Random Classification Noise does not defeat All Convex Potential Boosters Irrespective of Model Choice

Yishay Mansour [1 2] **Richard Nock** [2] **Robert C. Williamson** [3]

Abstract

A landmark negative result of Long and Servedio has had a considerable impact on research and development in boosting algorithms, around the now famous tagline that ”noise defeats all convex boosters”. In this paper, we appeal to the half-century+ founding theory of losses for class probability estimation, an extension of Long and Servedio's results and a new general convex booster to demonstrate that the source of their negative result is in fact the *model class*, linear separators. Losses or algorithms are neither to blame. This leads us to a discussion on an otherwise praised aspect of ML, *parameterisation*.

1. Introduction

In a now very influential paper, Long and Servedio (Long & Servedio, 2008b; 2010) made a series of observations on how simple symmetric label noise can "wipe out" the edge of a learner against the fair coin. The negative result is extreme in the sense that without noise, the learner fits a large margin, 100% accurate classifier but as soon as noise afflicts labels, *regardless of its magnitude*, the learner ends up with a classifier only as good as the fair coin; furthermore, the result also holds if we remove the algorithm from the equation and just focus on the convex loss' minimizer[*].

It is fair to say that the paper had sizeable impact on research and development in boosting at large, especially alongside the tagline retained by history that (emphasis ours)

noise defeats ***all*** *convex losses / boosters*,

as can be seen in papers (Amid et al., 2019a), theses (Ju, 2022), patents (Olabiyi et al., 2021), textbooks (Mohri et al., 2018), etc. (many others in Appendix, Section I)... (Notwithstanding mentions in the original papers of noise-tolerant partition-inducing boosting algorithms minimizing concave losses, thus *seemingly* following a different boosting blueprint (Long & Servedio, 2008b; 2010)).

Our contribution is primarily formal and shows that this tagline, taken at face value, is inaccurate. In the course of our arguments, we introduce a new convex booster that overcomes Long and Servedio's specific hardness result.

Our contribution starts with a striking paradox arising from the tagline above: when they are symmetric, *proper losses* (Savage, 1971) – loss functions eliciting Bayes optimal prediction and overwhelmingly popular in ML – have a dual surrogate form, for real-valued classification, which exactly fits to Long and Servedio's margin loss blueprint. Enters the aforementioned paradox: on Long and Servedio's data, such losses end up eliciting nothing better than a fair coin – quite arguably far from even the noise-dependent optimal prediction!

As we then show, this paradox has deeper roots in Savage's properness theory, as Long and Servedio's results survive to dropping the "symmetry" constraint on the loss. We thus extend their result to *any* strictly proper loss not necessarily admitting a margin form, albeit satisfying differentiability and lower-boundedness of the partial losses, which are weak constraints. Long and Servedio's result has no flaw, so the question that follows is naturally *what is the source of their negative result*?

We unveil the source of the paradox and show how to resolve it via a new convex booster. The functional pipeline that estimates class probabilities involves training a *model*, which is a linear model in Long & Servedio (2010). Informally, what we show – and which could be a tagline summary of our work – is that

linear models can break the promise of properness.

To show that, we introduce a simple and general "model-adaptive" convex booster (MODABOOST) following the "boosting blueprint" of Long & Servedio (2010). Through the reliance on an oracle called an *architecture emulation*

[1] Tel Aviv University [2] Google Research [3] University of Tübingen and Tübingen AI center. Correspondence to: Richard Nock <richardnock@google.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[*] Some convex boosters are consistent (Bartlett & Traskin, 2006; Telgarsky, 2013), but the results are based on assumptions that would not be met in the context of Long and Servedio.

oracle, MODABOOST boost a very general class of models that are able to emulate, among others, linear separators, decision trees, alternating decision trees, nearest neighbor classifiers and labeled branching programs. Our main theoretical result is a general margin / edge boosting rate theorem for MODABOOST. When applied to these standard model classes the algorithm works with rates of convergence tied to the model class. Of independent interest is the fact that apart from linear separators (Schapire et al., 1998), we are not aware of the existence of formal margin-based boosting results for any of the other classes. Since it complies with the blueprint boosting algorithm of Long and Servedio's negative results, when it learns linear separators on Long and Servedio's data, MODABOOST can spectacularly fail and early hit fair coin prediction; *however*, we formally show that if it boosts *any other* class mentioned in the list above on Long and Servedio's data, it does learn *Bayes optimal predictor regardless of the noise level*.

We finally discuss the implications of our findings in the context of a much praised aspect of ML: *parameterisation* – parameterisation of a loss that results in it being convex, of an algorithm that results in it emulating a boosting blueprint, of a model that results in a specific architecture, etc.. Our discussion goes beyond algorithms, losses and models. All proofs and additional applications of our algorithm are given in an Appendix denoted "APP" for short.

2. Definitions and setting

Losses for class probability estimation A *loss for class probability estimation* (CPE), $\ell : \mathcal{Y} \times [0,1] \rightarrow \mathbb{R}$, is

$$\ell(y,u) \quad \doteq \quad [\![y=1]\!] \cdot \ell_1(u) + [\![y=-1]\!] \cdot \ell_{-1}(u), \quad (1)$$

where $[\![.]\!]$ is Iverson's bracket (Knuth, 1992). Functions ℓ_1, ℓ_{-1} are called *partial* losses. A CPE loss is *symmetric* when $\ell_1(u) = \ell_{-1}(1-u), \forall u \in [0,1]$ (Nock & Nielsen, 2008), *differentiable* (resp. *lower-bounded*) when its partial losses are differentiable (resp. lowerbounded).

The pointwise conditional risk of local guess $u \in [0,1]$ with respect to a ground truth $v \in [0,1]$ is:

$$L(u,v) \quad \doteq \quad v \cdot \ell_1(u) + (1-v) \cdot \ell_{-1}(u). \quad (2)$$

A loss is *proper* iff for any ground truth $v \in [0,1]$, $L(v,v) = \inf_u L(u,v)$, and strictly proper iff $u = v$ is the sole minimiser (Reid & Williamson, 2011). The (pointwise) *Bayes* risk is $\underline{L}(v) \doteq \inf_u L(u,v)$. For proper losses, we thus have:

$$\underline{L}(v) \quad = \quad v \cdot \ell_1(v) + (1-v) \cdot \ell_{-1}(v). \quad (3)$$

Proper losses have a long history in statistics and quantitative psychology that long predates their use in ML (Reid & Williamson, 2010; Shuford et al., 1966). Hereafter, unless otherwise stated, we assume the following about the loss at hand:

1. $|\underline{L}(0)|, |\underline{L}(1)|, |\ell_1(1)|, |\ell_{-1}(0)| \neq \infty$;
2. the loss is strictly proper and differentiable (we call such losses SPD for short).

Conventional proper losses like the log-, square- or Matusita- are SPD losses with $\underline{L}(0) = \underline{L}(1) = \ell_1(1) = \ell_{-1}(0) = 0$. Losses satisfying $\ell_1(1) = \ell_{-1}(0) = 0$ are called *fair* in (Reid & Williamson, 2010).

Population loss Usually in ML, we are given a training sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ where $\boldsymbol{x}_i$ is an observation from a domain $\mathcal{X}$ and $y_i \in \mathcal{Y} \doteq \{0,1\}$ a binary representation for classes in a two-classes problem (0 goes for the "negative class," 1 for the "positive class"). In the CPE setting, we wish to learn an estimated posterior $\tilde{\eta} : \mathcal{X} \rightarrow [0,1]$, and to do so, following some of (Long & Servedio, 2010)'s notations, we wish to learn $\tilde{\eta}$ by minimizing a population loss called a risk:

$$\Phi(\tilde{\eta}, \mathcal{S}) \quad \doteq \quad \mathbb{E}_{i \sim [m]} \left[\ell(y_i, \tilde{\eta}(\boldsymbol{x}_i)) \right]. \quad (4)$$

We assume training on the whole domain to fit in the framework of (Long & Servedio, 2010), so the question of the generalisation abilities of models does not arise. In such a case, Bayes rule can be computed from the training data.

3. Surrogate losses and a proper paradox

Link, canonical losses The *(canonical) inverse link* of a SPD loss is:

$$\tilde{\eta}(z) \quad \doteq \quad (-\underline{L}')^{-1}(z). \quad (5)$$

One can check that for any SPD loss, $\mathrm{Im}(\tilde{\eta}) = [0,1]$, and it turns out that the inverse link provides a maximum likelihood estimator of the posterior CPE given a learned real-valued predictor $h : \mathcal{X} \rightarrow \mathbb{R}$ (Nock & Nielsen, 2008, Section 5). A substantial part of ML learns real-valued models (from linear models to deep nets) so the link is important to "naturally" embed the prediction in a CPE loss. A loss using its own link for the embedding is called a *canonical loss* (Reid & Williamson, 2010). One can use a different link, in which case the loss is called "composite" but technical conditions arise to keep the whole construction proper (Reid & Williamson, 2010). We thus restrict ourselves to using the canonical link for such purpose. When used with real-valued prediction, each SPD loss has a remarkable analytical form – called in general a *surrogate loss* (Nock & Nielsen, 2008) (and references therein).

Surrogate losses It comes from *e.g.* Nock & Menon (2020, Theorem 1) that any SPD loss can be written for a real valued classifier $h : \mathcal{X} \rightarrow \mathbb{R}$ on example $(\boldsymbol{x}, y)$ with binary-described class $y \in \mathcal{Y}$ as:

$$\begin{aligned}
\ell(y, h(\boldsymbol{x})) &= D_{-\underline{L}}\left(y \| -\underline{L}'^{-1}(h(\boldsymbol{x}))\right) \\
&\doteq -\underline{L}(y) + (-\underline{L})^\star(h(\boldsymbol{x})) - yh(\boldsymbol{x}) \\
&= -\underline{L}(y) + \underbrace{\phi_\ell(-h(\boldsymbol{x})) - yh(\boldsymbol{x})}_{\text{model dependent term}} \quad (6) \\
\text{with } \phi_\ell(z) &\doteq (-\underline{L})^\star(-z). \quad (7)
\end{aligned}$$

We single out function ϕ_ℓ to follow notations from (Long & Servedio, 2010) (we add ℓ in index to remind it depends on the loss). Here, $D_{-\underline{L}}$ is a Bregman divergence with generator $-\underline{L}$. We use the definition relying on the convex conjugate, see *e.g.* (Amari & Nagaoka, 2000). The convex conjugate of scalar function f is $f^\star(z) \doteq \sup_{z' \in \mathrm{dom} f} zz' - f(z')$ (Boyd & Vandenberghe, 2004). Note that (6) does not fit to the classical margin loss definition in ML (as in, *e.g.*, (Long & Servedio, 2010)), *however*, when the loss is in addition symmetric – which happens to be the case for most ML losses like log-, square-, Matusita, etc. –, the formula simplifies further to a margin loss formulation. Indeed, we remark $\underline{L}(u) = \underline{L}(1-u)$ and it comes $(-\underline{L})^\star(-z) = (-\underline{L})^\star(z) - z$. Using a "dual" class $y^* \in \mathcal{Y}^* \doteq \{-1, 1\}$ (1 still goes to the positive class), we can write the loss

$$\ell(y^*, h(\boldsymbol{x})) = -\underline{L}\left(\frac{1+y^*}{2}\right) + \underbrace{\phi_\ell(y^* h(\boldsymbol{x}))}_{\text{model dependent term}} \quad (8)$$

We have overloaded notation ℓ in (1) to *reparameterize* it as a function of the real-valued prediction h (instead of class probability u). This has an important consequence: the Bayes risk $\underline{L}$ in (3) is concave in its probability argument, while ϕ_ℓ in (7) is convex in its real-valued argument. Popular choices for ℓ, like log-, square-, Matusita, yield as popular forms for ϕ_ℓ, respectively logistic, square and Matusita. They are often called losses as well since they quantify a discrepancy, but equally often they are called *surrogates* (or surrogate losses) for the simple reason that when properly scaled, they yield upper bounds of the "historic loss" of ML, the 0/1 loss (Kearns et al., 1987), which with our notations equates $[\![\mathrm{sign}(h(\boldsymbol{x})) \neq y^*]\!]$. For learning, we can focus only in the model dependent term in (6), (8) and thus define the population (surrogate) risk as:

$$\begin{aligned}
\Phi(h, \mathcal{S}) &= \mathbb{E}_{i\sim[m]}\left[\ell(y_i^*, h(\boldsymbol{x}_i))\right] + \mathbb{E}_{i\sim[m]}\left[\underline{L}\left(\frac{1+y_i^*}{2}\right)\right] \\
&= \left\{ \begin{array}{ll} \mathbb{E}_{i\sim[m]}\left[\phi_\ell(-h(\boldsymbol{x}_i)) - y_i h(\boldsymbol{x}_i)\right] & \text{(general form)} \\ \mathbb{E}_{i\sim[m]}\left[\phi_\ell(y_i^* h(\boldsymbol{x}_i))\right] & \text{(for symmetric losses)} \end{array} \right. (9)
\end{aligned}$$

A Bayes born paradox There is one technical argument that needs to be shown to relate the surrogate form in (9) to (Long & Servedio, 2010)'s results: we need to show that the corresponding surrogates of any symmetric SPD loss fits to their blueprint margin loss.

Lemma 1. *For any* SPD *loss,* ϕ_ℓ *is* C^1*, convex, decreasing, has* ${\phi_\ell}'(0) < 0$ *and* $\lim_{z\rightarrow+\infty} \phi_\ell(z) = \underline{L}(0)$.

Proof in APP, Section II.1. Hence, if we offset the constant $\underline{L}(0)$ or just assume it is 0, any *symmetric* (8) SPD loss fits to Long & Servedio (2010, Definition 1). We now explain Long & Servedio (2010, Section 4)'s data. The domain $\mathcal{X} = \mathbb{R}^2$ and we have a (multi)set (or bag)

$$\mathcal{S}_{\text{clean}} \doteq \left\{ \left(\left[\begin{array}{c} 1 \\ 0 \end{array}\right], 1\right), \left(\left[\begin{array}{c} \gamma \\ -\gamma \end{array}\right], 1\right), \left(\left[\begin{array}{c} \gamma \\ -\gamma \end{array}\right], 1\right), \left(\left[\begin{array}{c} \gamma \\ K\gamma \end{array}\right], 1\right) \right\} (10)$$

In (Long & Servedio, 2010), $K = 5$ and $\gamma > 0$ is a margin parameter. Since all labels are positive, we easily get Bayes prediction, $\eta(\boldsymbol{x}) = 1 = \mathbb{P}[\mathsf{Y} = 1 | \mathsf{X} = \boldsymbol{x}]$. In the setting of (Long & Servedio, 2010), it is a simple matter to check that the optimal real-valued linear separator (LS) h minimizing $\Phi(h, \mathcal{S}_{\text{clean}})$ makes zero mistakes on predicting labels for $\mathcal{S}_{\text{clean}}$. One would expect this to happen since the loss ℓ at the core is proper, yet this seems to all go sideways *as soon as* label noise enters the picture. We replace $\mathcal{S}_{\text{clean}}$ by a "noisy" $\mathcal{S}_{\text{noisy}}$,

$$\mathcal{S}_{\text{noisy}} \doteq N \text{ copies of } \mathcal{S}_{\text{clean}} \cup 1 \text{ copy of } \mathcal{S}_{\text{clean}} \text{with labels flipped.} \quad (11)$$

This mimics a symmetric label noise level $\eta_{\mathsf{Y}} = 1/(N+1)$, with $N > 1$ (Long & Servedio, 2010). The paradox mentioned above comes from the following two observations: (i) Bayes posterior prediction with noise becomes $\eta(\boldsymbol{x}) = 1 - \eta_{\mathsf{Y}} > 1/2$, which still makes no error on $\mathcal{S}_{\text{clean}}$, and (ii) (Long & Servedio, 2010) show that regardless of this noise level, for any margin loss ϕ_ℓ complying with Lemma 1, the optimal model *is as bad as the fair coin on* $\mathcal{S}_{clean}$. Since symmetric SPD losses (9) fit to Lemma 1, (Long & Servedio, 2010)'s optimal model should have the same properties as Bayes' predictor, yet this clearly does not happen. The picture looks even gloomier as algorithms enter the stage: despite its acclaimed performances (Friedman et al., 2000), boosting can perform so badly that after a *single* iteration its "strong" model hits the fair coin prediction. Not only do we hit a paradox from the standpoint of the optimal model, we also observe a stark failure of a powerful algorithmic machinery. There is clearly something that "breaks" the ML pipeline.

In the context of properness, we have shown that the "margin form" parameterisation of the loss used by Long & Servedio (2010) is in fact not mandatory as asymmetric losses do not comply with it. Since asymmetry alleviates ties between partial losses, one could reasonably hope that it could address the paradox. We show that it is not the case.

4. Long and Servedio's results hold without symmetry

We reuse some of Long & Servedio (2010)'s notations and first denote B^{ideal}_{ϕ} the algorithm returning the optimal linear separator (LS) h minimizing (9).

Lemma 2. *For any $N > 1$, there exists $\gamma > 0, K > 0$ such that when trained on $\mathcal{S}_{noisy}$, B^{ideal}_{ϕ}'s classifier has at most 50% accuracy on $\mathcal{S}_{clean}$.*

Proof in APP, Section II.2. The proof displays an interesting phenomenon for asymmetric losses, which is not observed on Long & Servedio (2010)'s results. If the noise η_{Y} is large enough and the asymmetry such that $\phi_\ell{}'(0) < \eta_{\mathrm{Y}} - 1$, then the optimal classifier can do more than 50% mistakes on $\mathcal{S}_{\text{clean}}$ – thus perform worse than the unbiased coin. This cannot happen with symmetric losses since in this case $\phi_\ell{}'(0) = -1/2$ and we constrain $\eta_{\mathrm{Y}} < 1/2$. What this shows is that asymmetry, while accomodating non-trivial different misclassification costs depending on the class, can lead to non-trivial pitfalls over noisy data.

Similarly to Long & Servedio (2010), we denote $B^{\text{early}}_{\phi,T}$ the booster of (9) which proceeds by following the boosting blueprint as described in Long & Servedio (2010); we assume that the weak learner chooses the weak classifier offering the largest absolute edge (20), returning nil if all possible edges are zero (and then the booster stops). We let $\mathcal{S}_{\text{clean},\theta}, \mathcal{S}_{\text{noisy},\theta}$ denote $\mathcal{S}_{\text{clean}}, \mathcal{S}_{\text{noisy}}$ with observations rotated by an angle θ.

Lemma 3. *For any $N > 1, T \geqslant 1$, there exists $\gamma > 0, K > 0, \theta \in [0, 2\pi]$ such that when trained on $\mathcal{S}_{noisy,\theta}$, within at most T boosting iterations $B^{early}_{\phi,T}$ outputs a classifier at most 50% accurate on $\mathcal{S}_{clean,\theta}$.*

Proof in APP, Section II.3.

5. The boosting blueprint does provide a fix

We investigate a new boosting algorithm learning model architectures that generalise those of decision trees and linear separators, among other model classes. We call such models *partition-linear models* (PLM). The algorithm boosts any SPD loss using the blueprint boosting algorithm of (Long & Servedio, 2010). To our knowledge, it is the first boosting algorithm which can provably boost asymmetric proper losses, which is non trivial as it involves two different forms of the corresponding surrogate that are not compliant with the classical margin representation (Long & Servedio, 2010). A simple way to define a PLM H_t from sequence of triples $(\alpha_j, h_j, \mathcal{X}_j)_{j\in[t]}$ ($\alpha_j \in \mathbb{R}, h_j \in \mathbb{R}^{\mathcal{X}}, \mathcal{X}_j \subseteq \mathcal{X}$) is, for $t \geqslant 1$:

$$H_t(\boldsymbol{x}) \doteq \begin{cases} H_{t-1}(\boldsymbol{x}) + \alpha_t h_t(\boldsymbol{x}) & \text{if} \quad \boldsymbol{x} \in \mathcal{X}_t \\ H_{t-1}(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{12}$$

$$= \sum_{t=1}^{T} [\![\boldsymbol{x} \in \mathcal{X}_t]\!] \cdot \alpha_t h_t(\boldsymbol{x}), \tag{13}$$

and we add $H_0(\boldsymbol{x}) \doteq 0, \forall \boldsymbol{x} \in \mathcal{X}$[†]. We also define the weight function

$$w((\boldsymbol{x}, y), H) \doteq y - y^* \cdot (-\underline{L}')^{-1}(H(\boldsymbol{x})), \tag{14}$$

which is in $[0, 1]$. Notice we use both (real and binary) class encodings in the weight function, recalling the relationship $y^* \doteq 2y - 1 \in \{-1, 1\}$. Algorithm MODABOOST presents the boosting approach to learning PLM. MODABOOST contains the core of boosting algorithms in Steps 2.2, 2.3 and 2.4, albeit in a slightly more general form than the classical blueprint, in part because the losses it optimizes can be asymmetric.

