information about the loss optimized, rejoining, in terms of information needed, that of the hugely popular (stochastic) gradient descent. But this was not a formal requirement of the initial setting and in this paper, we show that essentially any loss function can be boosted without this requirement. From this standpoint, our results put boosting in a slightly more favorable light than recent development on zeroth-order optimization since, to get boosting-compliant convergence, we do not need the loss to meet any of the assumptions that those analyses usually rely on. Of course, recent advances in zeroth-order optimization have also achieved substantial design tricks for the implementation of such algorithms, something that undoubtedly needs to be adressed in our case, such as for the efficient optimization of the offset oracle. We leave this as an open problem but provide in Appendix some toy *experiments* that a straightforward implementation achieves, hinting that SECBOOST can indeed optimize very "exotic" losses.

References

[1] A. Akhavan, E. Chzhen, M. Pontil, and A.-B. Tsybakov. A gradient estimator via l1-randomization for online zero-order optimization with two point feedback. In *NeurIPS*35*, 2022.

[2] A. Akhavan, M. Pontil, and A.-B. Tsybakov. Exploiting higher order smoothness in derivative-free optimization and continuous bandits. In *NeurIPS*33*, 2020.

[3] A. Akhavan, M. Pontil, and A.-B. Tsybakov. Distributed zero-order optimisation under adversarial noise. In *NeurIPS*34*, 2021.

[4] N. Alon, A. Gonen, E. Hazan, and S. Moran. Boosting simple learners. In *STOC'21*, 2021.

[5] S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. Oxford University Press, 2000.

[6] F. Bach. *Learning Theory from First Principles*. Course notes, MIT press (to appear), 2023.

[7] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with bregman divergences. In *Proc. of the* 4^{th} *SIAM International Conference on Data Mining*, pages 234–245, 2004.

[8] P. Bartlett, M. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *J. of the Am. Stat. Assoc.*, 101:138–156, 2006.

[9] P.-L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.

[10] G. Biau, B. Cadre, and L. Rouvière. Accelerated gradient boosting. *Mach. Learn.*, 108(6):971–992, 2019.

[11] M. Blondel, A.-F. T. Martins, and V. Niculae. Learning with Fenchel-Young losses. *J. Mach. Learn. Res.*, 21:35:1–35:69, 2020.

[12] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comp. Math. and Math. Phys.*, 7:200–217, 1967.

[13] S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.

[14] P.-S. Bullen. *Handbook of means and their inequalities*. Kluwer Academic Publishers, 2003.

[15] H. Cai, Y. Lou, D. McKenzie, and W. Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *38*th *ICML*, pages 1193–1203, 2021.

[16] C. Cartis and L. Roberts. Scalable subspace methods for derivative-free nonlinear least-squares optimization. *Math. Prog.*, 199:461–524, 2023.

[17] S. Cheamanunkul, E. Ettinger, and Y. Freund. Non-convex boosting overcomes random label noise. *CoRR*, abs/1409.2905, 2014.

[18] L. Chen, J. Xu, and L. Luo. Faster gradient-free algorithms for nonsmooth nonconvex stochastic optimization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 5219–5233. PMLR, 2023.

[19] X. Chen, S. Liu, K. Xu, X. Li, X. Lin, M. Hong, and D. Cox. ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. In *NeurIPS*32*, 2019.

[20] X. Chen, Y. Tang, and N. Li. Improve single-point zeroth-order optimization using high-pass and low-pass filters. In *39*th *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 3603–3620. PMLR, 2022.

[21] S. Cheng, G. Wu, and J. Zhu. On the convergence of prior-guided zeroth-order optimisation algorithms. In *NeurIPS*34*, 2021.

[22] Z. Cranko and R. Nock. Boosted density estimation remastered. In *36*th *ICML*, pages 1416–1425, 2019.

[23] W. de Vazelhes, H. Zhang, H. Wu, X. Yuan, and B. Gu. Zeroth-order hard-thresholding: Gradient error vs. expansivity. In *NeurIPS*35*, 2022.

[24] D. Dua and C. Graff. UCI machine learning repository, 2021.

[25] E. Fermi and N. Metropolis. Numerical solutions of a minimum problem. Technical Report TR LA-1492, Los Alamos Scientific Laboratory of the University of California, 1952.

[26] L. Flokas, E.-V. Vlatakis-Gkaragkounis, and G. Piliouras. Efficiently avoiding saddle points with zero order methods: No gradients required. In *NeurIPS*32*, 2019.

[27] H. Gao and H. Huang. Can stochastic zeroth-order frank-wolfe method converge faster for non-convex problems? In *37th ICML*, pages 3377–3386, 2020.

[28] A. Héliou, M. Martin, P. Mertikopoulos, and T. Rahier. Zeroth-order non-convex learning via hierarchical dual averaging. In *38th ICML*, pages 4192–4202, 2021.

[29] F. Huang, L. Tao, and S. Chen. Accelerated stochastic gradient-free and projection-free methods. In *37th ICML*, pages 4519–4530, 2020.

[30] B. Irwin, E. Haber, R. Gal, and A. Ziv. Neural network accelerated implicit filtering: Integrating neural network surrogates with provably convergent derivative free optimization methods. In *40th ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 14376–14389. PMLR, 2023.

[31] V. Kac and P. Cheung. *Quantum calculus*. Springer, 2002.

[32] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. M.I.T. Press, 1994.

[33] M.J. Kearns. Thoughts on hypothesis boosting, 1988. ML class project.

[34] M.J. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. *J. Comp. Syst. Sc.*, 58:109–128, 1999.

[35] J. Larson, M. Menickelly, and S.-M. Wild. Derivative-free optimization methods. *Acta Numerica*, pages 287–404, 2019.

[36] Z. Li, P.-Y. Chen, S. Liu, S. Lu, and Y. Xu. Zeroth-order optimization for composite problems with functional constraints. In *AAAI'22*, pages 7453–7461. AAAI Press, 2022.

[37] T. Lin, Z. Zheng, and M.-I. Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. In *NeurIPS*35*, 2022.

[38] P.-M. Long and R.-A. Servedio. Random classification noise defeats all convex potential boosters. *MLJ*, 78(3):287–304, 2010.

[39] C. Maheshwari, C.-Y. Chiu, E. Mazumdar, S. Shankar Sastry, and L.-J. Ratliff. Zeroth-order methods for convex-concave minmax problems: applications to decision-dependent risk minimization. In *25th AISTATS*, 2022.

[40] Y. Mansour, R. Nock, and R.-C. Williamson. Random classification noise does not defeat all convex potential boosters irrespective of model choice. In *40th ICML*, 2023.

[41] E. Mhanna and M. Assaad. Single point-based distributed zeroth-order optimization with a non-convex stochastic objective function. In *40th ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 24701–24719. PMLR, 2023.

[42] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

[43] Y. Nesterov and V. Spokoiny. Random gradient-free optimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

[44] F. Nielsen and R. Nock. The Bregman chord divergence. In *Geometric Science of Information - 4th International Conference, 2019*, pages 299–308, 2019.

[45] R. Nock and A. K. Menon. Supervised learning: No loss no cry. In *37th ICML*, 2020.

[46] R. Nock and F. Nielsen. Bregman divergences and surrogates for learning. *IEEE Trans.PAMI*, 31:2048–2059, 2009.

[47] R. Nock and R.-C. Williamson. Lossless or quantized boosting with integer arithmetic. In *36th ICML*, pages 4829–4838, 2019.

[48] N.-E. Pfetsch and Sebastian Pokutta. IPBoost - non-convex boosting via integer programming. In *37th ICML*, volume 119, pages 7663–7672, 2020.

[49] Y. Qiu, U.-V. Shanbhag, and F. Yousefian. Zeroth-order methods for nondifferentiable, nonconvex and hierarchical federated optimization. In *NeurIPS*36*, 2023.

[50] M. Rando, C. Molinari, L. Rosasco, and S. Villa. Structured zeroth-order for non-smooth optimization. In *NeurIPS*36*, 2023.

[51] M.-D. Reid and R.-C. Williamson. Composite binary losses. *JMLR*, 11:2387–2422, 2010.

[52] M.-D. Reid and R.-C. Williamson. Information, divergence and risk for binary experiments. *JMLR*, 12:731–817, 2011.

[53] Z. Ren, Y. Tang, and N. Li. Escaping saddle points in zeroth-order optimization: the power of two-point estimators. In *40^{th} ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 28914–28975. PMLR, 2023.

[54] A.-K. Sahu, M. Zaheer, and S. Kar. Towards gradient free and projection free stochastic optimization. In *22^{nd} AISTATS*, pages 3468–3477, 2019.

[55] W. Shi, H. Gao, and B. Gu. Gradient-free method for heavily constrained nonconvex optimization. In *39^{th} ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 19935–19955. PMLR, 2022.

[56] M.-K. Warmuth and S. V. N. Vishwanathan. Tutorial: Survey of boosting from an optimization perspective. In *26^{th} ICML*, 2009.

[57] T. Werner and P. Ruckdeschel. The column measure and gradient-free gradient boosting, 2019.

[58] H. Zhang and B. Gu. Faster gradient-free methods for escaping saddle points. In *ICLR'23*. OpenReview.net, 2023.

[59] H. Zhang, H. Xiong, and B. Gu. Zeroth-order negative curvature finding: Escaping saddle points without gradients. In *NeurIPS*35*, 2022.

Supplementary Material

Abstract

This is the Supplementary Material to Paper "How to Boost Any Loss Function" submitted to NeurIPS'24.

This is the Appendix to paper "How to Boost Any Loss Function". To differentiate with the numberings in the main file, the numbering of Theorems, etc. is letter-based (A, B, ...).