The architecture emulation oracle MODABOOST also contains a new component that we believe has no equivalent in previous (formal) boosting algorithms: what we denote as a *architecture emulation oracle* (AEO). What AEO effectively does is design a subset of the domain from which to compute a part of the training sample to give to the weak learner, to train the next weak classifier. Its *aim* is to design this subset in a way that the PLM learned *emulates* a specific model architecture (hence its name). At the end of training, we can then represent the PLM learned using a model architecture that is more familiar to the experimenter. For example, it is possible to use MODABOOST to learn decision trees (see application #2). In this case, AEO returns the subset of $\mathcal{X}$ corresponding to a leaf of the emulated current decision tree, leaf which will then be equivalently split by a specific choice of the weak classifier in Step 2.2. It is also possible to use MODABOOST to boost a nearest neighbor classifier (see application #4): in this case, AEO returns the subset of $\mathcal{X}$ corresponding to the observations for which a specific training point would vote.

Though each architecture choice is accompanied by a specific choice of AEO, there is in theory no restriction to the design of the oracle. The choice however *does* influence the boosting rates (Definition 5.1 and Theorem 1). Also, the requirements that AEO's inputs are the domain and current classifier in MODABOOST can be widened to accomodate for more architectures.

[†] We can equivalently consider that $h_t = 0$ in $\mathcal{X} \backslash \mathcal{X}_t$. We opt for (12) since it makes a clear distinction for $\mathcal{X}_t$. Notice that this setting generalizes boosting with weak hypotheses that abstain (Schapire & Singer, 1998).

```
Algorithm 1 ModaBoost($\mathcal{S}, \ell, \textsc{wl}, \textsc{aeo}, T$)
  Input: Dataset $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, SPD loss $\ell$, weak learner WL, architecture emulation oracle AEO, iteration number $T \geqslant 1$;
  Output: PLM $H_T$;
  Step 1 : $\forall i \in [m], w_{i,1} \doteq w((\boldsymbol{x}_i, y_i), H_0)$ // weight initialisation
  Step 2 : for $t = 1, 2, ..., T$
        Step 2.1 : $\mathcal{X}_t \leftarrow \textsc{aeo}(\mathcal{X}, H_{t-1})$;
        Step 2.2 : $h_t \leftarrow \textsc{wl}(\boldsymbol{w}_t^*, \mathcal{S}_t)$;
     // weak learner call: $\mathcal{S}_t \doteq \{(\boldsymbol{x}_i, y_i) \in \mathcal{S} : \boldsymbol{x}_i \in \mathcal{X}_t\}$;
     // $\boldsymbol{w}_t^* \doteq \boldsymbol{w}_t$ restricted to $\mathcal{S}_t$;
        Step 2.3 : compute $\alpha_t$ as the solution to:

        $\sum_{i \in [m]_t} w((\boldsymbol{x}_i, y_i), H_t) \cdot y_i^* h_t(\boldsymbol{x}_i) = 0; \quad (15)$

     // $[m]_t$ = indices of $\mathcal{S}$ in $\mathcal{S}_t$; $\alpha_t$ appears in $H_t$ (12)
        Step 2.4 : $\forall i \in [m]_t, w_{t+1,i} \doteq w((\boldsymbol{x}_i, y_i), H_t)$
     // weight update
  return $H_T(\boldsymbol{x}) \doteq \sum_{t=1}^T 1_{\boldsymbol{x} \in \mathcal{X}_t} \cdot \alpha_t h_t(\boldsymbol{x})$;
```

Solutions to (15) are finite We assume without loss of generality that $\pm h_t$ does not achieve 100% accuracy over $\mathcal{S}_t$ (Step 2.2; otherwise there would be no need for boosting, at least in $\mathcal{X}_t$) and that $\max_{\mathcal{S}_t} |h_t| \ll \infty$, "$\ll \infty$" denoting finiteness.

Lemma 4. *The solution to* (15) *satisfies* $|\alpha_t| \ll \infty$.

Proof in APP, Section II.5.

The boosting abilities of ModaBoost Given a real valued classifier H and an example $(\boldsymbol{x}, y^*)$, we define the (unnormalized) edge or margin of H on the example as $y^* H(\boldsymbol{x})$ (Nock & Nielsen, 2008; Schapire et al., 1998), a quantity that integrates both the accuracy of classification (its sign) and a confidence (its absolute value). Formal guarantees on edges / margins are not frequent in boosting (Nock & Nielsen, 2007; Schapire et al., 1998). We now provide one such general guarantee for ModaBoost. While requirements on the weak hypotheses follow the weak learning assumption of boosting, the constraints on the loss itself are minimal: they essentially require it to be SPD with partial losses meeting a lower-boundedness condition and a condition on derivatives.

Definition 5.1. *Let $\{u_t\}_{t \in \mathbb{N}_{>0}}$ be a sequence of strictly positive reals. We say that the architecture emulation oracle in Step 2.1 of* ModaBoost *is "u_t compliant" iff, letting $J(\mathcal{W}, t) \doteq \mathrm{Card}(\mathcal{W}) \cdot (\mathbb{E}_{i \sim \mathcal{W}}[w_{t,i}])^2$ where $\mathcal{W} \subseteq [m]$, Step 2.1 guarantees:*

$$J([m]_t, t) \quad \geqslant \quad u_t \cdot J([m], t), \forall t = 1, 2, ..., \qquad (16)$$

The fact that such a sequence exists will be denoted architecture emulation oracle compliance (AEOC).

Notice that the sum of terms $\sum_{t=1}^T u_t$ is strictly increasing and thus invertible. Let $U : \mathbb{N}_{>0} \to \mathbb{R}_+$ such that

$$U(T) \quad \doteq \quad \sum_{t=1}^{T} u_t. \qquad (17)$$

The role of J is fundamental in our results and can guide the choice of $\mathcal{X}_t$ in Step 2.1: in short, the larger u_t, the better the rates. Lemma 6 gives a concrete and intuitive simplification of J in the case of decision trees. In the most general case, it is good to keep in mind the intuition of boosting that the weight of an example is larger as the outcome of the current classifier gets *worse*. Hence, (16) encourages focus on $\mathcal{X}_t$ with a large number of examples ($\mathrm{Card}([m]_t)$) *and* with large weights ($\mathbb{E}_{i \sim [m]_t}[w_{t,i}]$) – hence with subpar current classification.

Theorem 1. *Suppose the following assumptions are satisfied on the loss and weak learner:*

LOSS *the loss is strictly proper differentiable; its partial losses are such that $\exists \kappa > 0, C \in \mathbb{R}$,*

$$\ell_{-1}(0), \ell_1(1) \quad \geqslant \quad C, \qquad (18)$$
$$\inf\{\ell'_{-1} - \ell'_1\} \quad \geqslant \quad \kappa. \qquad (19)$$

WLA *There exists a constant $\gamma_{\text{WL}} > 0$ such that at each iteration $t \in [T]$, the weak hypothesis h_t returned by* WL *satisfies*[‡]

$$\left| \sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i^* \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|} \right| \geqslant \gamma_{\text{WL}} (20)$$

AEOC *there exists a sequence $\{u_t\}_{t \in \mathbb{N}_{>0}}$ of strictly positive reals such that the choice of $\mathcal{X}_t$ in Step 2.1 is u_t compliant.*

Then for any $\theta \geqslant 0, \varepsilon > 0$, letting $\underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$, if ModaBoost *is run for at least*

$$T \quad \geqslant \quad U^{-1}\left(\frac{2\,(\Phi(H_0, \mathcal{S}) - C)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma_{\text{WL}}^2}\right) \qquad (21)$$

iterations, then we are guaranteed

$$\mathbb{P}_{i \sim [m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] \quad < \quad \varepsilon. \qquad (22)$$

Here, U is crafted as in (17).

Proof in APP, Section II.6. The proof of the Theorem involves as intermediate step the proof that the surrogate

[‡] The quantity in the absolute value is sometimes called the (normalized) edge of h_t; it takes values in $[-1, 1]$.

$\Phi(H_T, \mathcal{S})$ is also boosted, which is of independent interest given Long & Servedio (2010)'s framework and the potential asymmetry of the loss (Theorem B in APP).

Remark 1. *The **LOSS** requirements are weak. It can be shown that strict properness implies* $\inf\{\ell'_{-1} - \ell'_1\} > 0$ *(Reid & Williamson, 2010, Theorem 1); since the domain of the partial losses is closed, we are merely naming the strictly positive infimum with condition* $\inf\{\ell'_{-1} - \ell'_1\} \geqslant \kappa > 0$*. The "extremal" value condition for partial losses (*$\ell_{-1}(0), \ell_1(1) \geqslant C$*) is also weak as if it did not hold, partial losses would not be lower-bounded on each's respective best possible prediction, which would make little sense. Usually,* $C = 0$ *(the best predictions occur no loss) such as for the square-, log-, Matusita losses.*

We now give five possible instantiations of ModaBoost, each with a separate implementation of AEO and thus a separate discussion about u_t compliance and boosting rates. We start by the two most important ones: linear separators and decision trees.

Application of ModaBoost #1: linear separators (LS) This is a trivial use of ModaBoost.

▻ *u_t compliance of AEO and the weak learner*: $\mathcal{X}_t = \mathcal{X}, \forall t$ so we trivially have $u_t = 1 (\forall t)$ compliance and the weak learner returns an index of a feature to leverage.

▻ *Boosting rate*: we have the guarantee that $\mathbb{P}_{i\sim[m]}[y^*_i H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon$ if

$$T \geqslant \underbrace{\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma^2_{\text{WL}}}}_{\doteq \mathsf{B}_{\mathsf{LS}}} = \tilde{O}\left(\frac{1}{\varepsilon^2 \gamma^2_{\text{WL}}}\right), \quad (23)$$

a dependence (the tilde removes dependences in other factors) that fits to the general optimal lower-bound in γ_{WL} (Alon et al., 2021) but is suboptimal in ε, albeit not far from lowerbound $O(1/\varepsilon)$ (Telgarsky, 2012). The algorithm and its analysis generalise Nock & Nielsen (2008)'s approach.

▻ *Effect of Long and Servedio's data*: Since ModaBoost falls in the negative result's boosting blueprint of Long & Servedio (2010, Section 2.5), it does face the negative result of (Long & Servedio, 2010). A simple way to demonstrate that is a toy experiment using ModaBoost with LS provided in APP, Section III. It clearly displays that accuracy dramatically falls below a "threshold" margin parameter γ, an observation formally shown in Long & Servedio (2010).

Application of ModaBoost #2: decision trees (DT) This is a slightly more involved use of ModaBoost, from the "location" of the weak learner to the perhaps surprising observation that in this case, ModaBoost emulates and generalizes well known top-down induction schemes.

▻ *u_t compliance of AEO and the weak learner*: we investigate general u_t compliance, where $[m]_t \in \mathcal{P}([m])$, for $\mathcal{P}([m])$ a partition of $[m]$ in N_t subsets. Jensen's inequality yields

$$\begin{aligned}\sum_{\mathcal{W}\in\mathcal{P}([m])} J(\mathcal{W}, t) &= m \cdot \sum_{\mathcal{W}\in\mathcal{P}([m])} \frac{J(\mathcal{W}, t)}{m} \\ &= m \cdot \mathbb{E}_{\mathcal{W}\sim\mathcal{P}([m])}\left(\mathbb{E}_{i\sim\mathcal{W}}[w_{t,i}]\right)^2 \\ &\geqslant m \cdot \left(\mathbb{E}_{\mathcal{W}\sim\mathcal{P}([m])}\left[\mathbb{E}_{i\sim\mathcal{W}}[w_{t,i}]\right]\right)^2 \\ &= m \cdot \left(\mathbb{E}_{i\sim[m]}[w_{t,i}]\right)^2 = J([m], t),\end{aligned}$$

therefore there exists $\mathcal{W}^* \in \mathcal{P}([m])$ such that $J(\mathcal{W}^*, t) \geqslant (1/N_t) \cdot J([m], t)$ and picking any such "heavy" subset of indices $[m]_t = \mathcal{W}^*$ guarantees u_t compliance for $u_t = 1/N_t$. AEO makes ModaBoost grow a DT "in disguise" by computing as $\mathcal{X}_t$ the domain of a leaf in the current tree, initialized to a single root (thus, $\mathcal{X}_t = \mathcal{X}$ for the first iteration). When the current decision tree has t leaves, we see that we can guarantee $u_t \geqslant 1/t$. The weak learner is used to find splits in a way we now describe, which will be followed by how the PLM learned indeed emulates a decision tree, and how ModaBoost ends up being able to emulate well known top-down induction algorithms for DT induction.

Regarding the weak learner, ModaBoost iteratively replaces a leaf in the current tree by a decision stump. There are two strategies for that: the first consists in asking the weak learner for one complete split, just like in (Kearns & Mansour, 1996), but ModaBoost would then fit a single correction (leveraging coefficient $\alpha_.$) for both leaves and this would be suboptimal. To correct every single leaf prediction separately, we let the weak learner return a split and a corresponding real-valued prediction for *half the split*, *e.g.* for "split_predicate = true". It is easy to show that if this meets the **WLA**, then so does *the other half* (for "split_predicate = false"). In other words, we get two **WLA** compliant weak hypotheses for the price of a single query to the weak learner, and both turn out to define the split sought. This is formalized in the following Lemma, which assumes wlog that the split variable is x_i, continuous.

Lemma 5. *Suppose the weak learner returns* $1_{x_i \geqslant a} \cdot h_t$ *(*$h_t \in \mathbb{R}_{>0}$ *constant) that meets the **WLA** for the half split. Then the "companion" hypothesis* $h'_t(\boldsymbol{x}) \doteq 1_{x_i < a} \cdot (-h_t)$ *satisfies the **WLA**.*

Proof in APP, Section II.7. The choice of the leaf to split is simple: denote $\Lambda(H)$ the set of leaves of DT H and λ a general leaf. Since the leaves of a DT induce a partition of the tree, we denote $J(\lambda)$ the expression of $J(\mathcal{W}, t)$ for $\mathcal{W} = \{i : \boldsymbol{x}_i \text{ reaches } \lambda\}$, omitting index t for readability. Let us analyze what $\mathcal{W}^*$ would satisfy in this case.

Lemma 6. *We have*

$$J(\lambda) \quad \propto \quad p_\lambda \cdot (\underbrace{p^+_\lambda(1-p^+_\lambda)}_{\doteq \underline{L}^{\mathrm{SQ}}(p^+_\lambda)})^2, \qquad (24)$$

where $p_\lambda \doteq m_\lambda/m, p^+_\lambda \doteq m^+_\lambda/m_\lambda$, $m_\lambda \doteq \mathrm{Card}(\{i : \boldsymbol{x}_i \textit{ reaches } \lambda\}), m^+_\lambda \doteq \mathrm{Card}(\{i : \boldsymbol{x}_i \textit{ reaches } \lambda \wedge y_i = 1\})$ *and* $\underline{L}^{\mathrm{SQ}}(u) = u(1-u)$ *is Bayes risk for the square loss.*

Proof in APP, Section II.8. Hence, the leaf to split in Step 2.1 has a good compromise between its "weight" (p_λ) and its local error (since $2p^+_\lambda(1-p^+_\lambda) \geqslant \min\{p^+_\lambda, 1-p^+_\lambda\}$). In traditional "tree-based" boosting papers (from Kearns & Mansour (1996) to Nock & Williamson (2019)), one usually picks the heaviest leaf ($= \arg\max_\lambda p_\lambda$) but it may well be a leaf with zero error – thus preventing boosting through splitting. Inversely, focusing only on large error to pick a leaf might point to leaves with too small weights to bring overall boosting compliance. Criterion $J(.)$ strikes a balance weight vs error in the choice.

▷ *Boosting rate*: We have $u_t \geqslant 1/t$, with $\sum_{t=1}^T u_t \geqslant \int_0^T \mathrm{d}z/(1+z) = \log(1+T) \doteq U(T)$, and so we are guaranteed that $\mathbb{P}_{i\sim[m]}[y^*_i H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon$ if

$$\underbrace{T \geqslant \exp\left(\frac{2\left(\Phi(H_0,\mathcal{S})-C\right)}{\kappa\cdot\varepsilon^2\underline{w}(\theta)^2\gamma^2_{\mathrm{WL}}}\right)}_{\doteq \mathrm{B}_{\mathrm{DT}}} = \exp\tilde{O}\left(\frac{1}{\varepsilon^2\gamma^2_{\mathrm{WL}}}\right), \qquad (25)$$

which is comparable at $\theta = 0$ to the bound of Kearns & Mansour (1996, Theorem 1) for CART and otherwise generalizes their results to margin/edge-based bounds.

▷ *Miscellaneous*: we finish by a last analogy between ModaBoost and classical DT induction algorithms: there is a simple closed form solution for the leveraging coefficients $\alpha_.$ that simplifies the loss.

Lemma 7. *Running* ModaBoost *to learn a decision tree* H *gives* $\Phi(H,\mathcal{S}) = \mathbb{E}_{\lambda\sim\Lambda(H)}\left[\underline{L}(p^+_\lambda)\right]$, *where we recall* $p^+_\lambda \doteq m^+_\lambda/m_\lambda$ *and the weight of* λ *is* m_λ/m. *Furthermore, the* ModaBoost *prediction computed at leaf* λ, H_λ, *is* $H_\lambda = (-\underline{L}')\left(p^+_\lambda\right)$.

Proof in APP, Section II.9. We conclude that running ModaBoost to learn a decision tree is largely equivalent to the minimisation of classical DT induction criteria (Breiman et al., 1984; Quinlan, 1993; Kearns & Mansour, 1996; Nock & Williamson, 2019), and our boosting rate analysis generalizes those to asymmetric losses and edge / margin bounds. One can also finally notice that we can easily transform a DT learned using ModaBoost to a classical DT by "percolating" values down to the leaves, see Figure 1. such a connection between both types of models is not new as it dates back to Henry et al. (2007) and was later exploited in various work (*e.g.* Luna et al. (2019)).

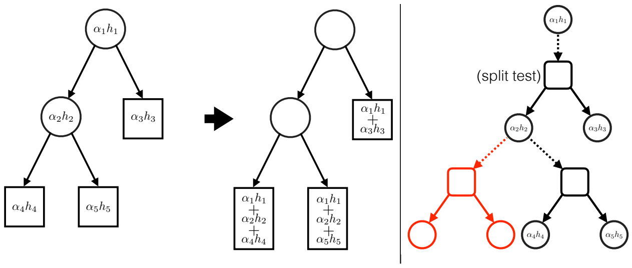

$\alpha_1 h_1$
$\alpha_2 h_2$
$\alpha_3 h_3$
$\alpha_4 h_4$
$\alpha_5 h_5$
$\alpha_1 h_1 + \alpha_3 h_3$
$\alpha_1 h_1 + \alpha_2 h_2 + \alpha_4 h_4$
$\alpha_1 h_1 + \alpha_2 h_2 + \alpha_5 h_5$
(split test)

Figure 1. Left: DT learned using ModaBoost and its "classical" equivalent DT. *Right*: an equivalent representation using an alternating decision tree (black) and a more general ADT (black + red).

▷ *Effect of Long and Servedio's data*: the triple (N, K, γ) being fixed in (10), (11), we say that ModaBoost with model set $\mathcal{H}$ is *Bayes optimal in* T *iterations on Long and Servedio's data* iff when run on Long & Servedio (2010)'s *noisy* $\mathcal{S}_{\text{noisy}}$ for T iteration, ModaBoost returns $H_T \in \mathcal{H}$ which has 100% accuracy on Long & Servedio (2010)'s *clean data* $\mathcal{S}_{\text{clean}}$.

Lemma 8. *For any* $(N, K, \gamma) \in \mathbb{N}^2_{>0} \times \mathbb{R}_{>0}$, ModaBoost *with* DT *is Bayes optimal in 1 iteration on Long and Servedio's data if the loss is symmetric.*

The proof, in APP, Section II.10, shows a much more general result, in particular encompassing asymmetric losses as well, a case a bit trickier to handle in terms of noise level.

Application of ModaBoost #3: alternating decision trees (ADT) Alternating decision trees were introduced in (Freund & Mason, 1999). An ADT roughly consists of a root constant prediction and a series of stumps branching from their leaf prediction nodes in a tree graph, see Figure 1. The equivalent ADT representation of a DT would have outgoing degree 1 for all these stumps' leaves. A general ADT makes this outdegree variable and a prediction is just the sum of the prediction along all paths an observation can follow from the ADT's root node. If a stumps' leaf branches on N stumps, then we sum the N corresponding predictions (and not just 1 for a DT). While using such models is interesting in terms of model's parameterisation, one also sees advantages in terms of boosting, since summing boosted predictions (23) is more efficient than branching on boosted predictions (25), but the paper of (Freund & Mason, 1999) contains no such rate (note that the loss optimized here is AdaBoost's exponential loss, which is not proper canonical).