Table of contents

reference	F conv.	diff.	Lip.	smooth	Lb	∇F diff.	main ML topic
[2]	✓	✓	✓	✓			online ML
[3]	✓		✓				distributed ML
[1]	✓		✓				online ML
[15]	✓	✓		✓		✓	alt. GD
[16]		✓		✓		✓	alt. GD
[19]		✓	✓				alt. GD
[18]			✓		✓		alt. GD
[20]	✓	✓	✓	✓			alt. GD
[21]	✓	✓		✓			alt. GD
[26]		✓	✓	✓			saddle pt opt
[27]		✓		✓			alt. FW
[29]		✓		✓			alt. FW
[23]	✓	✓					alt. GD
[28]			✓				online ML
[30]		✓		✓			deep ML
[36]	✓	✓		✓			alt. GD
[37]			✓				saddle pt opt
[39]	✓	✓	✓	✓			saddle pt opt
[41]		✓		✓		✓	distributed ML
[50]		✓		✓			alt. GD
[49]		✓	✓	✓			federated ML
[53]		✓	✓	✓		✓	saddle pt opt
[54]		✓		✓			alt. FW
[55]		✓	✓	✓			alt. GD
[58]		✓	✓	✓		✓	saddle pt opt
[59]		✓	✓	✓		✓	saddle pt opt

Table A1: Summary of formal assumptions about loss F used to prove algorithms' convergence in recent papers on zeroth order optimization, in different ML settings (see text for details). We use "smoothness" as a portmanteau for various conditions on the $\geqslant 1$ order differentiability condition of F. "conv." = convex, "diff." = differentiable, "Lip." = Lipschitz, "Lb" = lower-bounded, "alt. GD" = general alternative to gradient descent (stochastic or not), "alt. FW" = idem for Frank-Wolfe. Our paper relies on no such assumptions.

I A quick summary of recent zeroth-order optimization approaches

Table A1 summarizes a few dozens of recent approaches that can be related to zeroth-order optimization in various topics of ML. Note that no such approaches focus on boosting.

II Supplementary material on proofs

II.1 Helper results

We now show that the order of the elements of $\mathcal{V}$ does not matter to compute the $\mathcal{V}$-derivative as in Definition 4.2. For any $\boldsymbol{\sigma} \in \{0,1\}^n$, we let $1_{\boldsymbol{\sigma}} \doteq \sum_i \sigma_i$.

Lemma A. *For any $z \in \mathbb{R}$, any $n \in \mathbb{N}_*$ and any $\mathcal{V} \doteq \{v_1, v_2, ..., v_n\} \subset \mathbb{R}$,*

$$\delta_{\mathcal{V}} F(z) = \frac{\sum_{\boldsymbol{\sigma} \in \{0,1\}^n} (-1)^{n-1_\sigma} F(z + \sum_{i=1}^n \sigma_i v_i)}{\prod_{i=1}^n v_i}. \tag{24}$$

Hence, $\delta_{\mathcal{V}} F$ is invariant to permutations of the elements of $\mathcal{V}$.

Proof. We show the result by induction on the size of $\mathcal{V}$, first noting that

$$\delta_{\{v_1\}}F(z) = \delta_{v_1}F(z) \doteq \frac{F(z+v_1)-F(z)}{v_1} = \frac{1}{\prod_{i=1}^1 v_i} \cdot \sum_{\sigma\in\{0,1\}} (-1)^{1-1_\sigma} F(z+\sigma v_1). \quad (25)$$

We then assume that (24) holds for $\mathcal{V}_n \doteq \{v_1, v_2, ..., v_n\}$ and show the result for $\mathcal{V}_{n+1} \doteq \mathcal{V}_n \cup \{v_{n+1}\}$, writing (induction hypothesis used in the second identity):

$$\begin{aligned}
&\delta_{\mathcal{V}_{n+1}}F(z)\\
&\doteq \frac{\delta_{\mathcal{V}_n}F(z+v_{n+1}) - \delta_{\mathcal{V}_n}F(z)}{v_{n+1}}\\
&= \frac{\sum_{\boldsymbol{\sigma}\in\{0,1\}^n}(-1)^{n-1_{\boldsymbol{\sigma}}}F(z+\sum_{i=1}^n \sigma_i v_i + v_{n+1}) - \sum_{\boldsymbol{\sigma}\in\{0,1\}^n}(-1)^{n-1_{\boldsymbol{\sigma}}}F(z+\sum_{i=1}^n \sigma_i v_i)}{\prod_{i=1}^{n+1} v_i}\\
&= \frac{\sum_{\boldsymbol{\sigma}\in\{0,1\}^n}(-1)^{n-1_{\boldsymbol{\sigma}}}F(z+\sum_{i=1}^n \sigma_i v_i + v_{n+1}) + \sum_{\boldsymbol{\sigma}\in\{0,1\}^n}(-1)^{n-1_{\boldsymbol{\sigma}}+1}F(z+\sum_{i=1}^n \sigma_i v_i)}{\prod_{i=1}^{n+1} v_i}\\
&= \frac{\begin{cases} \sum_{\boldsymbol{\sigma'}\in\{0,1\}^{n+1}:\sigma'_{n+1}=1}(-1)^{n-(1_{\boldsymbol{\sigma'}}-1)}F(z+\sum_{i=1}^{n+1}\sigma'_i v_i) \\ +\sum_{\boldsymbol{\sigma'}\in\{0,1\}^{n+1}:\sigma'_{n+1}=0}(-1)^{n+1-1_{\boldsymbol{\sigma'}}}F(z+\sum_{i=1}^{n+1}\sigma'_i v_i)\end{cases}}{v^{n+1}}\\
&= \frac{\sum_{\boldsymbol{\sigma'}\in\{0,1\}^{n+1}}(-1)^{n+1-1_{\boldsymbol{\sigma'}}}F(z+\sum_{i=1}^{n+1}\sigma'_i v_i)}{\prod_{i=1}^{n+1} v_i}, \qquad (26)
\end{aligned}$$

as claimed. □

We also have the following simple Lemma, which is a direct consequence of Lemma A.

Lemma B. *For all $z, \in \mathbb{R}, v, z' \in \mathbb{R}_*$, we have*

$$\delta_v F(z+z') = \delta_v F(z) + z' \cdot \delta_{\{z',v\}}F(z). \quad (27)$$

Proof. It comes from Lemma A that $\delta_{\{z',v\}}F(z) = \delta_{\{v,z'\}}F(z) = (\delta_v F(z+z') - \delta_v F(z))/z'$ (and we reorder terms). □

II.2 Removing the $\neq 0$ part in Assumption 5.1

Because everything needs to be encoded, finiteness is not really an assumption. However, the non-zero assumption may be seen as limiting (unless we are happy to use first-order information about the loss (Section 5). There is a simple trick to remove it. Suppose h_t zeroes on some training examples. The training sample being finite, there exists an open neighborhood $\mathbb{I}$ in 0 such that $h'_t \doteq h_t + \delta$ does not zero anymore on training examples, for any $\delta \in \mathbb{I}$. This changes the advantage γ in the WLA (Definition 5.5) to some γ' satisfying (we assume $\delta > 0$ wlog)

$$\begin{aligned}
\gamma' &\geqslant \frac{\gamma M_t}{M_t+\delta} - \frac{\delta}{M_t+\delta}\\
&\geqslant \gamma - \frac{\delta}{M_t}\cdot(1+\gamma),
\end{aligned}$$

from which it is enough to pick $\delta \leqslant \varepsilon\gamma M_t/(1+\gamma)$ to guarantee advantage $\gamma' \geqslant (1-\varepsilon)\gamma$. If ε is a constant, this translates in a number of boosting iterations in Corollary 5.6 affected by a constant factor that we can choose as close to 1 as desired.

II.3 Proof of Lemma 5.2

We reformulate

$$\delta_{\{b,c\}}F(a) = \frac{2}{b}\cdot\frac{1}{c}\cdot\left(\underbrace{\frac{F(a+b+c)+F(a)}{2}}_{\doteq\mu_2} - \underbrace{\frac{F(a+b)+F(a+c)}{2}}_{\doteq\mu_1}\right). \quad (28)$$

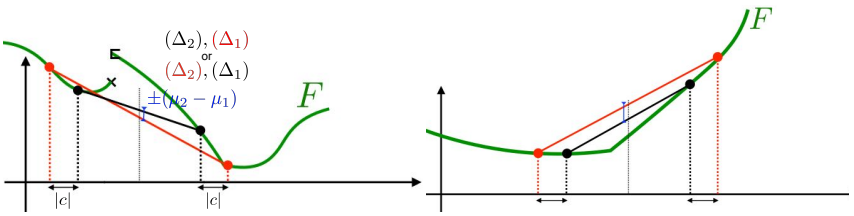


Figure II.1: *Left*: representation of the difference of averages in (28). Each of the secants (Δ_1) and (Δ_2) can take either the red or black segment. Which one is which depends on the signs of c and b, but the general configuration is always the same. Note that if F is convex, one necessarily sits above the other, which is the crux of the proof of Lemma 5.2. For the sake of illustration, suppose we can analytically have $b, c \to 0$. As c converges to 0 but b remains > 0, $\delta_{\{b,c\}} F(a)$ becomes proportional to the variation of the average secant midpoint; the then-convergence of b to 0 makes $\delta_{\{b,c\}} F(a)$ converge to the second-order derivative of F at a. *Right*: in the special case where F is convex, one of the secants always sits above the other.