▷ *u_t compliance of* AEO *and the weak learner*: these are just combinations of those for LS (when increasing a stump's leaf outgoing degree with a new stump) and DT (when finding the test of a stump). Denote N-ADT the set of ADTs where non-leaf prediction nodes' outdegree is fixed to be N (inclusive of the root node). Notice that we can then boost while guaranteeing that $u_t = 1$ for N boosting iterations (at the root), then $u_t \geqslant 1/2$ for N boosting iterations and so on until the last N iterations with $u_t \geqslant N/T$.

▻ Boosting rate: assuming wlog T a multiple of N, we have thus $\sum_{t=1}^{T} u_t \geqslant N \cdot \sum_{t=1}^{T/N} 1/t \geqslant N \cdot \int_0^{T/N} \mathrm{d}z/(1+z) = N \cdot \log(1+(T/N)) \doteq U(T)$, and so we are guaranteed that $\mathbb{P}_{i\sim[m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon$ if

$$T \geqslant \underbrace{N \cdot \exp\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{N\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma^2_{\mathrm{WL}}}\right)}_{\doteq \mathrm{B}_{\mathsf{ADT}}} = N \exp \tilde{O}\left(\frac{1}{N\varepsilon^2\gamma^2_{\mathrm{WL}}}\right).$$

Bearing in mind that $\mathrm{B}_{\mathsf{DT}}, \mathrm{B}_{\mathsf{ADT}}$ are non-tight lowerbounds, in such a regime, it is easy to see that an ADT can be exponentially more efficient than a DT, boosting-wise: for example, letting $N = \sqrt{\mathrm{B}_{\mathsf{LS}}}$, we obtain $\mathrm{B}_{\mathsf{ADT}} \leqslant \exp(-M\sqrt{\mathrm{B}_{\mathsf{LS}}}) \cdot \mathrm{B}_{\mathsf{DT}}$ for some constant $M > 0$.

▻ Effect of Long and Servedio's data: a single node ADT is also a single node DT. Since learning a DT achieves Bayes optimal prediction with a single root DT on Long & Servedio (2010), the same happens for a single root ADT.

Lemma 9. *For any $(N, K, \gamma) \in \mathbb{N}^2_{>0} \times \mathbb{R}_{>0}$, ModaBoost with ADT is Bayes optimal in 1 iteration on Long and Servedio's data if the loss is symmetric.*

Application of ModaBoost #4: (leveraged) nearest neighbors (NN) nearest neighbor classification is one of the oldest supervised learning techniques (Cover & Hart, 1967). Since we consider real-valued prediction, we implement NN classification by summing a real constant prediction at one observation's neighbors and assume that tie neighbors are included in the voting sample (so one observation can end up with more than K_{NN} neighbors). Local predictions can have varying magnitudes, which represents a generalisation of nearest neighbor classification where magnitude is constant, but we still call such classifiers nearest neighbors, omitting the "leveraging" part.

▻ u_t compliance of AEO and the weak learner: the weak learner returns an example to leverage and thus $\mathcal{X}_t$ is its *reciprocal neighborhood* (the set of examples for which it belongs to the K-NN). We assume wlog there are no "outliers" for classification, so the minimum size of this neighborhood is some $K_{\mathrm{rec}} \geqslant K_{\mathrm{NN}} > 0$, yielding $u_t = K_{\mathrm{rec}}/m, \forall t$.

▻ Boosting rate: we get $\mathbb{P}_{i\sim[m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon$ if

$$T \geqslant \frac{2m\left(\Phi(H_0, \mathcal{S}) - C\right)}{K_{\mathrm{rec}}\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma^2_{\mathrm{WL}}} = \frac{m\mathrm{B}_{\mathsf{LS}}}{K_{\mathrm{rec}}} = \tilde{O}\left(\frac{m}{K_{\mathrm{rec}}\varepsilon^2\gamma^2_{\mathrm{WL}}}\right), \quad (26)$$

a bound substantially better and more general than Nock et al. (2015, Theorem 4), which holds for $\theta = 0$: namely, our assumptions are weaker, our result cover asymmetric losses and the dependency of (26) in K_{rec} is better.

▻ Effect of Long and Servedio's data: it is not hard to see that the problem is equivalent to leveraging a constant prediction using all examples with a specific observation and the leveraging coefficient is the same as for a DT where the root node's support is restricted to the given observation. This applies for any choice of $K_{\mathrm{NN}} \geqslant 1$ neighbors for NN and we get the following.

Lemma 10. *For any $(N, K, \gamma) \in \mathbb{N}^2_{>0} \times \mathbb{R}_{>0}$ and any $K_{\mathrm{NN}} \geqslant 1$, ModaBoost with NN is Bayes optimal in 1 iteration on Long and Servedio's data if the loss is symmetric.*

Application of ModaBoost #5: labeled branching programs (LBP) A labeled branching program is a branching program (Mansour & McAllester, 2000) with real prediction values at each node, just like our encoding of DT, with the same way of classifying an observation – sum an observation's path values from the root to a leaf. The key difference with classical branching programs is that to one leaf can correspond as many possible predictions as there are paths leading to it. To save space, this application is fully developed in APP, Section II.11. Notably, while ModaBoost's boosting rate is suboptimal compared to the $\tilde{O}(\log^2(1/\varepsilon))$ dependence of Mansour & McAllester (2000) shown for $\theta = 0$, it achieves an exponential improvement over ModaBoost's rate with DT.

6. Discussion: on parameterization

A partial explanation for the confusion about the results of Long & Servedio (2010) can be offered via the notion of *parameterization*. We elaborate alongside three key ingredients: losses, models and algorithms. Much ML research seems blind to the difference between a change of object, and a change in the parametrization of an object. The clearest example of this is with losses, where it is known (van Erven et al., 2012) that the features of a loss function that govern its mixability depend only upon the induced geometry of its superprediction set. From this perspective, losses are better thought of, and analysed in terms of the sets that they induce (Cranko et al., 2019). The commonplace desire that the loss *function* be convex (as a function) is controllable, independently, via a *link* (Reid & Williamson, 2010; Williamson et al., 2016). Mathematically, the introduction of the link is tantamount to an (invertible, smooth) reparameterization of the loss.

There is one other point to be made about the loss function. The abstract idea of a loss function was developed by Wald (1950) as a formalisation of the notion that when solving a data-driven problem, one ultimately has some goal in mind, and that can be captured by an outcome-contingent utility (Berger, 1985), or 'loss'. Thus the loss is part of the problem statement. In contrast, in the ML literature, such as that arising from Long & Servedio (2010), a loss function is considered as part of the specification of a 'learning algorithm' (means of solving the problem). From Wald's perspective, all of the work inspired by Long & Servedio (2010) is a perhaps not so surprising side-effect of attempting to solve one problem (classification using 0-1 loss) by using a method that utilises a *different* loss. If one tries to repeat the negative example of Long & Servedio (2010) without the use of a surrogate, and always in terms of the Bayes optimal, there is nothing to see. When one adds some noise, the Bayes risk may change, but one will not see the apparent paradoxes of Long & Servedio (2010). Recently, there has been a burst of research around new loss functions whose formulation aims to reduce the difficulty of the learning task, some becoming overwhelmingly popular (Lin et al., 2017). One can see benefits of such a substantial shift from the normative view (of properness) to a more user-centric "*à-la-Wald*" design, but it usually comes with overloading loss functions with new hyperparameters. Technically, quantifying properties of the minimizers — in effect, answering the question "*what can be learned from this loss*" — can be non-trivial (Sypherd et al., 2022) but it is an important task: Long and Servedio's result brightly demonstrates that some choices can be statistically "unsuitable" (*e.g.* linear classifier, convex loss) if training data is subject to corruption. One would have reasons to stick with linear separators *e.g.* for their simplicity and interpretability. In such a case, one might have to break properness and eventually convexity of the loss, as *e.g.* recently shown in Sypherd et al. (2023).

A less widespread example is the reparameterization of a model class. It is known that the statistical complexity of a learning problem in the statistical batch setting is controlled by the complexity of the model class. This complexity is in terms of the class considered as a set, and is not influenced by how the elements of the class are parameterized. Thus the statistical complexity of learning with a model class comprising rational functions of degree n will not depend upon whether the functions are parameterized in factored form, as partial fractions, or as ratios of polynomials in canonical sum form. Lest it be objected that no-one would use such a strange class, we note that in the simplest case analysable, classical sigmoidal neural networks can be reparameterized in terms of rational functions, and thus at least these three parameterizations are open to use (Williamson & Helmke, 1995). The parameterization, whilst not changing the model class (or, say its VC dimension) *will* change the behaviour of learning algorithms, in particular gradient based algorithms, which can misbehave due to attractors at infinity (Blackmore et al., 1996) — a phenomenon caused by parameterization.

The final ingredient to consider is the algorithm. We have demonstrated that a boosting algorithm can be constructed that works successfully in the noisy situation (when using suitable model classes), but we have not really addressed head-on the perhaps more direct response to (Long & Servedio, 2010), which is to challenge the definition of what is, and what is not, a 'boosting algorithm.' There are several obvious ways to proceed here (e.g. in terms of what the boosting algorithm is provided as input, in the form of weak learners). But all such attempts stumble over a more challenging issue: namely that there is no sensible way to compare algorithms — we cannot even say 'when is one algorithm equal to another?' (Blass et al., 2009). The irony is that the object that is most valorised in machine learning research, namely the algorithm, hardly satisfies the conceptual properties one demands of any 'object' — namely that we can tell when two objects are the same or different. We do not attempt to resolve this challenge here; indeed we think it is intrinsically unresolvable except up to a family of canonical isomorphisms, which need to be made explicit to really qualify as a legitimate answer (Mazur, 2008) – perhaps this will give some insight into 'natural' parameterizations of different learning algorithms.

One promising algorithmic standpoint, we believe, is the need for boosting algorithms for more complex / overparameterized architectures. Quite remarkably, in the 40+ references citing Long & Servedio (2010) whose context we compile in APP, only one alludes to the key sufficient condition to solve Long & Servedio (2010)'s problem: Schapire (2013) mentions the potential lack of "richness" of hypotheses available to the weak learner. This resonates with a comment in Long & Servedio (2010) whereby linear separators lack capacity to control confidences as would richer classes do, a model's flaw exploited by the negative results.

7. Conclusion

In this paper, we have used the theory of proper losses and a new boosting algorithm to show that the source of the negative result in the context of Long and Servedio's results is the model class. We believe our results demonstrate "unforeseeable" pitfalls of general parameterisations of a ML problem, including model class but also the learning algorithm and the loss function it optimizes.

Acknowledgments

Many thanks to Phil Long for stimulating discussions around the material presented and tipping us on the rotation argument for the proof of Lemma 3. Thanks are also due to the reviewers for very useful remarks that helped us to substantially improve the readability and presentation. This work was funded in part by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy — EXC number 2064/1 — Project number 390727645.

References

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

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

Amid, E., Warmuth, M.-K., Anil, R., and Koren, T. Robust bi-tempered logistic loss based on bregman divergences. In *NeurIPS*32*, pp. 14987–14996, 2019a.

Amid, E., Warmuth, M.-K., and Srinivasan, S. Two-temperature logistic regression based on the Tsallis divergence. In *22^{nd} AISTATS*, volume 89, pp. 2388–2396, 2019b.

Bao, H., Scott, C., and Sugiyama, M. Calibrated surrogate losses for adversarially robust classification. In *$33\ ^{rd}$ COLT*, volume 125, pp. 408–451, 2020.

Bartlett, P. and Traskin, M. Adaboost is consistent. In *NIPS*19*, 2006.

Ben-David, S., Loker, D., Srebro, N., and Sridharan, K. Minimizing the misclassification error rate using a surrogate convex loss. In *$29\ ^{th}$ ICML*, 2012.

Berger, J. O. *Statistical Decision Theory and Bayesian Analysis*. Springer, New York, 1985.

Blackmore, K. L., Williamson, R. C., and Mareels, I. M. Local minima and attractors at infinity for gradient descent learning algorithms. *J. of Mathematical Systems Estimation and Control*, 6:231–234, 1996.

Blanchard, G., Flaska, M., Handy, G., Pozzi, S., and Scott, C. Classification with asymmetric label noise: consistency and maximal denoising. *Electronic J. of Statistics*, 10:2780–2824, 2016.

Blass, A., Dershowitz, N., and Gurevich, Y. When are two algorithms the same? *Bulletin of Symbolic Logic*, 15(2): 145–168, 2009.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

Breiman, L., Freidman, J. H., Olshen, R. A., and Stone, C. J. *Classification and regression trees*. Wadsworth, 1984.

Bun, M., Carmosino, M.-L., and Sorrell, J. Efficient, noise-tolerant, and private learning via boosting. In *COLT'20*, Proceedings of Machine Learning Research, pp. 1031–1077. PMLR, 2020.

Charoenphakdee, N., Lee, J., and Sugiyama, M. On symmetric losses for learning from corrupted labels. In *36^{th} ICML*, volume 97, pp. 961–970, 2019.

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

Chen, S.-T., Balcan, M.-F., and Chau, D.-H. Communication efficient distributed agnostic boosting. In *19^{th} AISTATS*, volume 51, pp. 1299–1307, 2016.

Chen, Y., Xu, K., Zhou, P., Ban, X., and He, D. Improved cross entropy loss for noisy labels in vision leaf disease classification. *IET Image Processing*, 16(6):1511–1519, 2022.

Cheng, J., Liu, T., Ramamohanarao, K., and Tao, D. Learning with bounded instance and label-dependent label noise. In *37^{th} ICML*, volume 119, pp. 1789–1799, 2020.

Choi, H.-I. Lectures on machine learning, 2017. Seoul National University.

Cover, T.-M. and Hart, P. Nearest neighbor pattern classification. *IEEE Trans. IT*, 13:21–27, 1967.

Cranko, Z., Williamson, R. C., and Nock, R. Proper-composite loss functions in arbitrary dimensions. *arXiv e-prints*, pp. arXiv–1902, 2019.

Diakonikolas, I., Impagliazzo, R., Kane, D.-M., Lei, R., Sorrell, J., and Tzamos, C. Boosting in the presence of Massart noise. In *$34\ ^{th}$ COLT*, volume 134, pp. 1585–1644, 2021.

Ding, N. and Vishwanathan, S.-V.-N. t-logistic regression. In *NIPS*23*, pp. 514–522, 2010.

Freund, Y. and Mason, L. The alternating decision tree learning algorithm. In *Proc. of the $16\ ^{th}$ International Conference on Machine Learning*, pp. 124–133, 1999.

Friedman, J., Hastie, T., and Tibshirani, R. Additive Logistic Regression : a Statistical View of Boosting. *Ann. of Stat.*, 28:337–374, 2000.

Gao, W., Wang, L., Li, Y.-F., and Zhou, Z.-H. Risk minimization in the presence of label noise. In *AAAI'16*, pp. 1575–1581, 2016.

Geist, M. Soft-max boosting. *MLJ*, 100(2-3):305–332, 2015.

Ghosh, A., Kumar, H., and Sastry, P.-S. Robust loss functions under label noise for deep neural networks. In *AAAI'17*, pp. 1919–1925, 2017a.

Ghosh, A., Manwani, N., and Sastry, P.-S. On the robustness of decision tree learning under label noise. In *PAKDD'17*, pp. 685–697, 2017b.

Henry, C., Nock, R., and Nielsen, F. ℝeal boosting *a la Carte* with an application to boosting Oblique Decision Trees. In *Proc. of the 21 st International Joint Conference on Artificial Intelligence*, pp. 842–847, 2007.

Ju, X. *Boosting for regression problems with complex data*. PhD thesis, Uppsala University, 2022.

Kalai, A. and Kanade, V. Potential-based agnostic boosting. In *NIPS*22*, pp. 880–888, 2009.

Kalai, A. and Servedio, R.-A. Boosting in the presence of noise. In *STOC'03*, pp. 195–205. ACM, 2003.

Kearns, M. and Mansour, Y. On the boosting ability of top-down decision tree learning algorithms. In *Proc. of the 28 th ACM STOC*, pp. 459–468, 1996.

Kearns, M., Li, M., Pitt, L., and Valiant, L. On the learnability of boolean formulae. In *Proc. of the 19 th ACM Symposium on the Theory of Computing*, pp. 285–295, 1987.

Knuth, D.-E. Two notes on notation. *The American Mathematical Monthly*, 99(5):403–422, 1992.

Li, A.-H. and Bradic, J. Boosting in the presence of outliers: Adaptive classification with nonconvex loss functions. *Journal of the American Statistical Association*, 113(522): 660–674, 2018.

Lin, T.-Y., Goyal, P., Girshick, R.-B., He, K., and Dollár, P. Focal loss for dense object detection. In *ICCV'17*, pp. 2999–3007, 2017.

Liu, X., Petterson, J., and Caetano, T.-S. Learning as MAP inference in discrete graphical models. In *NIPS*25*, pp. 1979–1987, 2012.

Long, P.-M. and Servedio, R.-A. Adaptive martingale boosting. In *NIPS*21*, pp. 977–984, 2008a.

Long, P.-M. and Servedio, R.-A. Random classification noise defeats all convex potential boosters. In *25th ICML*, pp. 608–615, 2008b.

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

Long, P.-M. and Servedio, R.-A. Learning large-margin halfspaces with more malicious noise. In *NIPS*24*, pp. 91–99, 2011.

Luna, J.-M., Gennatas, E.-D., Ungar, L.-H., Eaton, E., Diffenderfer, E.-S., Jensen, S.-T., Simone II, C.-B., Friedman, J.-H., Solberg, T.-D., and Valdes, G. Building more accurate decision trees with the additive tree. *PNAS*, 116: 19887—-19893, 2019.

Mansour, Y. and McAllester, D. Boosting using branching programs. In *Proc. of the 13 th International Conference on Computational Learning Theory*, pp. 220–224, 2000.

Mazur, B. When is one thing equal to some other thing? In Gold, B. and Simons, R. A. (eds.), *Proof and other Dilemmas: Mathematics and Philosophy*, pp. 221–241. The Mathematical Association of America, 2008.

Menon, A.-K. The risk of trivial solutions in bipartite top ranking. *MLJ*, 108(4):627–658, 2019.

Menon, A.-K., van Rooyen, B., and Natarajan, N. Learning from binary labels with instance-dependent noise. *MLJ*, 107(8-10):1561–1595, 2018.

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

Mukherjee, I. and Schapire, R.-E. A theory of multiclass boosting. *JMLR*, 14(1):437–497, 2013.

Mussmann, S. and Liang, P. Uncertainty sampling is preconditioned stochastic gradient descent on zero-one loss. In *NeurIPS*31*, pp. 6955–6964, 2018.

Natarajan, N., Dhillon, I.-S., Ravikumar, P., and Tewari, A. Learning with noisy labels. In *NeurIPS*26*, pp. 1196–1204, 2013.

Nguyen, T. and Sanner, S. Algorithms for direct 0-1 loss optimization in binary classification. In *30th ICML*, volume 28, pp. 1085–1093, 2013.

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

Nock, R. and Nielsen, F. A ℝeal Generalization of discrete AdaBoost. *Artificial Intelligence*, 171:25–41, 2007.

Nock, R. and Nielsen, F. On the efficient minimization of classification-calibrated surrogates. In *NIPS*21*, pp. 1201–1208, 2008.

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

Nock, R., Bel Haj Ali, W., D'Ambrosio, R., Nielsen, F., and Barlaud, M. Gentle nearest neighbors boosting over proper scoring rules. *IEEE Trans.PAMI*, 37(1):80–93, 2015.

Noy, A. and Crammer, K. Robust forward algorithms via PAC-bayes and laplace distributions. In *17th AISTATS*, volume 33, pp. 678–686, 2014.

Olabiyi, O., Mueller, E.-T., and Larson, C. Stochastic gradient boosting for deep neural networks, 2021. US patent 10,990,878.

Pfetsch, N.-E. and Pokutta, S. IPBoost - non-convex boosting via integer programming. In *37th ICML*, volume 119, pp. 7663–7672, 2020.

Quinlan, J. R. *C4.5 : programs for machine learning*. Morgan Kaufmann, 1993.

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

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

Saffari, A., Godec, M., Pock, T., Leistner, C., and Bischof, H. Online multi-class LPBoost. In *Proc. of the 23rd IEEE CVPR*, pp. 3570–3577, 2010.

Savage, L.-J. Elicitation of personal probabilities and expectations. *J. of the Am. Stat. Assoc.*, pp. 783–801, 1971.

Schapire, R.-E. Explaining adaboost. In Schölkopf, B., Luo, Z., and Vovk, V. (eds.), *Empirical Inference - Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52, 2013.

Schapire, R. E. and Singer, Y. Improved boosting algorithms using confidence-rated predictions. In *9 th COLT*, pp. 80–91, 1998.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. Boosting the margin : a new explanation for the effectiveness of voting methods. *Annals of statistics*, 26:1651–1686, 1998.

Scott, C., Blanchard, G., and Handy, G. Classification with asymmetric label noise: Consistency and maximal denoising. In *26 th COLT*, volume 30, pp. 489–511, 2013.

Shuford, E., Albert, A., and Massengil, H.-E. Admissible probability measurement procedures. *Psychometrika*, pp. 125–145, 1966.

Sypherd, T., Nock, R., and Sankar, L. Being properly improper. In *39th ICML*, 2022.

Sypherd, T., Stromberg, N., Nock, R., Berisha, V., and Sankar, L. Smoothly giving up: Robustness for simple models. In *26nd AISTATS*, 2023.

Talwar, K. On the error resistance of Hinge-loss minimization. In *NeurIPS*33*, 2020.

Telgarsky, M. A primal-dual convergence analysis of boosting. *JMLR*, 13:561–606, 2012.

Telgarsky, M. Boosting with the logistic loss is consistent. In *26 th COLT*, pp. 911–965, 2013.

Tripathi, S. and Hemachandra, N. Cost sensitive learning in the presence of symmetric label noise. In *PAKDD'19*, volume 11439, pp. 15–28, 2019.

van Erven, T., Reid, M. D., and Williamson, R. C. Mixability is bayes risk curvature relative to log loss. *Journal of Machine Learning Research*, 13(52):1639–1663, 2012. URL http://jmlr.org/papers/v13/vanerven12a.html.

van Rooyen, B. and Menon, A.-K. An average classification algorithm. *CoRR*, abs/1506.01520, 2015.

van Rooyen, B., Menon, A., and Williamson, R.-C. Learning with symmetric label noise: The importance of being unhinged. In *NIPS*28*, 2015.

Walach, E. and Wolf, L. Learning to count with CNN boosting. In *ECCV'16*, volume 9906, pp. 660–676, 2016.

Wald, A. *Statistical Decision Functions*. John Wiley & Sons, New York, 1950.

Ward, F. Essays in international macroeconomics and financial crisis forecasting, 2017. PhD Dissertation, Friedrich-Wilhelms-Universität Bonn.

Williamson, R. C. and Helmke, U. Existence and uniqueness results for neural network approximations. *IEEE Transactions on Neural Networks*, 6(1):2–13, 1995.

Williamson, R. C., Vernet, E., and Reid, M. D. Composite multiclass losses. *Journal of Machine Learning Research*, 17(222):1–52, 2016. URL http://jmlr.org/papers/v17/14-294.html.

Xie, M. and Huang, S. CCMN: A general framework for learning with class-conditional multi-label noise. *IEEE T. PAMI*, 2022.

Zhu, Z., Liu, T., and Liu, Y. A second-order approach to learning with instance-dependent label noise. In *34th IEEE CVPR*, pp. 10113–10123, 2021.

Appendix

This is the Appendix to Paper ”Random Classification Noise does not defeat All Convex Potential Boosters Irrespective of Model Choice” (# 725) submitted to ICML’23. To differentiate with the numberings in the main file, the numbering of Theorems, etc. is letter-based (A, B, ...).

Table of contents

I. What the papers say

Disclaimer: these are cut-paste exerpts of many papers citing (Long & Servedio, 2010) (or the earlier ICML version)[§], with emphasis on (i) most visible venues, (ii) variability (not just papers but also patents, etc.). Apologies for the eventual loss of context due to cut-paste.

"*Long and Servedio [2010] further recently showed the failure of boosting algorithms based on convex potentials to tolerate random noise [...]*" — (Mohri et al., 2018)

"*For any boosting estimator with a convex loss, Long and Servedio (2010) constructed datasets that can be fitted well if they do not contain label noise, but cannot be learned to achieve more than 50% accuracy in the presence of any ratio of label noise.*" — (Ju, 2022)

"*Although desirable from an optimization standpoint, convex losses have been shown to be prone to outliers [25]*" — (Chen et al., 2022)

"*Servedio and Long [8] proved that, in general, any boosting algorithm that uses a convex potential function can be misled by random label noise*" — (Cheamanunkul et al., 2014)

"*In fact, Long and Servedio (2010) proved that any boosting algorithm utilizing a convex potential function (i.e. belonging to the Anyboost framework) can be deceived by random label noise. This assertion was further tested in a simulation setting by Freund et al. (2014), which finds merit to the use of non-convex potential boosters.*" — (Ju, 2022)

"*Long and Servedio [2010] prove that any method based on a convex potential is inherently ill-suited to random label noise*" — (Natarajan et al., 2013)

"*Robustness of risk minimization depends on the loss function. For binary classification, it is shown that 0–1 loss is robust to symmetric or uniform label noise while most of the standard convex loss functions are not robust (Long and Servedio 2010; Manwani and Sastry 2013)*" — (Ghosh et al., 2017a)

"*Furthermore, the assumption of sufficient richness among the weak hypotheses can also be problematic. Regarding this last point, Long and Servedio [18] presented an example of a learning problem which shows just how far off a universally consistent algorithm like AdaBoost can be from optimal when this assumption does not hold, even when the noise affecting the data is seemingly very mild.*" — (Schapire, 2013)

"[...] *it was shown that some boosting algorithms including AdaBoost are extremely sensitive to outliers [30].*" — (Walach & Wolf, 2016)

"*Long and Servedio [2010] showed that there exist linearly separable D where, when the learner observes some corruption $\tilde{D}$ with symmetric label noise of any nonzero rate, minimisation of any convex potential over a linear function class results in classification performance on D that is equivalent to random guessing. Ostensibly, this establishes that convex losses are not "SLN-robust" and motivates the use of non-convex losses [Stempfel and Ralaivola, 2009, Masnadi-Shirazi et al., 2010, Ding and Vishwanathan, 2010, Denchev et al., 2012, Manwani and Sastry, 2013].*" — (van Rooyen et al., 2015)

"*Long and Servedio (2008) have shown that boosting with convex potential functions (i.e., convex margin losses) is not robust to random class noise*" — (Reid & Williamson, 2010)

[§] Source: `https://scholar.google.com/scholar?oi=bibs&hl=en&cites=14973709218743030313&as_sdt=5`

"*Negative results for convex risk minimization in the presence of label noise have been established by Long and Servido (2010) and Manwani and Sastry (2011). These works demonstrate a lack of noise tolerance for boosting and empirical risk minimization based on convex losses, respectively, and suggest that any approach based on convex risk minimization will require modification of the loss, such that the risk minimizer is the optimal classifier with respect to the uncontaminated distributions*" — (Scott et al., 2013)

"*Boosting with convex loss functions is proven to be sensitive to outliers and label noise [19].*" — (Saffari et al., 2010)

"*While hinge loss used in SVMs (Cortes & Vapnik, 1995) and log loss used in logistic regression may be viewed as convex surrogates of the 0–1 loss that are computationally efficient to globally optimize (Bartlett et al., 2003), such convex surrogate losses are not robust to outliers (Wu & Liu, 2007; Long & Servedio, 2010; Ding & Vishwanathan, 2010)*" — (Nguyen & Sanner, 2013)

"[...] *For Theorem 29 to hold for AdaBoost, the richness assumption (72) is necessary, since there are examples due to Long and Servedio (2010) showing that the theorem may not hold when that assumption is violated*" — (Mukherjee & Schapire, 2013)

"[...] *Long & Servedio (2008) essentially establish that if one does not assume that margin error, ν, of the optimal linear classifier is small enough then any algorithm minimizing any convex loss φ (which they think of as a "potential") can be forced to suffer a large misclassification error.*" — (Ben-David et al., 2012)

"*The advantage of using a symmetric loss was investigated in the symmetric label noise scenario (Manwani & Sastry, 2013; Ghosh et al., 2015; Van Rooyen et al., 2015a). The results from Long & Servedio (2010) suggested that convex losses are non-robust in this scenario*" — (Charoenphakdee et al., 2019)

"*Overall, label noise is ubiquitous in real-world datasets and will undermine the performance of many machine learning models (Long & Servedio, 2010; Frenay & Verleysen, 2014).*" — (Cheng et al., 2020)

"*Although desirable from an optimization standpoint, convex losses have been shown to be prone to outliers [15]*" — (Amid et al., 2019a)

"*This is in contrast to recent work by Long and Servedio, showing that convex potential boosters cannot work in the presence of random classification noise [12].*" — (Kalai & Kanade, 2009)

"*The second strand has focussed on the design of surrogate losses robust to label noise. Long and Servedio [2008] showed that even under symmetric label noise, convex potential minimisation with such scorers will produce classifiers that are akin to random guessing.*" — (Menon et al., 2018)

"*Negative results for convex risk minimization in the presence of label noise have been established by Long and Servido [26] and Manwani and Sastry [27]. These works demonstrate a lack of noise tolerance for boosting and empirical risk minimization based on convex losses, and suggest that any approach based on convex risk minimization will require modification of the loss, [...]*" — (Blanchard et al., 2016)

"*For example, the random noise (Long and Servedio 2010) defeats all convex potential boosters [...]*" — (Gao et al., 2016)

"*Long and Servedio (2010) proved that any convex potential loss is not robust to uniform or symmetric label noise.*" —

(Ghosh et al., 2017b)

“*We previously [23] showed that any boosting algorithm that works by stagewise minimization of a convex “potential function” cannot tolerate random classification noise*” — (Long & Servedio, 2011)

“*However, the convex loss functions are shown to be prone to mistakes when outliers exist [25].*” — (Zhu et al., 2021)

“*[...] However, Long and Servedio (2010) pointed out that any boosting algorithm with convex loss functions is highly susceptible to a random label noise model.*” — (Li & Bradic, 2018)

“*One drawback of many standard boosting techniques, including AdaBoost, is that they can perform poorly when run on noisy data [FS96, MO97, Die00, LS08].*” — (Long & Servedio, 2008a)

“*Therefore, it has been shown that the convex functions are not robust to noise [13].*” — (Amid et al., 2019b)

“*This is because many boosting algorithms are vulnerable to noise (Dietterich, 2000; Long and Servedio, 2008).*” — (Chen et al., 2016)

“*Long and Servedio (2010) showed that there is no convex loss that is robust to label noises.*” — (Bao et al., 2020)

“*[...] However, as was recently shown by Long and Servedio [4], learning algorithms based on convex loss functions are not robust to noise*” — (Ding & Vishwanathan, 2010)

“*[...] For instance, several papers show how outliers and noise can cause linear classifiers learned on convex surrogate losses to suffer high zero-one loss (Nguyen and Sanner, 2013; Wu and Liu, 2007; Long and Servedio, 2010).*” — (Mussmann & Liang, 2018)

“*This is as opposed to most boosting algorithms that are highly susceptible to outliers [24].*” — (Noy & Crammer, 2014)

“*Moreover, in the case of boosting, it has been shown that convex boosters are necessarily sensitive to noise (Long and Servedio 2010 [...]*” — (Geist, 2015)

“*Ostensibly, this result establishes that convex losses are not robust to symmetric label noise, and motivates using non-convex losses [40, 31, 17, 15, 30].*” — (van Rooyen & Menon, 2015)

“*Interestingly, (Long and Servedio, 2010) established a lower bound against potential-based convex boosting techniques in the presence of RCN.*” — (Diakonikolas et al., 2021)

“*However, it was shown in (Long & Servedio, 2008; 2010) that any convex potential booster can be easily defeated by a very small amount of label noise*” — (Pfetsch & Pokutta, 2020)

“*A major roadblock one has to get around in label noise algorithms is the non-robustness of linear classifiers from convex potentials as given in [10].* ” — (Tripathi & Hemachandra, 2019)

“*Coming from the other end, the main argument for non-convexity is that a convex formulation very often fails to capture fundamental properties of a real problem (e.g. see [1, 2] for examples of some fundamental limitations of convex loss* *functions).*" — (Liu et al., 2012)

"*A theoretical analysis proposed in [21] proves that any method based on convex surrogate loss is inherently ill-suited to random label noise.*" — (Xie & Huang, 2022)

"*It has been observed that application of Friedman's stochastic gradient boosting to deep neural network training often led to training instability . See , e.g. Philip M. Long , et al , " Random Classification Noise Defeats All Convex Potential Boosters , " in Proceedings of the 25th International Conference on Machine Learning*" — (Olabiyi et al., 2021)

"*Long and Servedio [2010] showed that random classification noise already makes a large class of convex boosting-type algorithms fail.*" — (Talwar, 2020)

"*On the other hand, it has been known that boosting methods work rather poorly when the input data is noisy. In fact, Long and Servedio show that any convex potential booster suffer from the same problem [6].*" — (Choi, 2017)

"*Noise-resilience also appears to make CTEs outperform one of their most prominent competitors – boosting – whose out-of-sample AUC estimates appear to be held back by the level of noise in macroeconomic data (also see Long and Servedio, 2010)* " — (Ward, 2017)

"*The brittleness of convex surrogates is not unique to ranking, and plagues their use in standard binary classification as well (Long and Servedio 2010; Ben-David et al. 2012).* " — (Menon, 2019)

II. Supplementary material on proofs

II.1. Proof of Lemma 1

Strict convexity follows from its definition. Letting $\mathbb{I} \doteq \underline{L}'([0,1])$, we observe:

$$\phi_\ell(z) \doteq \sup_{u \in [0,1]} \{-zu + \underline{L}(u)\} = \left\{ \begin{array}{ccc} -z + \underline{L}(1) & \text{if} & z \leqslant \inf \mathbb{I} \\ -z \cdot \tilde{\eta}(-z) + \underline{L}(\tilde{\eta}(-z)) & \text{if} & z \in \mathbb{I} \\ \underline{L}(0) & \text{if} & z \geqslant \sup \mathbb{I} \end{array} \right. . \quad (27)$$

This directly establishes $\lim_{+\infty} \phi_\ell(z) = \underline{L}(0)$. Strict properness and differentiability ensure $\underline{L}'$ strictly decreasing. We also have

$$\phi_\ell{}'(z) = \left\{ \begin{array}{ccc} -1 & \text{if} & z \leqslant \inf \mathbb{I} \\ -(\underline{L}'^{-1})(-z) & \text{if} & z \in \mathbb{I} \\ 0 & \text{if} & z \geqslant \sup \mathbb{I} \end{array} \right. , \quad (28)$$

which shows $\phi_\ell{}'(z) \leqslant 0, \forall z \in \mathbb{R}$ and so ϕ_ℓ is decreasing. The definition of $\mathbb{I}$ ensures $\lim_{\inf \mathbb{I}} \phi_\ell{}'(z) = -1, \lim_{\sup \mathbb{I}} \phi_\ell{}'(z) = 0$ so ϕ_ℓ is differentiable. Convexity follows from the definition of ϕ_ℓ.

We now note the useful relationship coming from properness condition and (2) (main file):

$$\underline{L}'(u) \quad = \quad \ell_1(u) - \ell_{-1}(u). \quad (29)$$

This relationship brings two observations: first, the partial losses being differentiable, they are continuous and thus $\underline{L}'$ is continuous as well, which, together with $\mathrm{dom}(\underline{L}) = [0,1]$ brings the continuity of $\phi_\ell{}'$ and so ϕ_ℓ is C^1. The second is $\phi_\ell{}'(0) < 0$. We first show $0 \in \mathrm{int}\mathbb{I}$. Because of (29), if $0 \notin \mathrm{int}\mathbb{I}$, we either have $\ell_1(0) - \ell_{-1}(0) \leqslant 0$ or $\ell_1(1) - \ell_{-1}(1) \geqslant 0$. The integral representation of proper losses (Reid & Williamson, 2010)(Theorem 1) (Nock & Menon, 2020) (Appendix Section 9) yields that there exists a non-negative weight function $w : (0,1) \rightarrow \mathbb{R}_+$ such that

$$\ell_1(u) = \int_u^1 (1-t)w(t)\mathrm{d}t \quad ; \quad \ell_{-1}(u) = \int_0^u tw(t)\mathrm{d}t. \quad (30)$$

The condition $\ell_1(0) - \ell_{-1}(0) \leqslant 0$ imposes

$$\lim_{u\rightarrow 0} \int_u^1 (1-t)w(t)\mathrm{d}t = \ell_1(0) \quad \leqslant \quad \ell_{-1}(0) = \lim_{u \rightarrow 0}\int_0^u t w(t) \mathrm{d}t = 0, \tag{31}$$

which imposes $w(.) = 0$ almost everywhere and $\ell_1(u) = 0, \forall u$. Similarly, the condition $\ell_1(1) - \ell_{-1}(1) \geqslant 0$ imposes

$$\lim_{u\rightarrow 1} \int_u^1 (1-t)w(t)\mathrm{d}t = \ell_1(1) \quad \geqslant \quad \ell_{-1}(1) = \lim_{u \rightarrow 1}\int_0^u t w(t) \mathrm{d}t = 0, \tag{32}$$

which also imposes $w(.) = 0$ almost everywhere and $\ell_{-1}(u) = 0, \forall u$. $w(.) = 0$ almost everywhere implies $\ell_1(u) = \ell_{-1}(u) = \underline{L}(u) = 0, \forall u$, which is impossible given strict properness. So we get $0 \in \mathrm{int}\mathbb{I}$ and since $\underline{L}'$ is strictly decreasing, ${\underline{L}'}^{-1}(0) > 0$, implying

$$\phi_\ell{}'(0) = -(-\underline{L}')^{-1}(0) = -({\underline{L}'}^{-1})(0) < 0, \tag{33}$$

and ending the proof of Lemma 1.

II.2. Proof of Lemma 2

We first simplify the loss to a criterion equivalent to (Long & Servedio, 2010, eq. 5) (notations follow theirs):

$$\begin{aligned}
\tilde{\Phi}(h, \mathcal{S}) &= (N+1)\phi_\ell(-\alpha_1) - N\alpha_1 + 2(N+1)\phi_\ell(-\alpha_1\gamma + \alpha_2\gamma) - 2N(\alpha_1\gamma - \alpha_2\gamma) \\
&\quad +(N+1)\phi_\ell(-\alpha_1\gamma - K\alpha_2\gamma) - N(\alpha_1\gamma + K\alpha_2\gamma) \\
&= (N+1)\cdot(\phi_\ell(-\alpha_1) + 2\phi_\ell(-\alpha_1\gamma + \alpha_2\gamma) + \phi_\ell(-\alpha_1\gamma - K\alpha_2\gamma)) \\
&\quad -N\cdot((1+3\gamma)\alpha_1 + (K-2)\alpha_2\gamma)
\end{aligned}$$

We are interested in the properties of the linear classifier h minimizing that last expression. Denote for short:

$$\begin{aligned}
\varphi(z) &\doteq \phi_\ell{}'(z) + (1-\eta_{\scriptscriptstyle \mathrm{Y}}), \\
\tilde{P}_1(\alpha_1, \alpha_2) &\doteq \frac{1}{N+1}\cdot\frac{\partial\tilde{\Phi}(h,\mathcal{S})}{\partial\alpha_1}, \\
\tilde{P}_2(\alpha_1, \alpha_2) &\doteq \frac{1}{\gamma(N+1)}\cdot\frac{\partial\tilde{\Phi}(h,\mathcal{S})}{\partial\alpha_2}.
\end{aligned}$$

We note φ is increasing and satisfies $\lim_{-\infty}\varphi = -\eta_{\scriptscriptstyle \mathrm{Y}}, \lim_{+\infty}\varphi = 1 - \eta_{\scriptscriptstyle \mathrm{Y}}$. We get

$$\begin{aligned}
\tilde{P}_1(\alpha_1, \alpha_2) &= -\phi_\ell{}'(-\alpha_1) - \gamma\cdot\left\{2\phi_\ell{}'((\alpha_2 - \alpha_1)\gamma) + \phi_\ell{}'(-(\alpha_1 + K\alpha_2)\gamma)\right\} - \frac{N(1+3\gamma)}{N+1} \\
&= -\varphi(-\alpha_1) - 2\gamma\varphi((\alpha_2-\alpha_1)\gamma) - \gamma\varphi(-(\alpha_1+K\alpha_2)\gamma),
\end{aligned} \tag{34}$$

and

$$\begin{aligned}
\tilde{P}_2(\alpha_1, \alpha_2) &= 2\phi_\ell{}'((\alpha_2 - \alpha_1)\gamma) - K\phi_\ell{}'(-(\alpha_1 + K\alpha_2)\gamma) - \frac{N(K-2)}{N+1} \\
&= 2\varphi((\alpha_2-\alpha_1)\gamma) - K\varphi(-(\alpha_1+K\alpha_2)\gamma).
\end{aligned} \tag{35}$$

The system that zeroes both functions $\tilde{P}_1(\alpha_1, \alpha_2), \tilde{P}_2(\alpha_1, \alpha_2)$ is thus equivalent to having

$$\left\{\begin{array}{rrcl}
(i) & \varphi(-(\alpha_1 + K\alpha_2)\gamma) & = & \frac{2}{K}\cdot\varphi((\alpha_2-\alpha_1)\gamma) \\
(ii) & \frac{-\varphi(-\alpha_1)}{\gamma} & = & \frac{2(K+1)}{K}\cdot\varphi((\alpha_2-\alpha_1)\gamma)
\end{array}\right. . \tag{36}$$

We have two cases to solve this system, presented in Figure 2: a ”red” case, representing ”high” noise, for which $\varphi(0) \doteq u^* < 0$, and a ”blue” case, representing ”low” noise, for which $\varphi(0) \doteq u^* > 0$.