Both μ_1 and μ_2 are averages that can be computed from the midpoints of two secants (respectively):

$$(\Delta_1) \doteq [(a + c, F(a + c)), (a + b, F(a + b))],$$
$$(\Delta_2) \doteq [(a, F(a)), (a + b + c, F(a + b + c))].$$

Also, the midpoints of both secants have the same abscissa (and the ordinates are μ_1 and μ_2), so to study the sign of $\delta_{\{b,c\}} F(a)$, we can study the position of both secants with respect to each other. F being convex, we show that the abscissae of one secant are included in the abscissae of the other, this being sufficient to give the position of both secants with respect to each other. We distinguish four cases.

Case 1: $c > 0, b > 0$. We have $a + b + c > \max\{a + b, a + c\}$ and $a < \min\{a + b, a + c\}$. F being convex, (Δ_2) sits above (Δ_1). So, $\mu_2 \geqslant \mu_1$ and finally $\delta_{\{b,c\}} F(a) \geqslant 0$.

Case 2: $c < 0, b < 0$. We now have $a + b + c < \min\{a + b, a + c\}$ while $a > \max\{a + b, a + c\}$, so (Δ_2) sits above (Δ_1). Again, $\mu_2 \geqslant \mu_1$ and finally $\delta_{\{b,c\}} F(a) \geqslant 0$.

Case 3: $c > 0, b < 0$. We have $a + b < a$ and $a + b < a + b + c$. Also $a + c > \max\{a + b + c, a\}$, so this time (Δ_2) sits below (Δ_1) but $cb < 0$, so $\delta_{\{b,c\}} F(a) \geqslant 0$ again.

Case 4: $c < 0, b > 0$. So $a + c < a < a + b$ and $a + c < a + b + c$. So $a + c < \min\{a, a + b + c\}$ and $a + b > \max\{a, a + c\}$, so (Δ_2) sits below (Δ_1). Since $cb < 0$, so $\delta_{\{b,c\}} F(a) \geqslant 0$ again.

II.4 Proof of Theorem 5.3

Let us remind key simplified notations about edges, $\forall t \geqslant 0$:

$$\tilde{e}_{ti} \doteq y_i \cdot H_t(\boldsymbol{x}_i), \tag{29}$$
$$e_{ti} \doteq y_i \cdot \alpha_t h_t(\boldsymbol{x}_i) = \tilde{e}_{ti} - \tilde{e}_{(t-1)i}. \tag{30}$$

For short, we also let:

$$Q^*_{ti} \doteq Q^*_F(\tilde{e}_{ti}, \tilde{e}_{(t-1)i}, v_{i(t-1)}), \tag{31}$$
$$\Delta_{ti} \doteq \delta_{v_{i(t-1)}} F(\tilde{e}_{ti}) - \delta_{v_{i(t-1)}} F(\tilde{e}_{(t-1)i}), \tag{32}$$

where $Q^*_{\cdot\cdot}$ is defined in (8). We also split the computation of the leveraging coefficient α_t in SECBOOSTin two parts, the first computing a real a_t as:

$$a_t \in \frac{1}{2(1 + \varepsilon_t) M_t^2 \overline{W}_{2,t}} \cdot [1 - \pi_t, 1 + \pi_t], \tag{33}$$

and then using $\alpha_t \leftarrow a_t\eta_t$. We now use Lemma 4.7 (main file) and get

$$\mathbb{E}_{i\sim[m]}\left[S_{F|v_{ti}}(\tilde{e}_{ti}\|\tilde{e}_{(t+1)i})\right] \geqslant -\mathbb{E}_{i\sim D}\left[Q^*_{(t+1)i}\right], \forall t \geqslant 0. \tag{34}$$

If we reorganise (34) using the definition of $S_{F|.}(.\|.)$, we get:

$$\begin{aligned}
&\mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{(t+1)i})\right] \\
&\leqslant \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - \mathbb{E}_{i\sim[m]}\left[(\tilde{e}_{ti} - \tilde{e}_{(t+1)i})\cdot \delta_{v_{ti}}F(\tilde{e}_{(t+1)i})\right] + \mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right] \\
&= \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - \mathbb{E}_{i\sim[m]}\left[-e_{(t+1)i}\cdot \delta_{v_{ti}}F(\tilde{e}_{(t+1)i})\right] + \mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right] &(35)\\
&= \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] + \alpha_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[y_i h_{t+1}(\boldsymbol{x}_i)\cdot \delta_{v_{ti}}F(\tilde{e}_{(t+1)i})\right] + \mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right] &(36)\\
&= \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] + a_{t+1}\eta_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[y_i h_{t+1}(\boldsymbol{x}_i)\cdot \delta_{v_{ti}}F(\tilde{e}_{ti})\right] \\
&\quad + a_{t+1}\eta_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[y_i h_{t+1}(\boldsymbol{x}_i)\cdot \Delta_{(t+1)i}\right] + \mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right] &(37)\\
&= \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - a_{t+1}\eta_{t+1}\cdot \underbrace{\mathbb{E}_{i\sim[m]}\left[w_{(t+1)i}\underbrace{y_i h_{t+1}(\boldsymbol{x}_i)}\right]}_{=\eta_{t+1}} + a_{t+1}\eta_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[y_i h_{t+1}(\boldsymbol{x}_i)\cdot \Delta_{(t+1)i}\right] \\
&\quad + \mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right] \\
&= \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - a_{t+1}\eta^2_{t+1} + a_{t+1}\eta_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[y_i h_{t+1}(\boldsymbol{x}_i)\cdot \Delta_{(t+1)i}\right] + \mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right]. &(38)
\end{aligned}$$

(35) – (37) make use of definitions (30) (twice) and (32) as well as the decomposition of the leveraging coefficient in (33).

Looking at (38), we see that we can have a boosting-compliant decrease of the loss if the two quantities depending on $\Delta_{(t+1).}$ and $Q^*_{(t+1).}$ can be made small enough compared to $a_{t+1}\eta^2_{t+1}$. This is what we investigate.

Bounding the term depending on $\Delta_{(t+1).}$ – We use Lemma B with $z \doteq \tilde{e}_{ti}, z' \doteq e_{(t+1)i}, v \doteq v_t$, which yields (also using (30) and the assumption that $h_{t+1}(\boldsymbol{x}_i) \neq 0$):

$$\begin{aligned}
\Delta_{(t+1)i} &\doteq \delta_{v_{ti}}F(\tilde{e}_{(t+1)i}) - \delta_{v_{ti}}F(\tilde{e}_{ti}) \\
&= \delta_{v_{ti}}F(\tilde{e}_{ti} + e_{(t+1)i}) - \delta_{v_{ti}}F(\tilde{e}_{ti}) \\
&= e_{(t+1)i}\cdot \delta_{\{e_{(t+1)i}, v_{ti}\}}F(\tilde{e}_{ti}) \\
&= y_i\cdot \alpha_{t+1}h_{t+1}(\boldsymbol{x}_i)\cdot \delta_{\{e_{(t+1)i}, v_{ti}\}}F(\tilde{e}_{ti}),
\end{aligned} \tag{39}$$

and so we get:

$$\begin{aligned}
&a_{t+1}\eta_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[y_i h_{t+1}(\boldsymbol{x}_i)\cdot \Delta_{(t+1)i}\right] \\
&= a_{t+1}\eta_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[\alpha_{t+1}(y_i h_{t+1}(\boldsymbol{x}_i))^2\cdot \delta_{\{e_{(t+1)i}, v_{ti}\}}F(\tilde{e}_{ti})\right] \\
&= a^2_{t+1}\eta^2_{t+1}\cdot \mathbb{E}_{i\sim[m]}\left[(h_{t+1}(\boldsymbol{x}_i))^2\cdot \delta_{\{e_{(t+1)i}, v_{ti}\}}F(\tilde{e}_{ti})\right] \\
&\leqslant a^2_{t+1}\eta^2_{t+1}M^2_{t+1}\cdot \overline{W}_{2,t+1}.
\end{aligned} \tag{40}$$

Bounding the term depending on $Q^*_{.(t+1)}$ – We immediately get from the value picked in argument of $\mathbb{I}_{t+1}$ in step 2.5 of SECBOOST, the definition of $\mathbb{I}_{ti}(.)$ in (16) and our decomposition $\alpha_t \leftarrow a_t\eta_t$ that $Q^*_{(t+1)i} \leqslant \varepsilon_{t+1}\cdot a^2_{t+1}\eta^2_{t+1}M^2_{t+1}\cdot \overline{W}_{2,t+1}, \forall i \in [m]$, so that:

$$\mathbb{E}_{i\sim[m]}\left[Q^*_{(t+1)i}\right] \leqslant \varepsilon_{t+1}\cdot a^2_{t+1}\eta^2_{t+1}M^2_{t+1}\cdot \overline{W}_{2,t+1}. \tag{41}$$