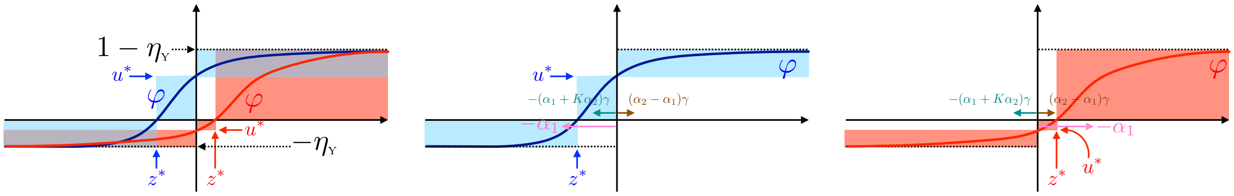


Figure 2. The two cases of our analysis for the proof of Lemma 2. In each case, we show the polarity of each of the arguments of system (36).

Red case: we solve (36) for the constraints $\alpha_2 > -\alpha_1, \alpha_1 < -z^*$; we pick $K = 2/(1+\varepsilon)$ for some small $0 < \varepsilon < 1$. Pick $\alpha_2 \doteq (1+\varepsilon)(1+B)\cdot -\alpha_1 > -\alpha_1$ for $B \geqslant 0$. The system (36) becomes:

$$\left\{\begin{array}{rrcl} (i) & \varphi((1+2B)\alpha_1\gamma) & = & (1+\varepsilon)\cdot\varphi((1+(1+\varepsilon)(1+B))\cdot-\alpha_1\gamma) \\ (ii) & \frac{-\varphi(-\alpha_1)}{\gamma} & = & (3+\varepsilon)\cdot\varphi((1+(1+\varepsilon)(1+B))\cdot-\alpha_1\gamma) \end{array}\right. . \quad (37)$$

Suppose

$$\alpha_1\gamma = \delta < 0,$$

for a small $|\delta|$. For any such constant $\delta > 0$, we see that

$$\frac{-\varphi(-\alpha_1)}{\gamma} = \frac{\alpha_1 \cdot -\varphi(-\alpha_1)}{\delta} \doteq V(\alpha_1)$$

and this time V satisfies $\lim_{-z^*} V = 0, \lim_{-\infty} V = -\infty$ and V is continuous because φ is, so for any value of the RHS in (ii) that keeps $(\alpha_2-\alpha_1)\gamma \in [0, z^*)$, the product $\alpha_1\gamma$ can be split in a couple (α_1, γ) for which the LHS in (ii) equates its RHS. We then just have to find a solution to (i) that meets our domain constraints. We observe that (i) becomes:

$$\varphi((1+2B)\delta) = (1+\varepsilon)\cdot\varphi((1+(1+\varepsilon)(1+B))\cdot-\delta), \quad (38)$$

whose quantities satisfy because of the monotonicity of φ,

$$\forall B \geqslant 0, \forall \delta \leqslant 0, \varphi((1+2B)\delta) \leqslant \varphi((2+B)\cdot-\delta), \quad (39)$$

which is (38) for $\varepsilon = 0$. We now show that there is a triple (ε, B, δ) with $\delta < 0, 0 < (\alpha_2-\alpha_1)\gamma = (1+(1+\varepsilon)(1+B))\cdot-\delta < z^*, B \geqslant 0, \varepsilon \geqslant 0$ which reverses the inequality, showing, by continuity of φ, a solution to (38). Fix small constants $\Delta_x, \Delta_y > 0$ such that we simultaneously have

$$\begin{aligned} (1+2B)\delta &= -\Delta_x, & (40)\\ \Delta_y &< \frac{-u^*}{3}, & (41)\\ \varphi(-\Delta_x) &\geqslant u^* - \Delta_y, & (42)\\ \varphi(\Delta_x) &\leqslant u^* + \Delta_y. & (43) \end{aligned}$$

The RHS of (38) becomes $(1+\varepsilon)\cdot\varphi\left(J(\varepsilon, B)\cdot\Delta_x\right)$ with

$$J(\varepsilon, B) \doteq \frac{1+(1+\varepsilon)(1+B)}{1+2B}. \quad (44)$$

$J(\varepsilon, B)$ satisfies the following property (P):

$$\forall 0 \leqslant \varepsilon < 1, \exists B > 0 : J(\varepsilon, B) = J(0,0) = 1.$$

Thanks to (P) and the continuity of J and φ, all we need to show for the existence of a solution to (i) is that there exists $\varepsilon < 1$ such that the central inequality underscored with "?" can hold,

$$(1+\varepsilon)\cdot\varphi(\underbrace{J(\varepsilon, B)}_{=J(0,0)=1}\cdot\Delta_x)\underbrace{\leqslant}_{(43)}(1+\varepsilon)\cdot(u^*+\Delta_y)\underbrace{\leqslant}_{?}u^*-\Delta_y\underbrace{\leqslant}_{(42)}\varphi(-\Delta_x). \quad (45)$$

(41) is equivalent to:

$$\frac{2\Delta_y}{-(u^* + \Delta_y)} \quad < \quad 1,$$

so any $2\Delta_y / -(u^* + \Delta_y) \leqslant \varepsilon < 1$ brings equivalently $(1+\varepsilon)\cdot(u^* + \Delta_y) \leqslant u^* - \Delta_y$, which is ”?” above.

Then, to solve (i), we first choose Δ_y satisfying (41), then pick Δ_x so that (42) and (43) are satisfied. This fixes the LHS of (38). From its minimal value $\varepsilon = 0$, we progressively increase ε while computing B so that (P) holds and getting δ from (40); while for $\varepsilon = 0$ (39) holds, we know that there is an $\varepsilon < 1$ such that (45) holds, the continuity of φ then showing there must be a value in the interval of εs for which equality, and thus (i), holds.

Then, from the value $\delta = \alpha_1\gamma$ obtained, we compute the couple $(\alpha_1, \gamma), \alpha_1 < 0, \gamma > 0$ such that (ii) holds, and then get α_2 from the identity $\alpha_2 \doteq (1+\varepsilon)(1+B)\cdot -\alpha_1 > -\alpha_1$.

Blue case: we solve (36) for the constraints $\alpha_2 > \alpha_1 > 0$; we pick $K = 2/(1-\varepsilon)$ for some small $0 < \varepsilon < 1$. Pick $\alpha_2 \doteq (1-\varepsilon)(1+B)\alpha_1 > \alpha_1$ for $\alpha_1 > 0, B > \varepsilon/(1-\varepsilon)$. The system (36) becomes:

$$\left\{ \begin{array}{lrcl} (i) & \varphi(-(1+2(1+B))\alpha_1\gamma) & = & (1-\varepsilon)\cdot\varphi((B(1-\varepsilon)-\varepsilon)\alpha_1\gamma) \\ (ii) & \frac{-\varphi(-\alpha_1)}{\gamma} & = & (3-\varepsilon)\cdot\varphi((B(1-\varepsilon)-\varepsilon)\alpha_1\gamma) \end{array} \right. . \tag{46}$$

Suppose

$$\alpha_1\gamma \quad = \quad \delta > 0,$$

a small constant. For any such constant $\delta > 0$, we see that

$$\frac{-\varphi(-\alpha_1)}{\gamma} \quad = \quad \frac{\alpha_1 \cdot -\varphi(-\alpha_1)}{\delta} \doteq V(\alpha_1)$$

and V satisfies $\lim_{-z^*} V = 0, \lim_{+\infty} V = +\infty$ and V is continuous because φ is, so for any value of the RHS in (ii), there exists a solution (α_1, γ) to (ii). We just need to figure out a solution to (i) for *some* $\delta > 0$ such that all our domain constraints are met. We observe (i) becomes

$$\varphi(-(1+2(1+B))\delta) \quad = \quad (1-\varepsilon)\cdot\varphi((B(1-\varepsilon)-\varepsilon)\delta) \doteq W(\varepsilon). \tag{47}$$

As $\delta \rightarrow 0^+$, the domain of solutions (ε, B) to (i) converges to $\{0\} \times \mathbb{R}$. $B > 0$ being fixed, we observe W is continuous and (noting the constraint $\varepsilon < B/(1+B)$)

$$W(0) = \varphi(B\delta) \geqslant u^* \quad ; \quad W\left(\frac{B}{1+B}\right) = \frac{u^*}{1+B}.$$

Remark that if we pick δ such that

$$\varphi(-(1+2(1+B))\delta) \quad \in \quad \left(\frac{u^*}{1+B}, u^*\right),$$

then there exists a solution $0 < \varepsilon < B/1 + B$ to (i) so we get $K > 2$ and ratio $\alpha_2/\alpha_1 > 1$. Then we solve (ii) for (α_1, γ) and get $\alpha_1 > 0$ and $\gamma > 0$.

Summary: accuracy of the optimal solution on $\mathcal{S}_{\text{clean}}$. In the **blue case**, we see that $\alpha_1 > 0, \alpha_2 > \alpha_1$, thus the accuracy is 50%. In the **red case** however, we see that, because $\alpha_1 < 0, \alpha_2 > -\alpha_1$, three examples of $\mathcal{S}_{\text{clean}}$ are badly classified and the accuracy thus falls to 25%.

II.3. Proof of Lemma 3

The trick we use is the same as in (Long & Servedio, 2010): we rotate the whole sample (which rotates accordingly the optimum and thus does not change its properties, loss-wise) in such a way that any booster would pick a ”wrong direction" to start, where the direction picked is the one with the largest edge (20). Let the rotation matrix of angle θ, with $c \doteq \cos\theta, s \doteq \sin\theta$,

$$\mathrm{R}_\theta \quad \doteq \quad \left[\begin{array}{cc} c & -s \\ s & c \end{array}\right]. \tag{48}$$

Denoting the rotated sample

$$\boxed{\mathcal{S}_{\mathrm{clean},\theta} \quad \doteq \quad \left\{ \left(\left[\begin{array}{c} c \\ s \end{array}\right], 1\right), \left(\left[\begin{array}{c} (c+s)\gamma \\ (s-c)\gamma \end{array}\right], 1\right), \left(\left[\begin{array}{c} (c+s)\gamma \\ (s-c)\gamma \end{array}\right], 1\right), \left(\left[\begin{array}{c} (c-Ks)\gamma \\ (s+Kc)\gamma \end{array}\right], 1\right) \right\},} \tag{49}$$

We note the sum of weights W, letting $L \doteq (-\underline{L}')^{-1}(0) \in (0,1)$:

$$W \quad = \quad 4(1-\eta_{\mathrm{Y}})(1-L) + 4\eta_{\mathrm{Y}} L = 4(1 - \eta_{\mathrm{Y}} - L + 2\eta_{\mathrm{Y}} L), \tag{50}$$

and we compute both edges (20) for both coordinates with the noisy dataset $\mathcal{S}_{\mathrm{noisy},\theta}$ by ranging through left to right of the examples' observations in $\mathcal{S}_{\mathrm{noisy},\theta}$:

$$\begin{aligned} \mathtt{e}_x &= \frac{\left\{\begin{array}{c} (1-\eta_{\mathrm{Y}})(1-L)c - \eta_{\mathrm{Y}} Lc + 2(1-\eta_{\mathrm{Y}})(1-L)(c+s)\gamma - 2\eta_{\mathrm{Y}} L(c+s)\gamma \\ +(1-\eta_{\mathrm{Y}})(1-L)(c-Ks)\gamma - \eta_{\mathrm{Y}} L(c-Ks)\gamma \end{array}\right.}{W} \\ &= \frac{(1-\eta_{\mathrm{Y}}-L)(1+3\gamma)(c - a\cdot s)}{W}, \end{aligned}$$

and

$$\mathtt{e}_y \quad = \quad \frac{(1-\eta_{\mathrm{Y}}-L)(1+3\gamma)(a\cdot c + s)}{4},$$

with

$$a \quad \doteq \quad \frac{(K-2)\gamma}{1+3\gamma}.$$

We also remind from Lemma 2 function $\varphi(z) \doteq \phi_\ell{}'(z) + (1-\eta_{\mathrm{Y}})$ and the proof of Lemma 1 that $\phi_\ell{}'(0) = -(-\underline{L}')^{-1}(0)$, so we remark the key identity:

$$1 - \eta_{\mathrm{Y}} - L \quad = \quad 1 - \eta_{\mathrm{Y}} - (-\underline{L}')^{-1}(0) = \phi_\ell{}'(0) + 1 - \eta_{\mathrm{Y}} = \varphi(0), \tag{51}$$

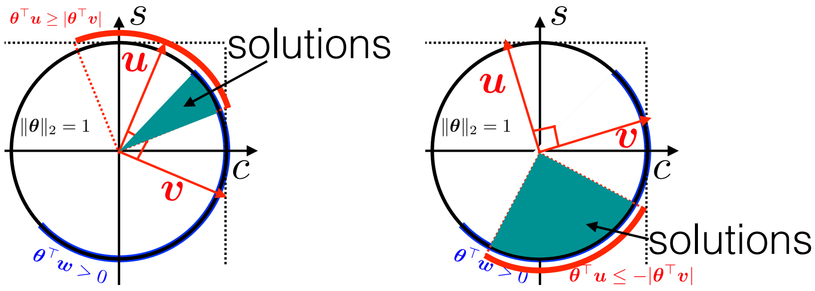
so the factor takes on two different signs in the **Blue** and **Red case** of Lemma 2. We thus distinguish two cases:

Blue case: we know (proof of Lemma 2) that $a > 0, \varphi(0) > 0$ and we want $\mathtt{e}_y > |\mathtt{e}_x|$ under the constraint that both y coordinates of the duplicated observations are negative: $(s-c)\gamma < 0$, so that the booster will pick the y coordinate with a positive leveraging coefficient and thus will badly classify the duplicated examples of $\mathcal{S}_{\mathrm{clean},\theta}$. We end up with the system (using $\gamma > 0$)

$$\left\{\begin{array}{rcl} a\cdot c + s & > & |c - a\cdot s|, \\ c - s & > & 0, \\ c^2 + s^2 & = & 1. \end{array}\right. , \tag{52}$$

which can be put in a vector form for graphical solving, letting $\boldsymbol{u} \doteq \left[\begin{array}{c} a \\ 1 \end{array}\right]$, $\boldsymbol{v} \doteq \left[\begin{array}{c} 1 \\ -a \end{array}\right]$ (note $\|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2$), $\boldsymbol{w} \doteq \left[\begin{array}{c} 1 \\ -1 \end{array}\right]$ and $\boldsymbol{\theta} \doteq \left[\begin{array}{c} c \\ s \end{array}\right]$, the vector of unknowns (with a slight abuse of notation), yielding

$$\left\{\begin{array}{rcl} \boldsymbol{\theta}^\top \boldsymbol{u} & > & |\boldsymbol{\theta}^\top \boldsymbol{v}|, \\ \boldsymbol{\theta}^\top \boldsymbol{w} & > & 0, \\ \|\boldsymbol{\theta}\|_2 & = & 1. \end{array}\right. \tag{53}$$



Blue case

Red case

Figure 3. Solutions that trick the booster in picking a first update that misclassifies the examples in $\mathcal{S}_{\text{clean},\theta}$ sharing the same observation (the "penalizers" in (Long & Servedio, 2010)).

Figure 3 (left) presents the computation of solutions.

Red case: we now have (proof of Lemma 2) that $a < 0, \varphi(0) < 0$. We now look to a solution to the following system:

$$\left\{\begin{array}{rcl} \boldsymbol{\theta}^\top \boldsymbol{u} & < & -|\boldsymbol{\theta}^\top \boldsymbol{v}|, \\ \boldsymbol{\theta}^\top \boldsymbol{w} & > & 0, \\ \|\boldsymbol{\theta}\|_2 & = & 1. \end{array}\right. \tag{54}$$

Figure 3 (right) presents the computation of solutions. The reason why it tricks again the booster in making at least 50% error on its first update is that $\boldsymbol{\theta}^\top \boldsymbol{u} < 0$ and thus $\mathrm{e}_y \propto \varphi(0) \cdot \boldsymbol{\theta}^\top \boldsymbol{u} > 0$ but also $|\mathrm{e}_y| > |\mathrm{e}_x| \propto |\varphi(0) \cdot \boldsymbol{\theta}^\top \boldsymbol{v}|$ and we check that because $\boldsymbol{\theta}^\top \boldsymbol{w} > 0$, the y coordinate of the two examples sharing the same observation (the "penalizers" in (Long & Servedio, 2010)) is negative and so they are both misclassified.

Remark 2. *The proof of Lemma 3 unveils what happens in the not-blue not-red case, when $\varphi(0) = 0$: in this case, the weak learner is totally "blind" as $\mathrm{e}_x = \mathrm{e}_y = 0$, so there is no possible update of the classifier as the weak learning assumption breaks down; the final classifier is thus the null vector = unbiased coin.*

II.4. A side negative result for ModaBoost with LS

We can show an impeding result for ModaBoost directly in the setting of Lemma 2: with the square loss (which allows to compute steps in closed form), ModaBoost hits a classifier as bad as the fair coin on (Long & Servedio, 2010)'s noise-free data in *at most 2 iterations only* for some values of the noise level and parameter γ (there is thus no need to use the rotation argument of Lemma 3 for the booster to "fail").

Lemma K. *Suppose* ModaBoost *is run with the square loss to learn a linear separator on $\mathcal{S}_{noisy}$ and* WL *returns a scaled vector from the canonical basis of $\mathbb{R}^2$. Then there exists $N > 1, 0 < \gamma < 1/6$ such that in at most* **two** *iterations,* ModaBoost *hits a linear separator with 50% accuracy on $\mathcal{S}_{clean}$.*

Proof: We recall the key parameters of the square loss for some constant $L > 0$ (unnormalized):

- partial losses: $\ell_1(u) = L(1-u)^2, \ell_{-1}(u) = Lu^2$, pointwise bayes risk: $\underline{L}(u) = Lu(1-u)$, convex surrogate:

$$\phi_\ell(z) = \left\{\begin{array}{ccc} -z & \text{if} & z < -L \\ \frac{L}{4} \cdot \left(1 - \frac{z}{L}\right)^2 & \text{if} & z \in [-L, L] \\ 0 & \text{if} & z > L \end{array}\right. .$$

- weight function $w(yH = z)$:

$$w(z) = \left\{\begin{array}{rcl} 1 & \text{if} & z < -L \\ \frac{1}{2} \cdot \left(1 - \frac{z}{L}\right) & \text{if} & z \in [-L, L] \\ 0 & \text{if} & z > L \end{array}\right. .$$

We recall and name the noisy examples, for $N > 1$:

- N copies of $(\gamma, 5\gamma)$ (call them A), N copies of $(1, 0)$ (call them B), $2N$ copies of $(\gamma, -\gamma)$ (call them C), all positive;
- 1 copy of $(\gamma, 5\gamma)$ (call them D), 1 copy of $(1, 0)$ (call them E), 2 copies of $(\gamma, -\gamma)$ (call them F), all negative;

We have two cases:
Case 1: suppose the first vector output by WL is proportional to $(1, 0)$.
Iteration 1: WL returns vector $h_1 = (U, 0)$ for $U > 0$, which labels correctly all observations in the noise-free case. The weights all equal $w(0) = 1/2$. The edge of h_1 is (we note $\max h_1 = U$):

$$
\begin{aligned}
\mathbb{e}_1(h_1) \doteq \left|\sum_{i\in[m]} \frac{w_{t,i}}{\sum_{j\in[m]} w_{t,j}} \cdot y_i^* \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j\in[m]} |h_t(\boldsymbol{x}_j)|}\right| &= \frac{\frac{N\gamma}{2} + \frac{N}{2} + \frac{2N\gamma}{2} - \frac{\gamma}{2} - \frac{1}{2} - \frac{2\gamma}{2}}{2(N+1)} \\
&= \frac{(1+3\gamma)}{4} \cdot \frac{N-1}{N+1}.
\end{aligned}
$$

The leveraging coefficient for h_1, α_1, is the solution of

$$
\begin{aligned}
&\underbrace{N\left(\frac{1}{2} - \frac{U\gamma\alpha_1}{L}\right)U\gamma}_{A} + \underbrace{N\left(\frac{1}{2} - \frac{U\alpha_1}{L}\right)U}_{B} + \underbrace{2N\left(\frac{1}{2} - \frac{U\gamma\alpha_1}{L}\right)U\gamma}_{C} \\
&\quad - \underbrace{\left(\frac{1}{2} + \frac{U\gamma\alpha_1}{L}\right)U\gamma}_{D} - \underbrace{\left(\frac{1}{2} + \frac{U\alpha_1}{L}\right)U}_{E} - \underbrace{2\left(\frac{1}{2} + \frac{U\gamma\alpha_1}{L}\right)U\gamma}_{F} = 0,
\end{aligned}
$$

giving

$$
\alpha_1 = \frac{2LU\left(\frac{N\gamma}{2} + \frac{N}{2} + \frac{2N\gamma}{2} - \frac{\gamma}{2} - \frac{1}{2} - \frac{2\gamma}{2}\right)}{U^2\left(\frac{(N+1)\gamma^2}{2} + \frac{N+1}{2} + \frac{2(N+1)\gamma^2}{2}\right)} = \frac{2L(N-1)(1+3\gamma)}{U(N+1)(1+3\gamma^2)}.
$$

We compute the new weight, with notation simplified to $w_2(.)$. For A, C, D, F, we remark

$$
\frac{|\alpha_1 h_1|}{L} = \frac{2\gamma(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} < \frac{1}{2},
$$

and so

$$
\begin{aligned}
w_2(A) = w_2(C) &= \frac{1}{2} - \frac{\gamma(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} \doteq \frac{1}{2} - \gamma k_2, \\
w_2(D) = w_2(F) &= \frac{1}{2} + \frac{\gamma(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} \doteq \frac{1}{2} + \gamma k_2,
\end{aligned}
$$

with

$$
k_2 \doteq \frac{(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)};
$$

while for B, E, we have

$$
\frac{|\alpha_1 h_1|}{L} = 2k_2.
$$

This, together with the fact that

$$
\frac{3+6\gamma+3\gamma^2}{1+6\gamma-3\gamma^2} \in [2,3], \forall \gamma \in [0,1], \tag{55}
$$

yields that if $N > 3$, then $w_2(B) = 0$, $w_2(E) = 1$ using the extreme expressions of the weight function. Let us assume $N \in \{2, 3\}$ to prevent this from happening (this simplifies derivations), so that

$$
\begin{aligned}
w_2(B) &= \frac{1}{2} - k_2, \\
w_2(E) &= \frac{1}{2} + k_2.
\end{aligned}
$$

Iteration 2: suppose WL returns vector $h_2 = (0, U)$ for $U > 0$. We note this time $\max h_2 = 5\gamma U$ and the edge is now

$$\begin{aligned}
\mathrm{e}_2(h_2) &= \left| U \cdot \frac{5N\gamma w_2(A) - 2N\gamma w_2(C) - 5N\gamma w_2(D) + 2N\gamma w_2(F)}{20\gamma N U} \right| \\
&= \left| \frac{-\frac{10N\gamma^2(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} + \frac{4N\gamma^2(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)}}{20\gamma N} \right| \\
&= \frac{3\gamma k_2}{10}.
\end{aligned}$$

The leveraging coefficient for h_2, α_2, is the solution of

$$\begin{aligned}
&\underbrace{N\left(\frac{1}{2} - \gamma k_2 - \frac{5U\gamma\alpha_2}{2L}\right)5U\gamma}_{A} + \underbrace{N\left(\frac{1}{2} - k_2\right)\cdot 0}_{B} - \underbrace{2N\left(\frac{1}{2} - \gamma k_2 + \frac{U\gamma\alpha_2}{2L}\right)\cdot(U\gamma)}_{C} \\
&- \underbrace{\left(\frac{1}{2} + \gamma k_2 + \frac{5U\gamma\alpha_2}{2L}\right)5U\gamma}_{D} - \underbrace{\left(\frac{1}{2} - k_2\right)\cdot 0}_{E} + \underbrace{2\left(\frac{1}{2} + \gamma k_2 - \frac{U\gamma\alpha_2}{2L}\right)\cdot(U\gamma)}_{F} = 0,
\end{aligned}$$

giving

$$\alpha_2 = \frac{3L(N-1)(1-2\gamma-3\gamma^2)}{27U(N+1)\gamma(1+3\gamma^2)}.$$

We check the new weights for A, C, D, F (others do not change). We remark for A, D:

$$\begin{aligned}
\frac{|\alpha_1 h_1 + \alpha_2 h_2|}{L} &= \frac{2\gamma(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} + \frac{15(N-1)(1-2\gamma-3\gamma^2)}{27(N+1)(1+3\gamma^2)} \\
&= \frac{N-1}{N+1} \cdot \frac{5+8\gamma+39\gamma^2}{9(1+3\gamma^2)} \\
&\leqslant \frac{5+8\gamma+39\gamma^2}{18(1+3\gamma^2)} \quad (N \in \{2,3\}) \\
&\leqslant \frac{1}{2} \quad (\gamma \leqslant 1/3),
\end{aligned}$$

while for C, F:

$$\begin{aligned}
\frac{|\alpha_1 h_1 + \alpha_2 h_2|}{L} &= \left| \frac{2\gamma(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} - \frac{(N-1)(1-2\gamma-3\gamma^2)}{9(N+1)(1+3\gamma^2)} \right| \\
&= \frac{N-1}{N+1} \cdot \frac{|-1+20\gamma+57\gamma^2|}{9(1+3\gamma^2)} \\
&\leqslant \frac{|-1+20\gamma+57\gamma^2|}{18(1+3\gamma^2)} \quad (N \in \{2,3\}) \\
&\leqslant \frac{1}{2} \quad (\gamma \leqslant 1/3),
\end{aligned}$$

so all the new weights are given not by the ”extreme” formulas of the weight function. We check the vector $\boldsymbol{\theta}_2$ learned after two iterations:

$$\begin{aligned}
\boldsymbol{\theta}_2 &= \left[\begin{array}{c} \frac{2L(N-1)(1+3\gamma)}{(N+1)(1+3\gamma^2)} \\ \frac{L(N-1)(1-2\gamma-3\gamma^2)}{9(N+1)\gamma(1+3\gamma^2)} \end{array} \right] \\
&= \frac{L(N-1)}{9(N+1)\gamma(1+3\gamma^2)} \cdot \left[\begin{array}{c} 18\gamma(1+3\gamma) \\ (1-2\gamma-3\gamma^2) \end{array} \right],
\end{aligned}$$

and we check that if $\gamma \leqslant 1/23$, then $\boldsymbol{\theta}_2$ misclassifies both positive examples with observation $(\gamma, -\gamma)$ (called the "penalizers" in (Long & Servedio, 2010)) in the noise-free dataset, thereby having 50% accuracy.

Case 2: suppose the first vector output by WL is proportional to $(0, 1)$.
Iteration 1: WL returns vector $h_1 = (0, U)$ for $U > 0$. We note $\max h_1 = 5\gamma U$ and the edge is

$$\begin{aligned}\mathrm{e}_1(h_1) &= U \cdot \frac{\frac{5\gamma UN}{2} + \frac{2\gamma UN}{2} - \frac{5\gamma U}{2} - \frac{2\gamma U}{2}}{20\gamma(N+1)U} \\ &= \frac{7(N-1)}{40(N+1)}.\end{aligned}$$

The leveraging coefficient for h_1, α_1, is the solution of

$$\begin{aligned}&\underbrace{N\left(\frac{1}{2} - \frac{5U\gamma\alpha_1}{2L}\right)5U\gamma}_{A} + \underbrace{0}_{B} - \underbrace{2N\left(\frac{1}{2} + \frac{U\gamma\alpha_1}{2L}\right)\cdot(U\gamma)}_{C} \\ &- \underbrace{\left(\frac{1}{2} + \frac{5U\gamma\alpha_2}{2L}\right)5U\gamma}_{D} + \underbrace{0}_{E} + \underbrace{2\left(\frac{1}{2} - \frac{U\gamma\alpha_2}{2L}\right)\cdot(U\gamma)}_{F} = 0,\end{aligned}$$

giving

$$\alpha_1 = \frac{L(N-1)}{9U(N+1)\gamma},$$

and we check the new weights for A, C, D, F (others do not change); we remark for A, D:

$$\frac{|\alpha_1 h_1|}{L} = \frac{5(N-1)}{9(N+1)} < \frac{1}{2},$$

while for C, F:

$$\frac{|\alpha_1 h_1|}{L} = \frac{(N-1)}{9(N+1)} < \frac{1}{2},$$

so after the first iteration, the vector $\boldsymbol{\theta}_1$ learned,

$$\boldsymbol{\theta}_1 = \left[\begin{array}{c} 0 \\ \frac{L(N-1)}{9(N+1)\gamma} \end{array}\right],$$

misclassifies again both positive examples with observation $(\gamma, -\gamma)$ (called the "penalizers" in (Long & Servedio, 2010)) in the noise-free dataset, thereby having 50% accuracy. □

Remark 3. *Remark that the edge substantially decreases between two iterations in Case 1 as:*

$$\frac{\mathrm{e}_2(h_2)}{\mathrm{e}_1(h_1)} = \frac{6\gamma}{5(1+3\gamma^2)}, \tag{56}$$

which indicates that if run for longer, the weak learning assumption will eventually end up being rapidly violated in MODABOOST, *preventing the application of Theorem 1.*

II.5. Proof of Lemma 4

The proof relies on five key observations (assuming wlog h_t does not zero over $\mathcal{S}_t$):

(1) The equation can be written with α_t explicit as $\sum_{i\in[m]_t}(y_i - y_i^*(-\underline{L}')^{-1}(H_{t-1}(\boldsymbol{x}_i) + \alpha_t \cdot h_t(\boldsymbol{x}_i))) \cdot y_i^* h_t(\boldsymbol{x}_i) = 0$, that is (since $(y_i^*)^2 = 1$),

$$\underbrace{\sum_{i\in[m]_t}(-\underline{L}')^{-1}(H_{t-1}(\boldsymbol{x}_i) + \alpha_t \cdot h_t(\boldsymbol{x}_i)) \cdot h_t(\boldsymbol{x}_i)}_{\doteq J_t(\alpha_t)} = \sum_{i\in[m]_t, y_i^*=1} h_t(\boldsymbol{x}_i). \tag{57}$$

(2) $\lim_{\alpha \nearrow} J_t(\alpha) = \sum_{i \in [m]_t, h_t(\boldsymbol{x}_i) > 0} h_t(\boldsymbol{x}_i) \doteq J_+$;

(3) $\lim_{\alpha \searrow} J_t(\alpha) = \sum_{i \in [m]_t, h_t(\boldsymbol{x}_i) < 0} h_t(\boldsymbol{x}_i) \doteq J_-$;

(4) $\sum_{i \in [m]_t, y^*_i = 1} h_t(\boldsymbol{x}_i) \in [J_-, J_+]$,

(5) $\mathrm{Im}(-\underline{L}')^{-1} = [0,1]$, since if there was an interval of non-zero measure missing then either $\underline{L}'$ would not be defined over such an interval (impossible by the differentiability assumption) or it would be constant (impossible given the strict properness condition). The same remarks for a single missing value;

which gives the statement of the Lemma. To get rid of infinite values, we remark that this happens only when $\sum_{i \in [m]_t, y^*_i = 1} h_t(\boldsymbol{x}_i) = \sum_{i \in [m]_t, h_t(\boldsymbol{x}_i) < 0} h_t(\boldsymbol{x}_i)$ ($-h_t$ makes perfect classification over $\mathcal{S}_t$) or $\sum_{i \in [m]_t, y^*_i = 1} h_t(\boldsymbol{x}_i) = \sum_{i \in [m]_t, h_t(\boldsymbol{x}_i) > 0} h_t(\boldsymbol{x}_i)$ (h_t makes perfect classification over $\mathcal{S}_t$), both of which are not possible.

II.6. Proof of Theorem 1

We proceed in several steps. The first shows a general guarantee on the decrease of the surrogate risk.

Lemma L. *Let D_F denote the Bregman divergence with (convex) generator F. The difference between two successive surrogate risks in* MODABOOST *satisfies:*

$$\Phi(H_t, \mathcal{S}) - \Phi(H_{t-1}, \mathcal{S}) \quad = \quad -p_t \cdot \mathbb{E}_{i \sim [m]_t} \left[\left\{ \begin{array}{ccc} D_{-\underline{L}}(w_{t+1,i} \| w_{t,i}) & \textit{if} & y_i = 0 \\ D_{-\underline{L}}(1 - w_{t+1,i} \| 1 - w_{t,i}) & \textit{if} & y_i = 1 \end{array} \right. \right], \tag{58}$$

where $[m]_t \subseteq [m]$ is the subset of indices of examples "fed" to the weak learner in $\mathcal{S}_t$ and $p_t \doteq \mathrm{Card}([m]_t)/m$.

Proof: We observe

$$\begin{aligned} & \Phi(H_t, \mathcal{S}) - \Phi(H_{t-1}, \mathcal{S}) \\ & \quad = \quad p_t \cdot \mathbb{E}_{i \sim [m]_t} \left[\phi_\ell(-H_t(\boldsymbol{x}_i)) - \phi_\ell(-H_{t-1}(\boldsymbol{x}_i)) - y_i (H_t - H_{t-1})(\boldsymbol{x}_i) \right]. \end{aligned} \tag{59}$$

For any example $(\boldsymbol{x}, y)$, if $y = 0$, by the definition of Bregman divergences and their dual symmetry property,

$$\begin{aligned} & \phi_\ell(-H_t(\boldsymbol{x})) - \phi_\ell(-H_{t-1}(\boldsymbol{x})) \\ & \quad = \quad (-\underline{L})^\star(H_t(\boldsymbol{x})) - (-\underline{L})^\star(H_{t-1}(\boldsymbol{x})) \\ & \quad = \quad -\left[(-\underline{L})^\star(H_{t-1}(\boldsymbol{x})) - (-\underline{L})^\star(H_t(\boldsymbol{x})) - (H_{t-1} - H_t)(\boldsymbol{x}) \cdot (-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \right] \\ & \qquad\quad -(H_{t-1} - H_t)(\boldsymbol{x}) \cdot (-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \\ & \quad = \quad -D_{(-\underline{L})^\star}(H_{t-1}(\boldsymbol{x}) \| H_t(\boldsymbol{x})) - (H_{t-1} - H_t)(\boldsymbol{x}) \cdot (-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \\ & \quad = \quad -D_{-\underline{L}}((-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \| (-\underline{L}')^{-1}(H_{t-1}(\boldsymbol{x}))) - (H_{t-1} - H_t)(\boldsymbol{x}) \cdot (-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \\ & \quad = \quad -D_{-\underline{L}}(w((\boldsymbol{x}, y), H_t) \| w((\boldsymbol{x}, y), H_{t-1})) + \alpha_t \cdot w((\boldsymbol{x}, y), H_t) h_t(\boldsymbol{x}) \\ & \quad = \quad -D_{-\underline{L}}(w((\boldsymbol{x}, y), H_t) \| w((\boldsymbol{x}, y), H_{t-1})) - \alpha_t \cdot w((\boldsymbol{x}, y), H_t) \cdot y^* h_t(\boldsymbol{x}), \end{aligned}$$

If $y = 1$ (we do not replace y by 1 to mark its locations),

$$\begin{aligned} & \phi_\ell(-H_t(\boldsymbol{x})) - \phi_\ell(-H_{t-1}(\boldsymbol{x})) - y(H_t - H_{t-1})(\boldsymbol{x}) \\ & \quad = \quad (-\underline{L})^\star(H_t(\boldsymbol{x})) - (-\underline{L})^\star(H_{t-1}(\boldsymbol{x})) - y(H_t - H_{t-1})(\boldsymbol{x}) \\ & \quad = \quad -D_{-\underline{L}}((-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \| (-\underline{L}')^{-1}(H_{t-1}(\boldsymbol{x}))) - (H_{t-1} - H_t)(\boldsymbol{x}) \cdot (-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \\ & \qquad\quad -y(H_t - H_{t-1})(\boldsymbol{x}) \\ & \quad = \quad -D_{-\underline{L}}((-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \| (-\underline{L}')^{-1}(H_{t-1}(\boldsymbol{x}))) + ((-\underline{L}')^{-1}(H_t(\boldsymbol{x})) - y) \cdot \alpha_t h_t(\boldsymbol{x}) \\ & \quad = \quad -D_{-\underline{L}}((-\underline{L}')^{-1}(H_t(\boldsymbol{x})) \| (-\underline{L}')^{-1}(H_{t-1}(\boldsymbol{x}))) - (y - (-\underline{L}')^{-1}(H_t(\boldsymbol{x}))) \cdot \alpha_t \cdot y^* h_t(\boldsymbol{x}) \\ & \quad = \quad -D_{-\underline{L}}(y - w((\boldsymbol{x}, y), H_t) \| y - w((\boldsymbol{x}, y), H_{t-1})) - \alpha_t \cdot w((\boldsymbol{x}, y), H_t) \cdot y^* h_t(\boldsymbol{x}), \end{aligned}$$

and thus, we get for ModaBoost the relationship between successive surrogate risks,

$$\begin{aligned}
&\Phi(H_t, \mathcal{S}) - \Phi(H_{t-1}, \mathcal{S}) \\
&= -p_t \cdot \mathbb{E}_{i\sim[m]_t}\left[\left\{\begin{array}{ccc} D_{-\underline{L}}(w_{t+1,i}\|w_{t,i}) & \text{if} & y_i = 0 \\ D_{-\underline{L}}(1-w_{t+1,i}\|1-w_{t,i}) & \text{if} & y_i = 1\end{array}\right.\right] \\
&\quad -\alpha_t \cdot \sum_{i\in[m]_t} w_{i,t+1}\cdot y_i^* h_t(\boldsymbol{x}_i) \\
&= -p_t \cdot \mathbb{E}_{i\sim[m]_t}\left[\left\{\begin{array}{ccc} D_{-\underline{L}}(w_{t+1,i}\|w_{t,i}) & \text{if} & y_i = 0 \\ D_{-\underline{L}}(1-w_{t+1,i}\|1-w_{t,i}) & \text{if} & y_i = 1\end{array}\right.\right], \qquad (60)
\end{aligned}$$

by (15). □

The following Theorem established a general boosting-compliant convergence bound, the central piece of our proof.

Theorem B. *Define the expected and normalized weights at iteration t:*

$$\overline{w^*}_t \doteq \frac{\sum_{i\in[m]_t} w_{t,i}}{\mathrm{Card}([m]_t)} \quad ; \quad w^{\textit{norm}}_{t,i} \doteq \frac{w_{t,i}}{\sum_{j\in[m]_t} w_{t,j}}, \qquad (61)$$

*and the following two assumptions (**LOSS0**, **WLA**):*

*(**LOSS0**) The loss chosen ℓ is strictly proper, differentiable and satisfies $\inf\{\ell'_{-1} - \ell'_1\} \geqslant \kappa$ for some $\kappa > 0$;*
*(**WLA**) There exists a constant $\gamma_{\mathrm{WL}} > 0$ such that at each iteration $t \in [T]$, the weak hypothesis h_t returned by WL satisfies*

$$\left|\sum_{i\in[m]_t} w^{\textit{norm}}_{t,i}\cdot y_i^* \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{i\in[m]_t}|h_t(\boldsymbol{x}_i)|}\right| \geqslant \gamma_{\mathrm{WL}}. \qquad (62)$$

*Then under **LOSS0** and **WLA** the following holds:*

$$\forall \Phi \in \mathbb{R}, \left(\sum_{t=1}^T p_t \overline{w^*}_t^2 \geqslant \frac{2(\Phi(H_0,\mathcal{S}) - \Phi)}{\kappa\gamma^2}\right) \Rightarrow (\Phi(H_T, \mathcal{S}) \leqslant \Phi). \qquad (63)$$

Proof: Assuming second-order differentiability, we have the classical Taylor approximation of Bregman divergences (Nock & Menon, 2020, Appendix II): for $t \in [T], i \in [m]_t, \exists u_{t,i}, v_{t,i} \in [0,1]$ such that:

$$\begin{aligned}
D_{-\underline{L}}(w_{t+1,i}\|w_{t,i}) &= \frac{(-\underline{L})''(u_{t,i})(w_{t+1,i}-w_{t,i})^2}{2}, \\
D_{-\underline{L}}(1-w_{t+1,i}\|1-w_{t,i}) &= \frac{(-\underline{L})''(v_{t,i})(w_{t+1,i}-w_{t,i})^2}{2},
\end{aligned}$$

It follows from Sypherd et al. (2022, Lemma 12) and assumption **LOSS0** that ℓ being strictly proper, we have $(-\underline{L})'' = \ell'_{-1} - \ell'_1 \geqslant \inf\{\ell'_{-1} - \ell'_1\} \geqslant \kappa$, so we get

$$p_t \cdot \mathbb{E}_{i\sim[m]_t}\left[\left\{\begin{array}{ccc} D_{-\underline{L}}(w_{t+1,i}\|w_{t,i}) & \text{if} & y_i = 0 \\ D_{-\underline{L}}(1-w_{t+1,i}\|1-w_{t,i}) & \text{if} & y_i = 1\end{array}\right.\right] \geqslant \frac{p_t\kappa}{2}\cdot \mathbb{E}_{i\sim[m]_t}\left[(w_{t+1,i}-w_{t,i})^2\right],$$