Finishing up with the proof – Suppose that we choose $\varepsilon_{t+1} > 0$, $\pi_{t+1} \in (0,1)$ and a_{t+1} as in (33). We then get from (38), (40), (41) that for any choice of v_{ti} in Step 2.5 of SECBOOST,

$$\begin{aligned}
&\mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{(t+1)i})\right] \\
&\leqslant \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - a_{t+1}\eta_{t+1}^2 + a_{t+1}^2\eta_{t+1}^2 M_{t+1}^2 \cdot \overline{W}_{2,t+1} + \varepsilon_{t+1} \cdot a_{t+1}^2\eta_{t+1}^2 M_{t+1}^2 \cdot \overline{W}_{2,t+1} \\
&= \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - a_{t+1}\eta_{t+1}^2 \cdot \left(1 - a_{t+1}\left(1+\varepsilon_{t+1}\right) M_{t+1}^2 \cdot \overline{W}_{2,t+1}\right) \\
&\leqslant \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{ti})\right] - \frac{\eta_{t+1}^2(1-\pi_{t+1}^2)}{4\left(1+\varepsilon_{t+1}\right) M_{t+1}^2 \cdot \overline{W}_{2,t+1}}, \qquad (42)
\end{aligned}$$

where the last inequality is a consequence of (33). Suppose we pick $H_0 \doteq h_0 \in \mathbb{R}$ a constant and $v_0 > 0$ such that

$$\delta_{v_0} F(h_0) \neq 0. \qquad (43)$$

The final classifier H_T of SECBOOST satisfies:

$$\mathbb{E}_{i\sim[m]}\left[F(y_i H_T(\boldsymbol{x}_i))\right] \leqslant F_0 - \frac{1}{4}\cdot\sum_{t=1}^{T}\frac{\eta_t^2(1-\pi_t^2)}{(1+\varepsilon_t)M_t^2\overline{W}_{2,t}}, \qquad (44)$$

with $F_0 \doteq \mathbb{E}_{i\sim[m]}\left[F(\tilde{e}_{i0})\right] \doteq \mathbb{E}_{i\sim[m]}\left[F(y_i H_0)\right] = \mathbb{E}_{i\sim[m]}\left[F(y_i h_0)\right]$. If we want $\mathbb{E}_{i\sim[m]}\left[F(y_i H_T(\boldsymbol{x}_i))\right] \leqslant F(z^*)$, assuming wlog $F(z^*) \leqslant F_0$, then it suffices to iterate until:

$$\sum_{t=1}^{T}\frac{1-\pi_t^2}{\overline{W}_{2,t}(1+\varepsilon_t)}\cdot\frac{\eta_t^2}{M_t^2} \geqslant 4(F_0 - F(z^*)). \qquad (45)$$

Remind that the edge η_t is not normalised. We have defined a normalised edge,

$$[-1,1] \ni \tilde{\eta}_t \doteq \sum_i \frac{|w_{ti}|}{W_t}\cdot\tilde{y}_{ti}\cdot\frac{h_t(\boldsymbol{x}_i)}{M_t}, \qquad (46)$$

with $\tilde{y}_{ti} \doteq y_i \cdot \mathrm{sign}(w_{ti})$ and $W_t \doteq \sum_i |w_{ti}| = \sum_i |\delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})|$. We have the simple relationship between η_t and $\tilde{\eta}_t$:

$$\begin{aligned}
\tilde{\eta}_t &= \sum_i \frac{|w_{ti}|}{W_t}\cdot(y_i\cdot\mathrm{sign}(w_{ti}))\cdot\frac{h_t(\boldsymbol{x}_i)}{M_t} \\
&= \frac{1}{W_t M_t}\cdot\sum_i w_{ti} y_i h_t(\boldsymbol{x}_i) \\
&= \frac{m}{W_t M_t}\cdot\eta_t, \qquad (47)
\end{aligned}$$

resulting in ($\forall t \geqslant 1$),

$$\begin{aligned}
\frac{\eta_t^2}{M_t^2} &= \tilde{\eta}_t^2\cdot\left(\frac{W_t}{m}\right)^2 \\
&= \tilde{\eta}_t^2\cdot\left(\mathbb{E}_{i\sim[m]}\left[|\delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})|\right]\right)^2 \\
&\geqslant \tilde{\eta}_t^2\cdot\left(\left|\mathbb{E}_{i\sim[m]}\left[\delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})\right]\right|\right)^2 \\
&= \tilde{\eta}_t^2\cdot\overline{W}_{1,t}^2, \qquad (48)
\end{aligned}$$

recalling $\overline{W}_{1,t} \doteq \left|\mathbb{E}_{i\sim D}\left[\delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})\right]\right|$. It comes from (48) that a sufficient condition for (45) to hold is:

$$\sum_{t=1}^{T}\frac{\overline{W}_{1,t}^2(1-\pi_t^2)}{\overline{W}_{2,t}(1+\varepsilon_t)}\cdot\tilde{\eta}_t^2 \geqslant 4(F_0 - F(z^*)), \qquad (49)$$

which is the statement of Theorem 5.3.

II.5 Proof of Lemma 5.7

We first observe that for any $a \in \mathbb{R}, b, c \in \mathbb{R}_*$,

$$\begin{aligned}
|\delta_{\{b,c\}}F(a)| &= \frac{1}{|bc|} \cdot \left| \begin{array}{c} F(a+b+c) - F(a+c) - bF'(a+c) \\ -(F(a+b) - F(a) - bF'(a)) \\ +b(F'(a+c) - F'(a)) \end{array} \right| \\
&\leqslant \frac{1}{|bc|} \cdot \left(\begin{array}{c} |F(a+b+c) - F(a+c) - bF'(a+c)| \\ +|(F(a+b) - F(a) - bF'(a))| \\ +|b(F'(a+c) - F'(a))| \end{array} \right) \\
&\leqslant \frac{1}{|bc|} \cdot \left(\frac{\beta}{2} \cdot b^2 + \frac{\beta}{2} \cdot b^2 + \beta|bc| \right) = \beta + \beta \cdot \frac{b^2}{|bc|}, \qquad (50)
\end{aligned}$$

where we used the β-smoothness of F and twice [13, Lemma 3.4]. We can also make a permutation in the expression of $\delta_{\{b,c\}}F(a)$ and instead write

$$\begin{aligned}
|\delta_{\{b,c\}}F(a)| &= \frac{1}{|bc|} \cdot \left| \begin{array}{c} F(a+b+c) - F(a+b) - cF'(a+b) \\ -(F(a+c) - F(a) - cF'(a)) \\ +c(F'(a+b) - F'(a)) \end{array} \right| \\
&\leqslant \frac{1}{|bc|} \cdot \left(\begin{array}{c} |F(a+b+c) - F(a+b) - cF'(a+b)| \\ +|(F(a+c) - F(a) - cF'(a))| \\ +|c(F'(a+b) - F'(a))| \end{array} \right) \\
&\leqslant \frac{1}{|bc|} \cdot \left(\frac{\beta}{2} \cdot c^2 + \frac{\beta}{2} \cdot c^2 + \beta|bc| \right) = \beta + \beta \cdot \frac{c^2}{|bc|}. \qquad (51)
\end{aligned}$$

We thus have

$$\begin{aligned}
|\delta_{\{b,c\}}F(a)| &\leqslant \beta + \beta \cdot \left(\frac{\min\{|b|, |c|\}}{\sqrt{|bc|}} \right)^2 \qquad (52) \\
&\leqslant 2\beta,
\end{aligned}$$

by the power mean inequality [14, Chapter III, Theorem 2]. Since $|h_t(\boldsymbol{x}_i)| \leqslant M_t$ by definition, we thus have

$$\left| \mathbb{E}_{i\sim[m]} \left[\delta_{\{e_{ti}, v_{(t-1)i}\}} F(\tilde{e}_{(t-1)i}) \cdot \left(\frac{h_t(\boldsymbol{x}_i)}{M_t} \right)^2 \right] \right| \leqslant 2\beta, \qquad (53)$$

which allows us to fix $\overline{W}_{2,t} = 2\beta$ and completes the proof of Lemma 5.7.

Remark C. *Our result is optimal in the sense that if we make one offset (say b) go to zero, then the ratio in* (52) *goes to zero and we recover the condition on the v-derivative of the derivative, $|\delta_c F'(z)| \leqslant \beta$.*

II.6 Proof of Theorem 5.8

We consider the upperbound::

$$\begin{aligned}
&\overline{W}_{2,t} \\
&\doteq \left| \mathbb{E}_{i\sim[m]} \left[\frac{h_t^2(\boldsymbol{x}_i)}{M_t^2} \cdot \delta_{\{e_{ti}, v_{(t-1)i}\}} F(\tilde{e}_{(t-1)i}) \right] \right| \\
&= \left| \mathbb{E}_{i\sim[m]} \left[\frac{h_t^2(\boldsymbol{x}_i)}{M_t^2} \cdot \frac{1}{\tilde{e}_{ti}} \cdot \left(\frac{F(\tilde{e}_{ti} + v_{(t-1)i}) - F(\tilde{e}_{ti})}{v_{(t-1)i}} - \frac{F(\tilde{e}_{(t-1)i} + v_{(t-1)i}) - F(\tilde{e}_{(t-1)i})}{v_{(t-1)i}} \right) \right] \right| \\
&= \left| \frac{1}{\alpha_t} \cdot \mathbb{E}_{i\sim[m]} \left[\frac{h_t(\boldsymbol{x}_i)}{y_i M_t^2} \cdot \left(\frac{F(\tilde{e}_{ti} + v_{(t-1)i}) - F(\tilde{e}_{ti})}{v_{(t-1)i}} - \frac{F(\tilde{e}_{(t-1)i} + v_{(t-1)i}) - F(\tilde{e}_{(t-1)i})}{v_{(t-1)i}} \right) \right] \right| \\
&= \left| \frac{1}{\alpha_t} \cdot \mathbb{E}_{i\sim[m]} \left[\frac{y_i h_t(\boldsymbol{x}_i)}{M_t^2} \cdot \left(\delta_{v_{(t-1)i}} F(\tilde{e}_{ti}) - \delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i}) \right) \right] \right| \qquad (54)
\end{aligned}$$