We remark

$$\begin{aligned}
\left(\sum_{i\in[m]_t} w_{t,i}\cdot y_i^* h_t(\boldsymbol{x}_i)\right)^2 &= \left(\sum_{i\in[m]_t} w_{t,i}\cdot y_i^* h_t(\boldsymbol{x}_i) - \sum_{i\in[m]_t} w_{t+1,i}\cdot y_i^* h_t(\boldsymbol{x}_i)\right)^2 \qquad (64)\\
&= \left(\sum_{i\in[m]_t} (w_{t,i} - w_{t+1,i})\cdot y_i^* h_t(\boldsymbol{x}_i)\right)^2 \qquad (65)\\
&\leqslant \sum_{i\in[m]_t} (w_{t,i} - w_{t+1,i})^2 \cdot \sum_{i\in[m]_t} h_t^2(\boldsymbol{x}_i) \qquad (66)\\
&\leqslant \mathrm{Card}([m]_t)^2 M_t^2 \cdot \mathbb{E}_{i\sim[m]_t}\left[(w_{t+1,i}-w_{t,i})^2\right], \qquad (67)
\end{aligned}$$

where we have used (15), Cauchy-Schwartz, the assumption that the distribution is uniform and let $M_t \doteq \max_{i \in [m]_t} |h(\boldsymbol{x}_i)|$. Thus,

$$\begin{aligned} & p_t \cdot \mathbb{E}_{i \sim [m]_t} \left[\left\{ \begin{array}{ccc} D_{-\underline{L}}(w_{t+1,i} \| w_{t,i}) & \text{if} & y_i = 0 \\ D_{-\underline{L}}(1 - w_{t+1,i} \| 1 - w_{t,i}) & \text{if} & y_i = 1 \end{array} \right. \right] \\ \geqslant & \frac{p_t \kappa}{2} \cdot \frac{\left(\sum_{i \in [m]_t} w_{t,i} \cdot y^*_i \cdot \frac{h_t(\boldsymbol{x}_i)}{M} \right)^2}{\mathrm{Card}([m]_t)^2} \\ & = \frac{p_t \kappa}{2} \cdot \underbrace{\left(\frac{\sum_{i \in [m]_t} w_{t,i}}{\mathrm{Card}([m]_t)} \right)^2}_{\doteq \overline{w*}_t^2} \cdot \underbrace{\left(\sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y^*_i \cdot \frac{h_t(\boldsymbol{x}_i)}{M_t} \right)^2}_{\geqslant \gamma^2 \text{ from } \textbf{WLA}} . \end{aligned} \tag{68}$$

Assumptions **LOSS0** and **WLA** thus imply the guaranteed decrease between two successive risks

$$\Phi(H_t, \mathcal{S}) \quad \leqslant \quad \Phi(H_{t-1}, \mathcal{S}) - \frac{p_t \kappa \gamma^2 \overline{w*}_t^2}{2}, \tag{69}$$

and we have, after collapsing summing inequalities for $t = 1, 2, ..., T$, the guarantee that as long as the WLA holds,

$$\forall \Phi \in \mathbb{R}, \left(\sum_{t=1}^{T} p_t \overline{w*}_t^2 \geqslant \frac{2(\Phi(H_0, \mathcal{S}) - \Phi)}{\kappa \gamma^2} \right) \quad \Rightarrow \quad (\Phi(H_T, \mathcal{S}) \leqslant \Phi) \, , \tag{70}$$

which is the statement of Theorem B. □

Because it involves $\overline{w*}_t$, this bound is not fully readable, but there is a simple way to remove its dependence as $\overline{w*}_t$ is also linked to the quality of the classifier H: roughly speaking, the smaller it is, the worse is the dependence in (70) *but* the better is H since weights tends to decrease as H gives the right class with increased confidence ($|H|$). The trick is thus to find a value of $\overline{w*}_t$ below which H is ”satisfying” (boosting-wise) and then plug this bound in (70), which then gives a number of iterations after which H becomes satisfying anyway.

We need the following definition.

Definition II.1. *(after (Bun et al., 2020)) Weights at iteration t are called ζ-dense if $\overline{w}_t \geqslant \zeta$, where*

$$\overline{w}_t \quad \doteq \quad \frac{\sum_{i \in [m]} w_{t,i}}{m}.$$

Notice that the expected weight here, $\overline{w}_t$, spans all the training sample, which is **not** the case for $\overline{w*}_t$ (which relies on the examples ”fed” to the weak learner). We make precise the notion of being ”satisfying” when weights are ”small”.

Lemma M. *For any $t \geqslant 1, \zeta \in [0, 1]$, suppose the weights at iteration $t + 1$ are not ζ-dense. Then $\forall \theta \in \mathbb{R}$,*

$$\mathbb{P}_{i \sim [m]}[y^*_i H_t(\boldsymbol{x}_i) \leqslant \theta] \quad < \quad \frac{\zeta}{\underline{w}(\theta)}, \tag{71}$$

where we let $\underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$.

Proof: We denote $[m]_+$ the set of indices whose examples have positive class. Let $z^+_1, z^+_2, ..., z^+_{\mathrm{Card}([m]_+)}$ some associated reals and $w_+(z) \doteq 1 - (-\underline{L}')^{-1}(z)$ the positive examples' weight function. Being non-increasing and with range in $[0, 1]$, we have $\forall \theta \in \mathbb{R}$

$$\mathbb{E}_{i \sim [m]_+}[w_+(z^+_i)] \quad \geqslant \quad \mathbb{P}_{[m]_+}[z^+_i \leqslant \theta] \cdot w_+(\theta) + \mathbb{P}_{[m]_+}[z^+_i > \theta] \cdot \inf w_+ \tag{72}$$

$$= \mathbb{P}_{[m]_+}[z^+_i \leqslant \theta] \cdot w_+(\theta), \tag{73}$$

so using this with $z^+_i = H_t(\boldsymbol{x}_i) = y^*_i H_t(\boldsymbol{x}_i)$ yields $w_+(z^+_i) = w_{t+1,i}$ and $\mathbb{E}_{i \sim [m]_+}[w_{t+1,i}] \geqslant \mathbb{P}_{[m]_+}[y^*_i H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot w_+(\theta)$.

Similarly, let $[m]_-$ the set of negative indices for iteration t. Let $z_1^-, z_2^-, ..., z_{\mathrm{Card}([m]_-)}^-$ some associated reals and $w_-(z) \doteq (-\underline{L}')^{-1}(z)$ the negative examples' weight function. Being non-decreasing and with range in $[0, 1]$, we have $\forall \theta \in \mathbb{R}$

$$\begin{aligned} \mathbb{E}_{i\sim[m]_-}[w_-(z_i^-)] &\geqslant \mathbb{P}_{[m]_-}[z_i^- \geqslant -\theta] \cdot w_-(-\theta) - \mathbb{P}_{[m]_-}[z_i^- < -\theta] \cdot \inf w_- \quad (74)\\ &= \mathbb{P}_{[m]_-}[-z_i^- \leqslant \theta] \cdot w_-(-\theta), \quad (75)\end{aligned}$$

so using this with $z_i^- = H_t(\boldsymbol{x}_i) = -y_i^* H_t(\boldsymbol{x}_i)$ yields $w_+(z_i^-) = w_{t+1,i}$ and $\mathbb{E}_{i\sim[m]_-}[w_{t+1,i}] \geqslant \mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot w_-(-\theta)$.

Denote $c(i) \in \{+, -\}$ the label of index i in $[m]$ and p^+, p^- the proportion of positive and negative examples in $[m]$. With a slight abuse of notation in indices, we have $p^+\mathbb{E}_{i\sim[m]_+}[w_+(z_i^+)] + p^-\mathbb{E}_{i\sim[m]_-}[w_-(z_i^-)] = \mathbb{E}_{i\sim[m]}[w_{c(i)}(z_i^{c(i)})] = \overline{w}_{t+1}$ where the last identity holds for the choices of the $z_i^\bullet$s made above. We thus have the lower-bound on $\overline{w}_{t+1}$:

$$\begin{aligned} \overline{w}_{t+1} &\geqslant p^+\mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot w_+(\theta) + p^-\mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot w_-(-\theta) \quad (76)\\ &\geqslant (p^+\mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] + p^-\mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \min\{w_+(\theta), w_-(-\theta)\} \quad (77)\\ &= \mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}, \quad (78)\end{aligned}$$

so for any $\varepsilon \in [0, 1]$,

$$(\mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \geqslant \varepsilon) \quad \Rightarrow \quad (\overline{w}_{t+1} \geqslant \varepsilon \cdot \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}), \quad (79)$$

so if $\overline{w}_{t+1} < \zeta$, then by contraposition

$$\begin{aligned} \mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] &< \frac{\zeta}{\min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}} \quad (80)\\ &= \frac{\zeta}{\underline{w}(\theta)}, \quad (81)\end{aligned}$$

as claimed. □

Fix from now on

$$\zeta \quad \doteq \quad \varepsilon \cdot \underline{w}(\theta). \quad (82)$$

We have two cases to conclude on our main result.

Case 1: sometimes during the induction, the weights for the "next iteration" ($t + 1$) fail to be ζ-dense. By Lemma M, $\mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon$ and we are done.

Case 2: weights are always ζ-dense:

$$\overline{w}_t^2 \quad \geqslant \quad \varepsilon^2 \cdot \underline{w}(\theta)^2, \forall t = 1, 2, ...$$

Recall the key statement of Theorem B:

$$\forall \Phi \in \mathbb{R}, \left(\sum_{t=1}^{T} p_t \overline{w^*}_t^2 \geqslant \frac{2(\Phi(H_0, \mathcal{S}) - \Phi)}{\kappa\gamma^2}\right) \quad \Rightarrow \quad (\Phi(H_T, \mathcal{S}) \leqslant \Phi) \, .$$

Provided we can assume a lowerbound of the form[¶]

$$p_t \overline{w^*}_t^2 \quad \geqslant \quad u_t \overline{w}_t^2, \forall t = 1, 2, ... \quad (83)$$

where $u_t > 0$ is not too small, we see that ζ-denseness thus enforces a decrease of $\Phi(H, \mathcal{S})$ via Theorem B, and we only need a link between this and $\mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]$, reminding

$$\Phi(H, \mathcal{S}) \doteq \mathbb{E}_{i\sim[m]}\left[\phi_\ell(-H(\boldsymbol{x}_i)) - y_i H(\boldsymbol{x}_i)\right] \quad , \quad \phi_\ell(z) \doteq (-\underline{L})^\star(-z).$$

[¶] Note that this is equivalent to u_t compliance in Definition 5.1.

Lemma N. *Let* $\underline{\phi_\ell}(z) \doteq \min\{\phi_\ell(z), \phi_\ell(-z) - z\}$*. For any* $t \geqslant 1$ *and any* $\theta \in \mathbb{R}$ *such that:*

$$\underline{\phi_\ell}(\theta) \quad > \quad \min_{i \in [m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i)),$$

we have for any $u \in [0, 1]$*,*

$$(\mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] > u) \quad \Rightarrow \quad \left(\Phi(H_t, \mathcal{S}_t) \geqslant u\underline{\phi_\ell}(\theta) + (1-u) \min_{i\in[m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i))\right).$$

Proof: We reuse some notations from Lemma M. We first note

$$\begin{aligned}\mathbb{E}_{i\sim[m]}\left[\phi_\ell(-H_t(\boldsymbol{x}_i)) - y_i H_t(\boldsymbol{x}_i)\right] \quad &= \quad p^-\mathbb{E}_{i\sim[m]_-}\left[\phi_\ell(-H_t(\boldsymbol{x}_i))\right] \\ &\quad +p^+\mathbb{E}_{i\sim[m]_+}\left[\phi_\ell(-H_t(\boldsymbol{x}_i)) - H_t(\boldsymbol{x}_i)\right].\end{aligned} \tag{84}$$

Let us analyse the term for negative examples and have $z_i^- \leftarrow H(\boldsymbol{x}_i)$ for short. Because $\phi_\ell(-z)$ is non-decreasing, for any $\theta \in \mathbb{R}$,

$$\begin{aligned}&\mathbb{E}_{[m]_-}[\phi_\ell(-z_i^-)] \\ &\quad\geqslant \quad \mathbb{P}_{[m]_-}[z_i^- \geqslant -\theta] \cdot \phi_\ell(\theta) + (1 - \mathbb{P}_{[m]_-}[z_i^- \geqslant -\theta]) \cdot \min_{i\in[m]_-} \phi_\ell(-z_i^-) \\ &\qquad = \mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot \phi_\ell(\theta) + (1 - \mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \min_{i\in[m]_-} \phi_\ell(y_i^* H_t(\boldsymbol{x}_i)) \\ &\quad\geqslant \quad \mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot \underline{\phi_\ell}(\theta) + (1 - \mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \min_{i\in[m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i)).\end{aligned}$$

Similarly for positive examples, letting $z_i^+ \leftarrow H(\boldsymbol{x}_i)$ for short, we remark that $\phi_\ell(-z) - z$ is non-increasing and so for any $\theta \in \mathbb{R}$,

$$\begin{aligned}&\mathbb{E}_{[m]_+}[\phi_\ell(-z_i^+) - z_i^+] \\ &\quad\geqslant \quad \mathbb{P}_{[m]_+}[z_i^+ \leqslant \theta] \cdot (\phi_\ell(-\theta) - \theta) + (1 - \mathbb{P}_{[m]_+}[z_i^+ \leqslant \theta]) \cdot \min_{i\in[m]_+} \phi_\ell(-z_i^+) - z_i^+ \\ &\qquad = \mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot (\phi_\ell(-\theta) - \theta) \\ &\qquad +(1 - \mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \min_{i\in[m]_+} \phi_\ell(-y_i^* H_t(\boldsymbol{x}_i)) - y_i^* H_t(\boldsymbol{x}_i) \\ &\quad\geqslant \quad \mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot \underline{\phi_\ell}(\theta) + (1 - \mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \min_{i\in[m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i)).\end{aligned}$$

Hence we get from (84) that for any $\theta \in \mathbb{R}$,

$$\begin{aligned}&\mathbb{E}_{i\sim[m]}\left[\phi_\ell(-H_t(\boldsymbol{x}_i)) - y_i H_t(\boldsymbol{x}_i)\right] \\ &\quad = \quad p^-\mathbb{E}_{i\sim[m]_-}\left[\phi_\ell(-H_t(\boldsymbol{x}_i))\right] + p^+\mathbb{E}_{i\sim[m]_+}\left[\phi_\ell(-H_t(\boldsymbol{x}_i)) - H_t(\boldsymbol{x}_i)\right] \\ &\quad \geqslant \quad (p^-\mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] + p^+\mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \underline{\phi_\ell}(\theta) \\ &\qquad\quad +((p^- + p^+) - (p^-\mathbb{P}_{[m]_-}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] + p^+\mathbb{P}_{[m]_+}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta])) \cdot \min_{i\in[m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i)) \\ &\qquad\quad = \mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] \cdot \underline{\phi_\ell}(\theta) + (1 - \mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta]) \cdot \min_{i\in[m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i)).\end{aligned}$$

We get that for any $\theta \in \mathbb{R}$ such that:

$$\underline{\phi_\ell}(\theta) \quad > \quad \min_{i \in [m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i)),$$

we have for any $u \in [0, 1]$,

$$\begin{aligned}&(\mathbb{P}_{i\sim[m]}[y_i^* H_t(\boldsymbol{x}_i) \leqslant \theta] > u) \\ &\qquad \Rightarrow \quad \left(\underbrace{\mathbb{E}_{i\sim[m]}\left[\phi_\ell(-H_t(\boldsymbol{x}_i)) - y_i H_t(\boldsymbol{x}_i)\right]}_{\doteq \Phi(H_t, \mathcal{S}_t)} \geqslant u\underline{\phi_\ell}(\theta) + (1-u) \min_{i\in[m]} \underline{\phi_\ell}(y_i^* H_t(\boldsymbol{x}_i))\right),\end{aligned}$$

as claimed. □

If all weights at iterations t are $\zeta \doteq \varepsilon \cdot \underline{w}(\theta)$-dense for $t = 1, 2, ...$, then, letting $u \doteq \varepsilon$ in Lemma N and

$$\Phi \quad \doteq \quad \varepsilon\underline{\phi_{\ell}}(\theta) + (1-\varepsilon)\phi_{\ell *}$$

in Theorem B, for some $\phi_{\ell *}$ to be made precise, then a sufficient condition to get $\Phi(H_T, \mathcal{S}) \leqslant \Phi$ is $\sum_{t=1}^{T} u_t \overline{w}_t^2 \geqslant 2(\Phi(H_0, \mathcal{S}) - \varepsilon\underline{\phi_{\ell}}(\theta) - (1-\varepsilon)\phi_{\ell *})/(\kappa\gamma^2)$ (using (83)), and integrating the ζ-denseness of weights, this condition becomes the sufficient condition:

$$\sum_{t=1}^{T} u_t \quad \geqslant \quad \frac{2(\Phi(H_0, \mathcal{S}) - \varepsilon\underline{\phi_{\ell}}(\theta) - (1-\varepsilon)\phi_{\ell *})}{\kappa\varepsilon^2\underline{w}(\theta)^2\gamma^2}. \tag{85}$$

So, if we pick $\phi_{\ell *} \doteq \min_{i\in[m]} \underline{\phi_{\ell}}(y_i^* H_T(\boldsymbol{x}_i))$, then from Lemma N we get

$$\mathbb{P}_{i\sim[m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] \quad < \quad \varepsilon,$$

which is what we want. We wrap up in two last steps. We first simplify the RHS of (85) by replacing it by a more readable sufficient condition: if the loss' partial losses satisfy

$$\ell_{-1}(0), \ell_1(1) \quad \geqslant \quad C \tag{86}$$

for some $C \in \mathbb{R}$ (such as if the loss is fair: $C = 0$), then we remark that for any $H \in \mathbb{R}$ and $y \in \{0, 1\}$,

$$\begin{aligned}
\phi_{\ell}(-H) - yH \quad &= \quad \sup_{u\in[0,1]} \{(u-y)H + \underline{L}(u)\} \\
&= \quad \sup_{u\in[0,1]} \{(u-y)H + u\ell_1(u) + (1-u)\ell_{-1}(u)\} \\
&\geqslant \quad y\ell_1(y) + (1-y)\ell_{-1}(y).
\end{aligned} \tag{87}$$

The integral representation of proper losses (Reid & Williamson, 2010, Theorem 1) (Nock & Menon, 2020, Appendix Section 9),

$$\ell_1(u) = \int_u^1 (1-t)w(t)\mathrm{d}t \quad , \quad \ell_{-1}(u) = \int_0^u tw(t)\mathrm{d}t,$$

where $(0, 1) \rightarrow \mathbb{R}_+$, shows that ℓ_1 is non-increasing and ℓ_{-1} is non-decreasing, so $\inf \ell_{-1} \geqslant C \neq \pm\infty$ and $\inf \ell_1 \geqslant C \neq \pm\infty$, so (87) yields

$$\phi_{\ell}(-H) - yH \quad \geqslant \quad \left\{ \begin{array}{ccc} \ell_1(1) \geqslant C & \text{if} & y = 1, \\ \ell_{-1}(0) \geqslant C & \text{if} & y = 0 \end{array} \right. , \tag{88}$$

so $\min_{i\in[m]} \phi_{\ell}(y_i^* H_T(\boldsymbol{x}_i)) \geqslant C$ and $\underline{\phi_{\ell}}(\theta) \geqslant C$, which allows us to replace (85) by the sufficient condition:

$$\sum_{t=1}^{T} u_t \quad \geqslant \quad \frac{2(\Phi(H_0, \mathcal{S}) - C)}{\kappa\varepsilon^2\underline{w}(\theta)^2\gamma^2}. \tag{89}$$

In our second step to wrap-up, if we have $\sum_{t=1}^{T} u_t \geqslant U(T)$ (for some U strictly increasing and thus invertible), then under the three conditions:

- **LOSS0** and (86) on the loss,
- **WLA** on the weak learner,
- (83) on Step 2.1 of MODABOOST (the architecture emulation oracle is compliant for the u_t shown),

we are guaranteed that anytime we have

$$T \quad \geqslant \quad U^{-1}\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2\underline{w}(\theta)^2\gamma^2}\right), \tag{90}$$

we are guaranteed

$$\mathbb{P}_{i\sim[m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] \quad < \quad \varepsilon,$$

which is the statement of the Theorem. Figure 4 depicts some key functions used in MODABOOST and Theorem 1.