(The last identity uses the fact that $y_i \in \{-1, 1\}$). Remark that we have extracted α_t from the denominator but it is still present in the arguments $\tilde{e}_{ti}$. For any classifier h, we introduce notation

$$\eta(\boldsymbol{w}, h) \doteq \mathbb{E}_{i\sim[m]}\left[w_i y_i h(\boldsymbol{x}_i)\right],$$

and so η_t (Step 2.2 in SECBOOST) is also $\eta(\boldsymbol{w}_t, h_t)$, which is guaranteed to be non-zero by the Weak Learning Assumption (5.5). We want, for *some* $\varepsilon_t > 0, \pi_t \in [0, 1)$,

$$\alpha_t \in \frac{\eta_t}{2(1+\varepsilon_t)M_t^2 \overline{W}_{2,t}} \cdot [1-\pi_t, 1+\pi_t]. \tag{55}$$

This says that the sign of α_t is the same as the sign of $\eta(\boldsymbol{w}_t, h_t) = \eta_t$. Since we know its sign, let us look for its absolute value:

$$|\alpha_t| \in \frac{|\eta_t|}{2(1+\varepsilon_t)M_t^2 \overline{W}_{2,t}} \cdot [1-\pi_t, 1+\pi_t]. \tag{56}$$

From (15) (main file), we can in fact search α_t in the union of all such intervals for $\varepsilon_t > 0, \pi_t \in [0, 1)$, which amounts to find first:

$$|\alpha_t| \in \left(0, \frac{|\eta_t|}{M_t^2 \overline{W}_{2,t}}\right),$$

and then find any $\varepsilon_t > 0, \pi_t \in [0, 1)$ such that (56) holds. Using (54) and simplifying the external dependency on α_t, we then need

$$1 \in \left(0, \underbrace{\frac{|\eta_t|}{\left|\mathbb{E}_{i\sim[m]}\left[y_i h_t(\boldsymbol{x}_i) \cdot \left(\delta_{v_{(t-1)i}} F(\alpha_t y_i h_t(\boldsymbol{x}_i) + \tilde{e}_{(t-1)i}) - \delta_{v_{(t-1)i}} F(\tilde{e}_{(t-1)i})\right)\right]\right|}}_{\doteq B(\alpha_t)}\right), \tag{57}$$

under the constraint that the sign of α_t be the same as that of η_t. But, using notation (23) (main file), we have

$$B(\alpha_t) = |\eta(\boldsymbol{w}_t, h_t) - \eta(\tilde{\boldsymbol{w}}_t(\alpha_t), h_t)|,$$

and so to get (57) satisfied, it is sufficient that

$$\frac{|\eta_t - \eta(\tilde{\boldsymbol{w}}_t(\alpha_t), h_t)|}{|\eta_t|} < 1, \tag{58}$$

which is Step 1 in SOLVE$_\alpha$. The Weak Learning Assumption (5.5) guarantees that the denominator is $\neq 0$ so this can always be evaluated. The continuity of F in all $\tilde{e}_{(t-1)i}$ guarantees $\lim_{\alpha_t \to 0} \eta(\tilde{\boldsymbol{w}}_t(\alpha_t), h_t) = \eta_t$, and thus guarantees the existence of solutions to (58) for some $|\alpha_t| > 0$.

To summarize, finding α_t can be done in two steps, (i) solve

$$\frac{|\eta_t - \eta(\tilde{\boldsymbol{w}}_t(\mathrm{sign}(\eta_t) \cdot a), h_t)|}{|\eta_t|} < 1$$

for some $a > 0$ and (ii) let $\alpha_t \doteq \mathrm{sign}(\eta_t) \cdot a$. This is the output of SOLVE$_\alpha(\mathcal{S}, \boldsymbol{w}_t, h_t)$, which ends the proof of Theorem 5.8.

II.7 Implementation of the offset oracle: particular cases

Consider the "spring loss" that we define, for $[.]$ denoting the nearest integer, as:

$$F_{\text{SL}}(z) \doteq \log(1 + \exp(-z)) + 1 - \sqrt{1 - 4\left(z - [z]\right)^2}. \tag{59}$$

Figure II.2 plots this loss, which composes the logistic loss with a "U"-shaped term. This loss would escape all optimization algorithms of Table A1 (Appendix), yet there is a trivial implementation of our offset oracle, as explained in Figure II.2:

1. if the interval $\mathbb{I}$ defined by $\tilde{e}_{(t-1)i}$ and $\tilde{e}_{ti}$ contains at least one peak, compute the tangence point (z_t) at the closest local "U" that passes through $(\tilde{e}_{(t-1)i}, F(\tilde{e}_{(t-1)i}))$; then if $z_t \in \mathbb{I}$ then $v_{ti} \leftarrow z_t - \tilde{e}_{(t-1)i}$, else $v_{ti} \leftarrow \tilde{e}_{ti} - \tilde{e}_{(t-1)i}$;

2. otherwise F in $\mathbb{I}$ is strictly convex and differentiable: a simple dichotomic search can retrieve a feasible v_{ti} (see convex losses below);

Notice that one can alleviate the repetitive dichotomic search by pre-tabulating a feasible v for a set of differences $|a - b|$ (a, b belonging to the abscissae of the same "U") decreasing by a fixed factor, choosing $v_{ti} \leftarrow v$ of the largest tabulated $|a - b|$ no larger than $|\tilde{e}_{ti} - \tilde{e}_{(t-1)i}|$.

Discontinuities discontinuities do not represent issues if the argument z of $\mathbb{I}_{ti}(z)$ is large enough, as shown from the following simple Lemma.

Lemma D. *Define the discontinuity of F as:*

$$\mathrm{disc}(F) \doteq \max \left\{ \begin{array}{l} \sup_z |F(z) - \lim_{z^-} F(z)|, \\ \sup_z |F(z) - \lim_{z^+} F(z)| \end{array} \right\}. \tag{60}$$

For any $z \geqslant 0$, if $\mathrm{disc}(F) \leqslant z$ then $\mathbb{I}_{ti}(z) \neq \varnothing, \forall t \geqslant 1, \forall i \in [m]$.

Figure 4 (c) shows a case where the discontinuity is larger than z. In this case, an issue eventually happens for computing the next weight happens, only when the current edge is at the discontinuity. We note that as iterations increase and the weak learner finds it eventually more difficult to return weak hypotheses with $\eta_\cdot$ large enough, the discontinuities may become an issue for SECBOOST to not stop at Step 2.5. Or one can always use a simple trick to avoid stopping and which relies on the leveraging coefficient α_t: this is described in the Appendix, Section II.9.

The case of convex losses If F is convex (not necessarily differentiable nor strictly convex), there is a simple way to find a valid output for the offset oracle, which relies on the following Lemma.

Lemma E. *Suppose F convex. Then for any $z, z' \in \mathbb{R}, v \neq 0$,*

$$\begin{array}{l} \{v > 0 : Q_F^*(z, z', v) = r\} \\ = \left\{ v > 0 : D_F \left(z \left\| \dfrac{F(z+v) - F(z)}{v} \right. \right) = r \right\}. \end{array} \tag{61}$$

(proof in Appendix, Section II.8) By definition, $\mathbb{I}_{ti}(z') \subseteq \mathbb{I}_{ti}(z)$ for any $z' \leqslant z$, so a simple way to implement the offset oracle's output $\textsc{oo}(t, i, z)$ is, for some $0 < r < z$, to solve the Bregman identity in the RHS of (61) and then return any relevant v. If F is strictly convex, there is just one choice.

If solving the Bregman identity is tedious but F is strictly convex, there a simple dichotomic search that is guaranteed to find a feasible v. It exploits the fact that the abscissa maximizing the difference between any secant of F and F has a simple closed form (see [22, Supplement, Figure 13]) and so the OBI in (4) (Definition 4.6) has a closed form as well. In this case, it is enough, after taking a first non-zero guess for v (either positive or negative), to divide it by a constant > 1 until the corresponding OBI is no larger than the z in the query $\textsc{oo}(t, i, z)$.

II.8 Proof of Lemma E

F being convex, we first want to compute the set

$$\mathbb{I}_{z,v,r} \doteq \{v > 0 : Q_F(z, z+v, z+v) = r\}, \tag{62}$$

where r is supposed small enough for $\mathbb{I}_{z,v,r}$ to be non-empty. There is a simple graphical solution to this which, as Figure II.3 explains, consists in finding v solution of

$$\sup_t F(z+v) - \left(F(t) + \left(\frac{F(z+v) - F(z)}{v} \right) \cdot (z + v - t) \right) = r. \tag{63}$$

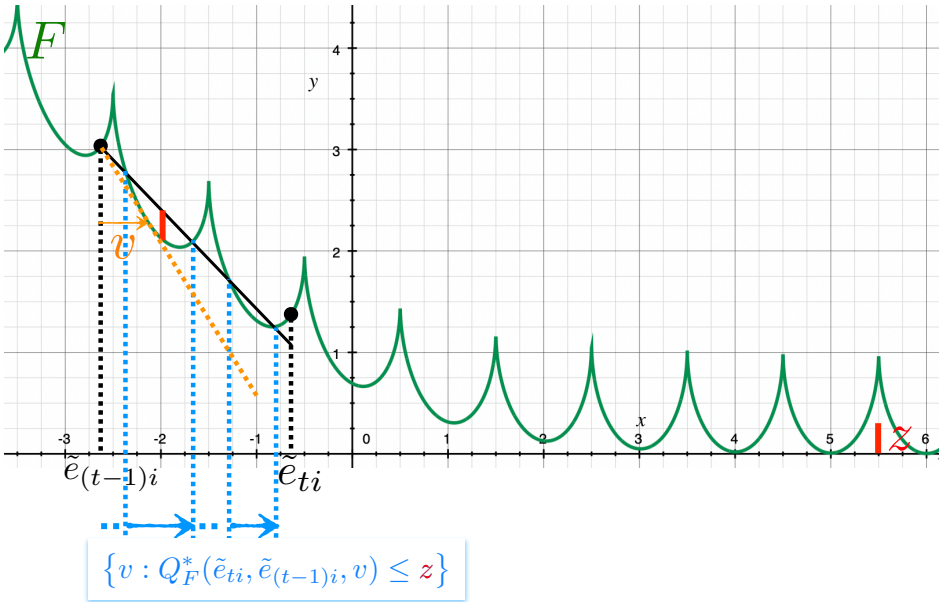


Figure II.2: The spring loss in (59) is neither convex, nor Lipschitz or differentiable and has an infinite number of local minima. Yet, an implementation of the offset oracle is trivial as an output for OO can be obtained from the computation of a single tangent point (here, the orange v, see text; best viewed in color).