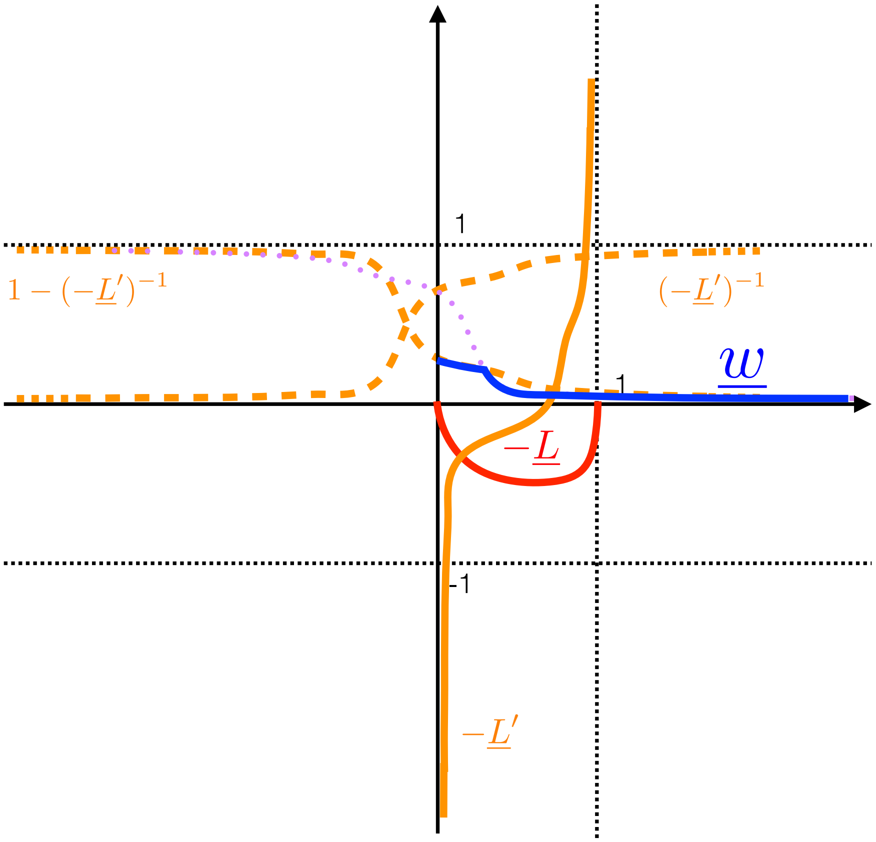


Figure 4. Schematic depiction of key functions used for weights (14) and Theorem 1 on an example of loss whose Bayes risk is schematised in red.

II.7. Proof of Lemma 5

Denote W^+, W^- the total sum of (unnormalized) boosting weights in $\mathcal{S}_t$ before the call for splitting the node. Denote $W^+_{\mathrm{r}}, W^-_{\mathrm{r}}$ the corresponding weights at the new leaf λ and $W^+_{\mathrm{l}}, W^-_{\mathrm{l}}$ the corresponding weights at the next leaf λ', completing the split (See Figure 5). The proof of the Lemma relies on the following observations:

(1) $W^+ = W^-$ (before split, the current leaf is balanced); also, $W^+ = W^+_{\mathrm{l}} + W^+_{\mathrm{r}}$ and $W^- = W^-_{\mathrm{l}} + W^-_{\mathrm{r}}$ (every example in $\mathcal{X}_t$ goes to exactly one new leaf);

(2) the weak learner predicts $1_{x_i \geqslant a} \cdot h_t \in \{-1, 0, 1\}$ wlog (call it the "prediction at the right node of the split at $\mathcal{X}_t$");

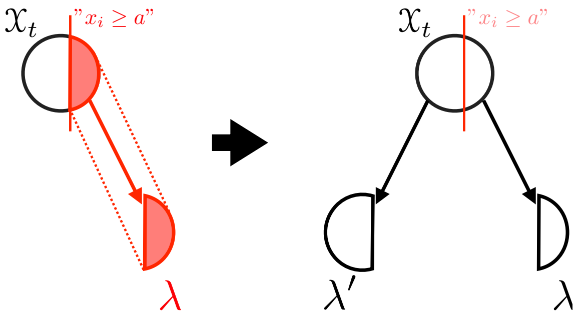


Figure 5. When MODABOOST learns a decision trees, $\mathcal{X}_t$ in Step 2.1 is the domain corresponding to a leaf λ. The weak learner gives the split and fits the prediction of one leaf only to guarantee the WLA. The WLA at this split guarantees that the other split also complies with the WLA (the red parts are chosen by the weak learner, see text).

We then derive from the quantity in absolute value of the **WLA** (20):

$$\begin{aligned}
\sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i^* \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|} &= \sum_{i \in [m]_t : x_i \geqslant a} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i^* \cdot h_t \\
&= \left(\frac{W_{\mathrm{r}}^+}{W^+ + W^-} - \frac{W_{\mathrm{r}}^-}{W^+ + W^-} \right) h_t \\
&= \left(\frac{W^+ - W_{\mathrm{l}}^+}{W^+ + W^-} - \frac{W^- - W_{\mathrm{l}}^-}{W^+ + W^-} \right) h_t \\
&= \left(\frac{W^+ - W^-}{W^+ + W^-} + \frac{W_{\mathrm{l}}^-}{W^+ + W^-} - \frac{W_{\mathrm{l}}^+}{W^+ + W^-} \right) h_t \\
&= \left(\frac{W_{\mathrm{l}}^-}{W^+ + W^-} - \frac{W_{\mathrm{l}}^+}{W^+ + W^-} \right) h_t \\
&= \sum_{i \in [m]_t : x_i < a} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i^* \cdot (-h_t) \\
&= \sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i^* \cdot \frac{h'_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h'_t(\boldsymbol{x}_j)|}, \qquad (91)
\end{aligned}$$

with

$$h'_t(\boldsymbol{x}) \doteq 1_{x_i < a} \cdot (-h_t), \qquad (92)$$

which is both (i) a function computing a prediction for the left node of the split at $\mathcal{X}_t$ and (ii) satisfying the **WLA** since $1_{x_i \geqslant a} \cdot h_t$ does satisfy the **WLA**.

II.8. Proof of Lemma 6

Let λ denote a leaf of the decision tree. We have

$$\begin{aligned} J(\lambda) &= m_\lambda \cdot \left(\frac{\sum_{i\sim\lambda} y_i - y_i^* \cdot (-\underline{L}')^{-1}(H_\lambda)}{m_\lambda}\right)^2 \\ &= m_\lambda \cdot \left(\frac{m_\lambda^+ - (m_\lambda^+ - m_\lambda^-) \cdot \frac{m_\lambda^+}{m_\lambda}}{m_\lambda}\right)^2 \\ &= \frac{(m_\lambda^+)^2}{m_\lambda^3} \cdot \left(m_\lambda - m_\lambda^+ + m_\lambda^-\right)^2 \\ &= 4 \cdot m_\lambda \cdot \left(\frac{m_\lambda^+}{m_\lambda}\right)^2 \cdot \left(\frac{m_\lambda^-}{m_\lambda}\right)^2 \\ &= 4m \cdot \frac{m_\lambda}{m} \cdot \left(\frac{m_\lambda^+}{m_\lambda}\right)^2 \cdot \left(\frac{m_\lambda^-}{m_\lambda}\right)^2 \\ &\propto p_\lambda \cdot (2p_\lambda^+(1-p_\lambda^+))^2, \end{aligned} \quad (93)$$

as claimed. In (93), we have made use of the expression in (97).

II.9. Proof of Lemma 7

We proceed in three steps. Suppose a new leaf λ has been put, with prediction h_λ, by the weak learner (this is in fact ”half a split” as usually described for DTs). We compute the leveraging coefficient α_λ in Step 2.3 of MODABOOST. Denote $\mathrm{parent}(\nu)$ the parent node of ν in H.

$$H_{\mathrm{parent}(\lambda)} \doteq \sum_{\nu\in\mathrm{path}(\lambda)\backslash\{\lambda_t\}} \alpha_\nu h_\nu \quad (94)$$

is the prediction computed from the root of the tree up to the parent of λ. Given a constant prediction h_λ at leaf λ, We wish to find α_λ so that (15) holds. We reuse notations from Lemma 6 and its proof. We note that (15) is equivalent to

$$m_\lambda^+ - (m_\lambda - m_\lambda^+) \cdot (-\underline{L}')^{-1}(\alpha_\lambda h_\lambda + H_{\mathrm{parent}(\lambda)}) - m_\lambda^+ \cdot (-\underline{L}')^{-1}(\alpha_\lambda h_\lambda + H_{\mathrm{parent}(\lambda)}) = 0, \quad (95)$$

which gives, since $p_\lambda^+ = m_\lambda^+/m_\lambda$,

$$\alpha_\lambda = \frac{1}{h_\lambda} \cdot \left((-\underline{L}')\left(p_\lambda^+\right) - H_{\mathrm{parent}(\lambda)}\right), \quad (96)$$

Our second step computes the final decision tree prediction at the new leaf λ, which is trivially:

$$\begin{aligned} H_\lambda &\doteq H_{\mathrm{parent}(\lambda)} + \alpha_\lambda h_\lambda \\ &= (-\underline{L}')\left(p_\lambda^+\right). \end{aligned} \quad (97)$$

Plugging this prediction in the SPD, it simplifies as

$$\begin{aligned} \Phi(H, \mathcal{S}) &= \mathbb{E}_{\lambda\sim\Lambda(h)}\left[(-\underline{L})^\star(-\underline{L}'(p_\lambda^+)) + p_\lambda^+\underline{L}'(p_\lambda^+)\right] \quad (98) \\ &= \mathbb{E}_{\lambda\sim\Lambda(h)}\left[-\underline{L}'(p_\lambda^+) \cdot (-\underline{L}')^{-1}(-\underline{L}'(p_\lambda^+)) + \underline{L} \circ (-\underline{L}')^{-1}(-\underline{L}'(p_\lambda^+)) + p_\lambda^+\underline{L}'(p_\lambda^+)\right] \\ &= \mathbb{E}_{\lambda\sim\Lambda(h)}\left[-p_\lambda^+\underline{L}'(p_\lambda^+) + \underline{L}(p_\lambda^+) + p_\lambda^+\underline{L}'(p_\lambda^+)\right] \\ &= \mathbb{E}_{\lambda\sim\Lambda(h)}\left[\underline{L}(p_\lambda^+)\right], \quad (99) \end{aligned}$$

as claimed.

II.10. Proof of Lemma 8

We can now show a more general result for any loss, encompassing Lemma 8. We reason in terns of noise probability η_{Y} instead of N.

Lemma O. *If one of the two conditions is satisfied:*

- **(S)** *the loss ℓ is symmetric, $\eta_{\text{Y}} < 1/2$ $(= \underline{w}(0))$ and* ModaBoost *is run for any number $T \geqslant 1$ of iterations, or*
- **(A)** *the loss ℓ is asymmetric, $\eta_{\text{Y}} < \underline{w}(0)$ and* ModaBoost *is run for a number of iterations T such that the following condition holds on the* DT *leaves ($p^* \doteq (-\underline{L}')^{-1}(0)$):*

$$\forall \lambda \in \Lambda(H), \left(p^+_\lambda \leqslant \min\left\{\frac{p^* - \eta_{\text{Y}}}{1 - 2\eta_{\text{Y}}}, \frac{1}{2}\right\}\right) \vee \left(1 - p^+_\lambda \leqslant \min\left\{\frac{(1-p^*) - \eta_{\text{Y}}}{1 - 2\eta_{\text{Y}}}, \frac{1}{2}\right\}\right), \quad (100)$$

then ModaBoost *with* DT *is Bayes optimal in T iterations on Long and Servedio's data.*

Proof: We treat case (**S**) first. The proof is straightforward since the noisy proportion of positive examples at the root leaf (first iteration) λ, $\tilde{p}^+_\lambda$, satisfies $\tilde{p}^+_\lambda = \eta_{\text{Y}} + (1 - 2\eta_{\text{Y}})p^+_\lambda$ (p^+_λ = noise-free proportion). Hence, $\tilde{p}^+_\lambda > 1/2$ iff $p^+_\lambda > 1/2$, and $\tilde{p}^+_\lambda < 1/2$ iff $p^+_\lambda < 1/2$. The prediction at the root node is obtained using (97), has the same sign with and without noise since when the loss is symmetric, $-\underline{L}'$ zeroes at $1/2$ (and it is strictly monotonic because the loss is strictly proper). If $T > 1$, we note that for any observation with > 0 probability of occurrence, the local proportion of positive examples is the same for all three distinct observations in Long and Servedio's dataset, hence our argument for the root is still valid for any node given that the prediction at any leaf is still (97).

If the loss is not symmetric, the picture changes compared to symmetric losses. To have a sign flip between no noise and noise, we need either:

$$p^+_\lambda > p^* > \eta_{\text{Y}} + (1 - 2\eta_{\text{Y}})p^+_\lambda, \quad (101)$$

$$p^+_\lambda < p^* < \eta_{\text{Y}} + (1 - 2\eta_{\text{Y}})p^+_\lambda. \quad (102)$$

Note that looking at the extremes, we see that (101) implies $p^+_\lambda > 1/2$ while (102) implies $p^+_\lambda < 1/2$. We have two cases:

Case 1: we reach the point where

$$p^+_\lambda \quad \leqslant \quad \min\left\{\frac{p^* - \eta_{\text{Y}}}{1 - 2\eta_{\text{Y}}}, \frac{1}{2}\right\}.$$

We note that the $1/2$ upperbound prevents (101), while the other one can be reformulated as $\eta_{\text{Y}} + (1 - 2\eta_{\text{Y}})p^+_\lambda \leqslant p^*$, preventing (102).

Case 2: we reach the point where

$$1 - p^+_\lambda \quad \leqslant \quad \min\left\{\frac{(1-p^*) - \eta_{\text{Y}}}{1 - 2\eta_{\text{Y}}}, \frac{1}{2}\right\}.$$

We note that the $1/2$ upperbound prevents (102), while the other one can be reformulated as $p^* \leqslant \eta_{\text{Y}} + (1 - 2\eta_{\text{Y}})p^+_\lambda$, preventing (101).

This ends the proof of Lemma O. □

II.11. Application of ModaBoost #5: labeled branching programs (LBP)

A labeled branching program is a branching program with prediction values at each node, just like our encoding of DT, with the same way of classifying an observation – sum an observation's path values from the root to a leaf. The key difference with classical branching programs is that to one leaf can correspond as many possible predictions as there are paths leading to it. See Figure 6 for an example.

▷ u_t *compliance of* AEO *and the weak learner*: the weak learner is the same as for DT, *except* it looks for a split over the *union* of a set of leaves in the current LBP, with the constraint that this split has to cut every leaf's domain in two (this requirement can be removed if the user is comfortable that some inner nodes in the LBP may have out-degree 1). After the

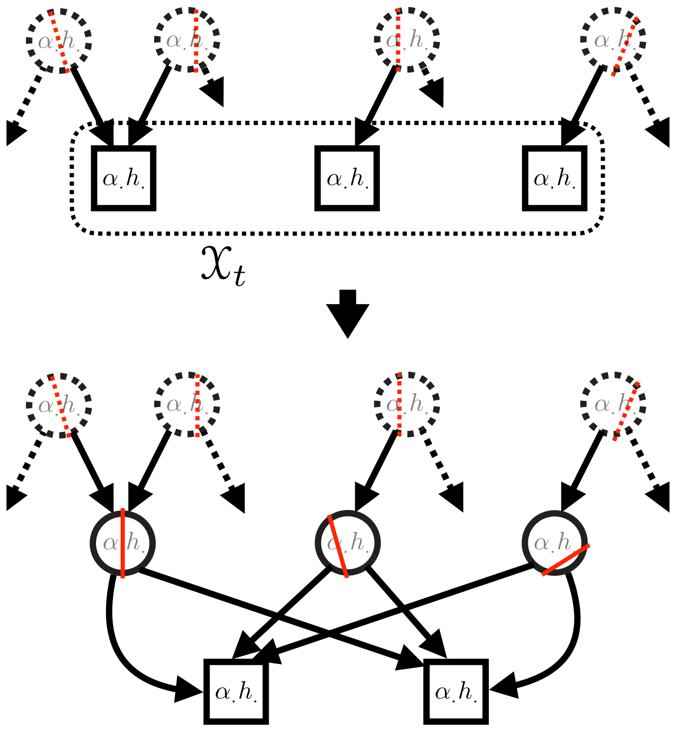


Figure 6. Using MODABOOST to learn a LBP: the main difference with DT is the current $\mathcal{X}_t$ is the union of the domains of several leaves, yielding larger u_ts and more efficient boosting.

split is found, it is carried at each node and the outgoing arcs get to two new leaves only by merging the leaves of the stumps accordingly (call this procedure the *split-merge* process), as displayed in Figure 1. This makes the weak learner have the same properties as for DT, but of course, yields larger u_t compliance than for DT, and so bring better boosting rates as we now show.

▻ *Boosting rate*: suppose we run MODABOOST as for DT and start to merge nodes to always ensure $u_t \geqslant \beta$ for some $\beta > 1/T$. We get $\sum_{t=1}^{T} u_t \geqslant \sum_{t=1}^{\lfloor 1/\beta \rfloor}(1/t) + \beta(T - \lfloor 1/\beta \rfloor) \geqslant \log(1 + \lfloor 1/\beta \rfloor) + \beta T - 1 \geqslant \log(1/\beta) + \beta T - 1$. The choice $\beta = T^{-c}$ for a constant $c \in (0,1)$ immediately leads that $\mathbb{P}_{i\sim[m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon$ if

$$T \quad \geqslant \quad \underbrace{\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma^2_{\text{WL}}}\right)^{\frac{1}{1-c}}}_{\doteq \mathrm{B}_{\mathsf{LBP}}} = (\mathrm{B}_{\mathsf{LS}})^{\frac{1}{1-c}} = \tilde{O}\left(\frac{1}{\varepsilon^{\frac{2}{1-c}} \gamma_{\text{WL}}^{\frac{2}{1-c}}}\right), \tag{103}$$

a bound which is exponentially better than (25) for DT. While it does extend previous boosting rates to margins / edges (Kalai & Servedio, 2003; Mansour & McAllester, 2000), (103) is suboptimal compared to the $\tilde{O}(\log^2(1/\varepsilon))$ dependence of Mansour & McAllester (2000) shown for $\theta = 0$.

▻ *Effect of Long and Servedio's data*: a single node LBP is also a single node DT. Since learning a DT achieves Bayes optimal prediction with a single root DT on Long & Servedio (2010), the same happens for a single root LBP.

Lemma P. *For any $(N, K, \gamma) \in \mathbb{N}_*^2 \times \mathbb{R}_{+*}$,* MODABOOST *with* LBP *is Bayes optimal in 1 iteration on Long and Servedio's data if the loss is symmetric.*

III. Toy experiment

Table A1 presents a toy result of MODABOOST. As predicted by Long & Servedio (2010), LS learned always have very substantial degradation in their estimated posterior below a threshold margin parameter γ, which translates to classifiers as

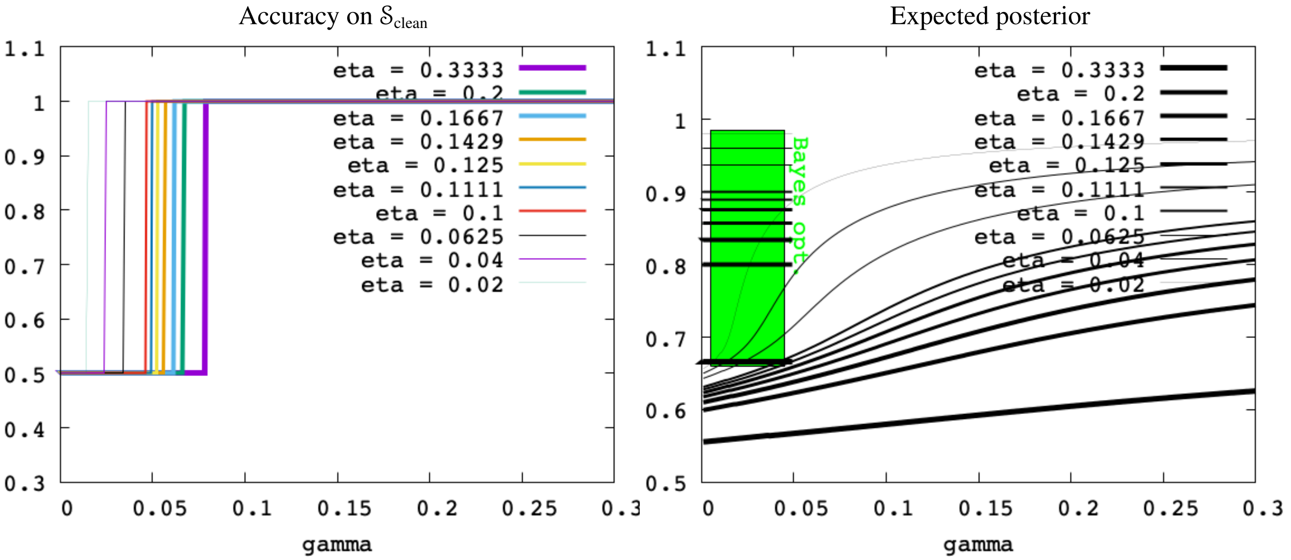


Table A1. MODABOOST's results with the induction of linear separators (LS) on Matusita's loss. Each plot has parameter γ in abscissa (10) and in ordinate respectively the accuracy on $\mathcal{S}_{\text{clean}}$ and the expected posterior estimation from $\mathcal{S}_{\text{noisy}}$ (5) (Bayes' optimum indicated in a green rectangle). Different curves correspond to different values of the noise parameter η_{Y} (indicated as `eta`).

accurate as the unbiased coin on $\mathcal{S}_{\text{clean}}$.