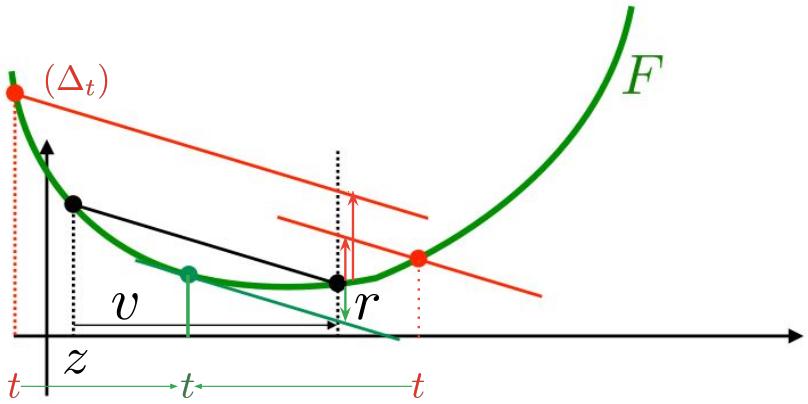


Figure II.3: Computing the OBI $Q_F(z, z+v, z+v)$ for F convex, (z, v) being given and $v > 0$. We compute the line (Δ_t) crossing F at any point t, with slope equal to the secant $[(z, F(z)), (z+v, F(z+v))]$ and then the difference between F at $z+v$ and this line at $z+v$. We move t so as to maximize this difference. The optimal t (in green) gives the corresponding OBI. In (64), we are interested in finding v given this difference, r. We also need to replicate this computation for $v < 0$.

The LHS simplifies:

$$
\begin{aligned}
&\sup_t F(z+v) - \left(F(t) + \left(\frac{F(z+v)-F(z)}{v}\right) \cdot (z+v-t)\right) \\
&= \frac{(z+v)F(z) - zF(z+v)}{v} + \sup_t \left\{ t \cdot \frac{F(z+v)-F(z)}{v} - F(t) \right\} \\
&= \frac{(z+v)F(z) - zF(z+v)}{v} + F^\star\left(\frac{F(z+v)-F(z)}{v}\right) \\
&= F(z) + F^\star\left(\frac{F(z+v)-F(z)}{v}\right) - z \cdot \frac{F(z+v)-F(z)}{v} \\
&= D_F\left(z \middle\| \frac{F(z+v)-F(z)}{v}\right), 22
\end{aligned}
$$

so we end up with an equivalent but more readable definition for $\mathbb{I}_{z,v,r}$:

$$\mathbb{I}_{z,v,r} = \left\{ v > 0 : D_F \left(z \middle\| \frac{F(z+v) - F(z)}{v} \right) = r \right\}, \tag{64}$$

which yields the statement of the Lemma.

II.9 Handling discontinuities in the offset oracle to prevent stopping in Step 2.5

Theorem 5.3 and Lemma 5.6 require to run SECBOOST for as many iterations are required. This implies not early stopping in Step 2.5. Lemma D shows that early stopping can only be triggered by too large local discontinuities at the edges. This is a weak requirement on running SECBOOST, but there exists a weak assumption on the discontinuities of the loss itself that simply prevent any early stopping and does not degrade the boosting rates. The result exploits the freedom in choosing α_t in Step 2.3.

Lemma F. *Suppose F is any function defined over $\mathbb{R}$ discontinuities of zero Lebesgue measure. Then Corollary 5.6 holds for boosting F with its inequality strict while never triggering early stopping in Step 2.5 of* SECBOOST.

Proof. To show that we never trigger stopping in Step 2.5, it is sufficient to show that we can run SECBOOSTwhile ensuring F is continuous in an open neighborhood around all edges $y_i H_t(\boldsymbol{x}_i), \forall i \in [m], \forall t \geqslant 0$ (by letting $H_0 \doteq h_0$). Remind that $\tilde{e}_{ti} \doteq \tilde{e}_{(t-1)i} + \alpha_t \cdot y_t h_t(\boldsymbol{x}_i)$, so changing α_t changes all edges. We just have to show that either computing α_t ensures such a continuity, or α_t can be slightly modified to do so. We have two ways to compute α_t:

1. using a value for $\overline{W}_{2,t}$ that represents an "absolute" upperbound in the sense of (14) (*e.g.* Lemma 5.7) and then compute α_t as in Step 2.3 of SECBOOST;
2. using algorithm SOLVE$_\alpha$.

Because of the assumption on F, we can always ensure that F is continuous in an open neighborhood of all edges (the basis of the induction amounts to a straightforward choice for h_0). This proves the Lemma for [2.].

If we rely on [1.] and the α_t computed leads to some discontinuities, then we have complete control to change α_t: any continuous change of ε_t induces a continuous change in α_t and thus a continuous change of all edges as well. So, starting from the initial ε_t chosen in Step 2.3, we increase it to a value $\varepsilon_t^* > \varepsilon_t$, which we want to keep as small as possible. We can define for each $i \in [m]$ an open set (a_i, b_i) which is the interval spanned by the new $\tilde{e}_{ti}(\varepsilon_t')$ using $\varepsilon_t' \in (\varepsilon_t, \varepsilon_t^*)$. Since there are only finitely many discontinuities on F, there exists a small $\varepsilon_t^* > \varepsilon_t$ such that

$$\forall i \in [m], \forall z \in (a_i, b_i), F \text{ is continuous on } z.$$

This means that $\forall \varepsilon_t' \in (\varepsilon_t, \varepsilon_t^*)$, we end up with a loss without any discontinuities on the new edges. Now comes the reason why we want $\varepsilon_t^* - \varepsilon_t$ small: we can check that there always exist a small enough $\varepsilon_t^* > \varepsilon_t$ such that for any ε_t' we choose, the boosting rate in Corollary 5.6 is affected by at most 1 additional iteration. Indeed, while we slightly change parameter ε_t to land all new edges outside of discontinuities of F, we *also* increase the contribution of the boosting iteration in the RHS of (21) by a quantity $\delta > 0$ which can be made as small as required — hence we can just replace the inequality in (21) by a strict inequality. This proves the statement of the Lemma if we rely on [1.] above.

This completes the proof of Lemma F. □

II.10 A boosting pattern that can "survive" above differentiability

Suppose F is strictly convex and strictly decreasing as for classical convex surrogates (*e.g.* logistic loss). Assuming wlog all $\alpha_\cdot > 0$ and example i has both $y_i h_t(\boldsymbol{x}_i) > 0$ and $y_i h_{t-1}(\boldsymbol{x}_i) > 0$, as long as z is small enough, we are guaranteed that any choice $v_{t-1} \in \mathbb{I}_{(t-1)i}(z)$ and $v_t \in \mathbb{I}_{ti}(z)$ results in $0 < w_{(t+1)i} < w_{ti}$, which follows the classical boosting pattern that examples receiving the right class by weak hypotheses have their weight decreased (See Figure II.4). If $z = z'$ is large enough, then this does not hold anymore as seen from Figure II.4.

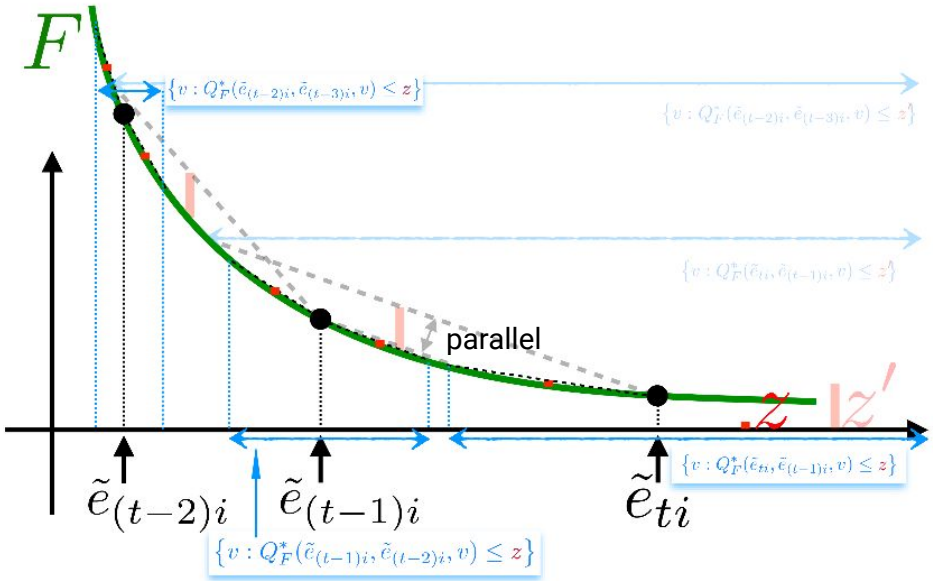


Figure II.4: Case F strictly convex, with two cases of limit OBI z and z' in $\mathbb{I}_{.i}(.)$. Example i has $e_{ti} > 0$ and $e_{(t-1)i} > 0$ (**??**) large enough (hence, edges with respect to weak classifiers h_t and h_{t-1} large enough) so that $\mathbb{I}_{ti}(z) \cap \mathbb{I}_{(t-1)i}(z) = \mathbb{I}_{(t-1)i}(z) \cap \mathbb{I}_{(t-2)i}(z) = \mathbb{I}_{ti}(z) \cap \mathbb{I}_{(t-2)i}(z) = \varnothing$. In this case, regardless of the offsets chosen by OO, we are guaranteed that its weights satisfy $w_{(t+1)i} < w_{ti} < w_{(t-1)i}$, which follows the boosting pattern that examples receiving the right classification by weak classifiers have their weights decreasing. If however the limit OBI changes from z to a larger z', this is not guaranteed anymore: in this case, it may be the case that $w_{(t+1)i} > w_{ti}$.

II.11 The case of piecewise constant losses for SOLVE$_\alpha$

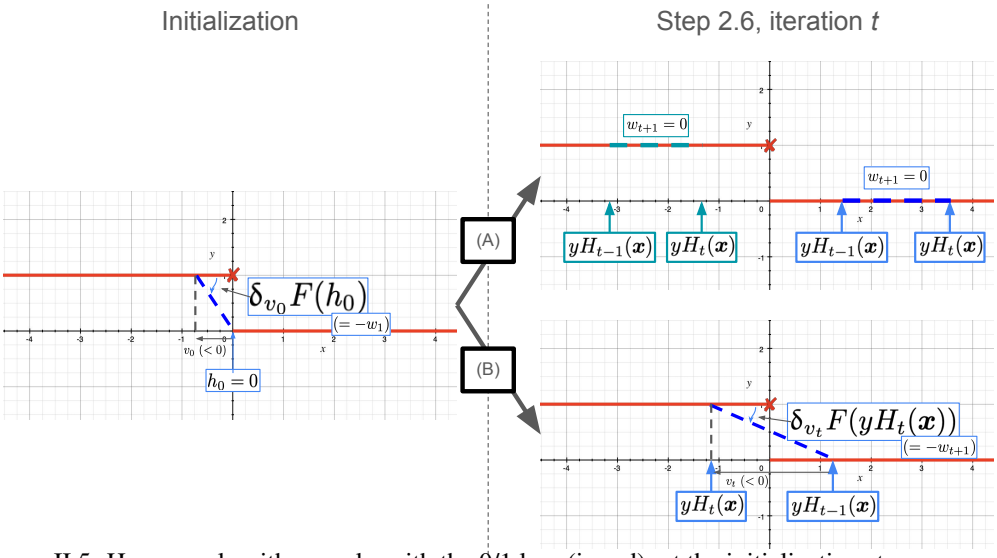


Figure II.5: How our algorithm works with the 0/1 loss (in red): at the initialization stage, assuming we pick $h_0 = 0$ for simplicity and some $v_0 < 0$, all training examples get the same weight, given by negative the slope of the thick blue dashed line. All weights are thus > 0. At iteration t when we update the weights (Step 2.6), one of two cases can happen on some training example $(\boldsymbol{x}, y)$. In **(A)**, the edge of the strong model remains the same: either both are positive (blue) or both negative (olive green) (the ordering of edges is not important). In this case, regardless of the offset, the new weight will be 0. In **(B)**, both edges have different sign (again, the ordering of edges is not important). In this case, the examples will keep non-zero weight over the next iteration. See text below for details.

Figure II.5 schematizes a run of our algorithm when training loss = 0/1 loss. At the initialization, it is easy to get all examples to have non-zero weight. The weight update for example $(\boldsymbol{x}, y)$ of our algorithm in Step 2.3 is (negative) the slope of a secant that crosses the loss in two points, both being in between $yH_{t-1}(\boldsymbol{x})$ and $yH_t(\boldsymbol{x})$. Hence, if the predicted label does not change ($\text{sign}(H_t(\boldsymbol{x})) = \text{sign}(H_{t-1}(\boldsymbol{x}))$), then the next weight (w_{t+1}) of the example *will be zero* (Figure II.5, case (A)). However, if the predicted label does change ($\text{sign}(H_t(\boldsymbol{x})) \neq \text{sign}(H_{t-1}(\boldsymbol{x}))$) then the example may get a non-zero weight depending on the offset chosen.

Hence, our generic implementation of Algorithms 3 and 4 may completely fail at providing non-zero weights for the next iteration, which makes the algorithm stop in step 2.7. And even when not all weights are zero, there may be just a too small subset of those, that would break the Weak Learning Assumption for boosting compliance of the next iteration (Assumption 5.5).

III Supplementary material on algorithms, implementation tricks and a toy experiment

III.1 Algorithm and implementation of SOLVE$_\alpha$ and how to find parameters from Theorem 5.8

As Theorem 5.8 explains, SOLVE$_\alpha$ can easily get to not just the leveraging coefficient α_t, but also other parameters that are necessary to implement SECBOOST: $\overline{W}_{2,t}$ and ε_t (both used in Step 2.5). We now provide a simple pseudo code on how to implement SOLVE$_\alpha$ amnd get, on top of it, the two other parameters. We do not seek π_t since it is useful only in the convergence analysis. Also, our proposal implementation is optimized for complexity (because of the geometric updating of δ, W in their respective loops) but much less so for for accuracy. Algorithm SOLVE_extended explains the overall procedure.

Algorithm 3 SOLVE_extended($\mathcal{S}, \boldsymbol{w}, h, M$)

Input sample $\mathcal{S} = \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$, $\boldsymbol{w} \in \mathbb{R}^m$, $h : \mathcal{X} \to \mathbb{R}$, $M \neq 0$.
// in our case, $\boldsymbol{w} \leftarrow \boldsymbol{w}_t; h \leftarrow h_t; M \leftarrow M_t$ (current weights, weak hypothesis and max confidence, see Step 2.3 in SECBOOST and Assumption 5.1)
Step 1 : // all initializations

$$\eta_{\text{init}} \leftarrow \eta(\boldsymbol{w}, h); \tag{65}$$
$$\delta \leftarrow 1.0; \tag{66}$$
$$W_{\text{init}} \leftarrow 1.0; \tag{67}$$

Step 2 : **do** // Step 2 computes the leveraging coefficient α_t
$\alpha \leftarrow \delta \cdot \text{sign}(\eta_{\text{init}})$;
$\eta_{\text{new}} \leftarrow \eta(\tilde{\boldsymbol{w}}(\alpha), h)$;
if $|\eta_{\text{new}} - \eta_{\text{init}}| < |\eta_{\text{init}}|$ **then** `found_alpha` $\leftarrow$ `true` **else** $\delta \leftarrow \delta/2$;
while `found_alpha` = `false`;
Step 3 : $W \leftarrow$ Left Hand Side of (14) (main file) // Step 3 computes $\overline{W}_{2,t}$
// we can use (14) (main file) because we know α
if $W =_{\text{machine}} 0$ **then**
// the LHS of (14) is (machine) 0: just need to find W such that (15) holds !
$W \leftarrow W_{\text{init}}$;
while $|\alpha| > |\eta_{\text{init}}|/(W \cdot M^2)$ **do** $W \leftarrow W/2$;
endif
Step 4 : $b_{\text{sup}} \leftarrow |\eta_{\text{init}}|/(W \cdot M^2)$; // Step 4 computes ε_t
$\varepsilon \leftarrow (b_{\text{sup}}/\alpha) - 1$;
Return (α, W, ε);

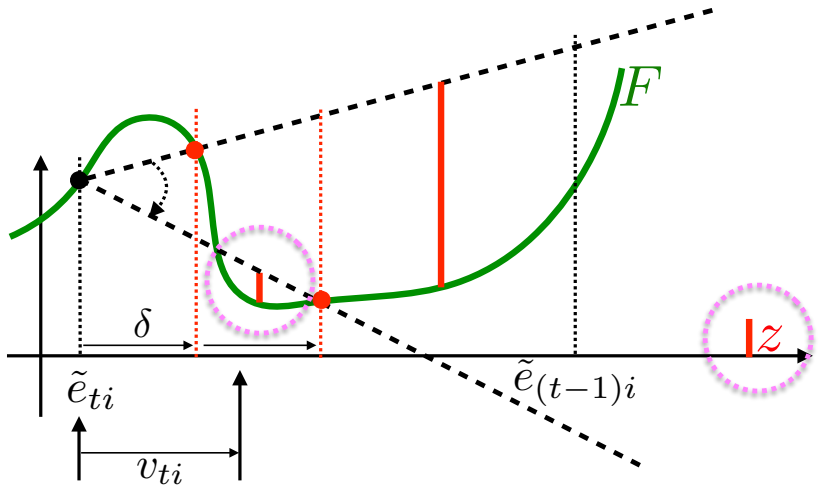


Figure III.1: How to find some $v \in \mathbb{I}_{ti}(z)$: parse the interval $[\tilde{e}_{ti}, \tilde{e}_{(t-1)i}]$ with a regular step δ, seek the secant with minimal slope (because $\tilde{e}_{ti} < \tilde{e}_{(t-1)i}$; otherwise, we would seek the secant with maximal slope). It is necessarily the one minimizing the OBI among all regularly spaced choices. If the OBI is still too large, decrease the step δ and start the search again.

III.2 Algorithm and implementation of the offset oracle

There exists a very simple trick to get some adequate offset v to satisfy (17) (main file), explained in Figure III.1. In short, we seek the optimally bended secant and check that the OBI is no more than a required z. This can be done via parsing the interval $[\tilde{e}_{ti}, \tilde{e}_{(t-1)i}]$ using regularly spaced values. If the OBI is too large, we can start again with a smaller step size. Algorithm OO_simple details the key part of the search.

Algorithm 4 OO_simple$(F, \tilde{e}_t, \tilde{e}_{t-1}, z, Z)$

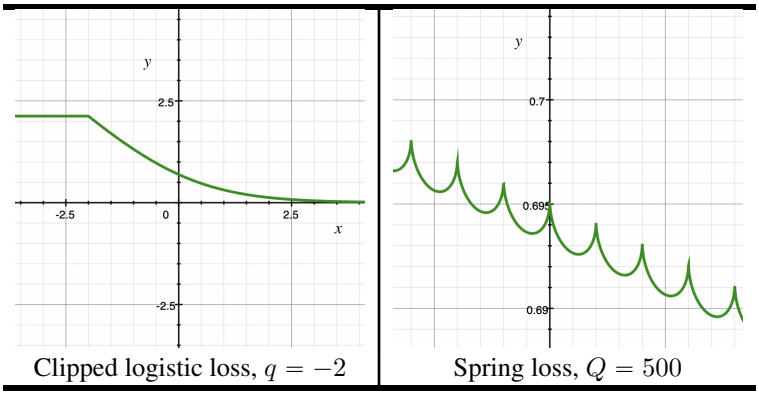
Input loss F, two last edges $\tilde{e}_t, \tilde{e}_{t-1}$, maximal OBI z, precision Z.
// in our case, $\tilde{e}_t \leftarrow \tilde{e}_{ti}; \tilde{e}_{t-1} \leftarrow \tilde{e}_{(t-1)i}$; (for training example index $i \in [m]$)
Step 1 : // all initializations

$$\delta \leftarrow \frac{\tilde{e}_{t-1} - \tilde{e}_t}{Z}; \tag{68}$$

$$z_c \leftarrow \tilde{e}_t + \delta; \tag{69}$$

$$i \leftarrow 0; \tag{70}$$

Step 2 : **do**
$s_c \leftarrow \text{SLOPE}(F, \tilde{e}_t, z_c)$;
// returns the slope of the secant passing through $(\tilde{e}_t, F(\tilde{e}_t))$ and $(z_c, F(z_c))$
if $(i = 0) \vee ((\delta > 0) \wedge (s_c < s_*)) \vee ((\delta < 0) \wedge (s_c > s_*)))$ **then** $s_* \leftarrow s_c; z_* \leftarrow z_c$
endif
$z_c \leftarrow z_c + \delta$;
$i \leftarrow i + 1$;
while $(z_c - \tilde{e}_t) \cdot (z_c - \tilde{e}_{t-1}) < 0$; // checks that z_c is still in the interval
Return $z_* - \tilde{e}_t$; // this is the offset v



Clipped logistic loss, $q = -2$ | Spring loss, $Q = 500$

Figure III.2: Crops of the two losses whose optimization has been experimentally tested with SECBOOST, in addition to the logistic loss. See text for details.

III.3 A toy experiments

We provide here a few toy experiments using SECBOOST. These are just meant to display that a simple implementation of the algorithm, following the blueprints given above, can indeed manage to optimize various losses. These are not meant to explain how to pick the best hyperparameters (*e.g.* (66)) nor how to choose the best loss given a domain, a problem that is far beyond the scope of our paper.

In this implementation, the weak learner learns decision trees and we minimize Matusita's loss at the leaves of decision trees to learn fixed size trees, see [34] for the criterion and induction scheme, which is standard for decision trees. SECBOOST is implemented as is given in the paper, and so are the implementation of SOLVE$_\alpha$ and the offset oracle provided above. We have made no optimization whasoever, with one exception: when numerical approximation errors lead to an offset that is machine 0, we replace it by a small random value to prevent the use of derivatives in SECBOOST.

We have investigated three losses. The first is the well known logistic loss:

$$F_{\text{LOG}}(z) \doteq \log(1 + \exp(-z)). \tag{71}$$

The other two are tweaks of the logistic loss. We have investigated a clipped version of the logistic loss,

$$F_{\text{CL},q}(z) \doteq \min\{\log(1 + \exp(-z)), \log(1 + \exp(-q))\}, \tag{72}$$

with $q \in \mathbb{R}$, which clips the logistic loss above a certain value. This loss is non-convex and non-differentiable, but it is Lipschitz. We have also investigated a generalization of the spring loss (main file):

$$F_{\text{SL},Q}(z) \doteq \log(1+\exp(-z)) + \frac{1-\sqrt{1-4\left(z_Q-[z_Q]\right)^2}}{Q}, \tag{73}$$

with $z_Q \doteq Qz - 1/2$ ([.] is the closest integer), which adds to the logistic loss regularly spaced peaks of variable width. This loss is non-convex, non-differentiable, non-Lipschitz. Figure III.2 provides a crop of the clipped logistic loss and spring loss we have used in our test. Notice the "hardness" that the spring loss intuitively represents for ML.

We provide an experiment on public domain UCI `tictactoe` [24] (using a 10-fold stratified cross-validation to estimate test errors). In addition to the three losses, we have crossed them with several other variables: the size of the trees (either they have a single internal node = stumps or at most 20 nodes) and, to give one example of how changing a (key) hyperparameter can change the result, we have tested for a scale of changes on the initial value of δ in (66). Finally, we have crossed all these variables with the existence of symmetric label noise in the training data, following the setup of [38, 40]. We flip each label in the training sample with probability η. Table III.3 summarizes the results obtained. One can see that SECBOOST manages to optimize all losses in pretty much all settings, with an eventual early stopping required for the spring loss if δ is too large. Note that the best initial value for δ depends on the loss optimized in these experiments: for $\delta = 0.1$, test error from the spring loss decreases much faster than for the other losses, yet we remind that the spring loss is just the logistic loss plus regularly spaced peaks. This could signal interesting avenues for the best possible implementation of SECBOOST, or a further understanding of the best formal ways to fix those paramaters, all of which are out of the scope of this paper.

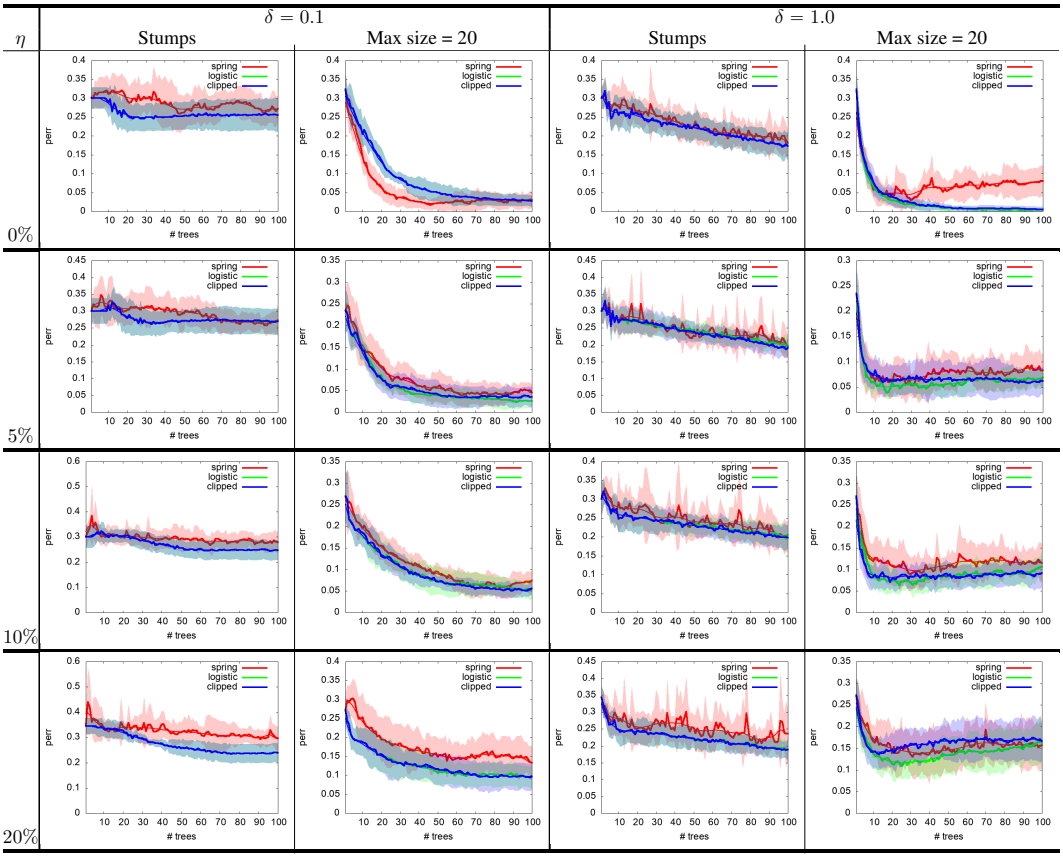


Figure III.3: Experiments on UCI `tictactoe` showing estimated test errors after minimizing each of the three losses we consider, with varying training noise level η, max tree size and initial hyperparameter δ value in (66). See text.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our paper is a theory paper: all claims are properly formalized and used.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion section is devoted to limitations and improvement of our results

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our paper is a theory paper: all assumptions, statements and proofs provided.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Though our paper is a theory paper, we have included in the supplement a detailed statement of all related algorithms and a toy experiment of a simple implementation of these algorithms showcasing a simple run on a public UCI domain.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Our paper is a theory paper. All algorithms we introduce are either in the main file or the appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: Our paper is a theory paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: Our paper is a theory paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA] .

   Justification: Our paper is a theory paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research of the paper follows the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper is a theory paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No release of data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: no outside code, data or models used requiring licensing.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.