# Random Classification Noise Defeats All Convex Potential Boosters

**Philip M. Long** PLONG@GOOGLE.COM
Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043

**Rocco A. Servedio** ROCCO@CS.COLUMBIA.EDU
Computer Science Department, Columbia University, New York, NY 10027

## Abstract

A broad class of boosting algorithms can be interpreted as performing coordinate-wise gradient descent to minimize some potential function of the margins of a data set. This class includes AdaBoost, LogitBoost, and other widely used and well-studied boosters. In this paper we show that for a broad class of convex potential functions, any such boosting algorithm is highly susceptible to random classification noise. We do this by showing that for any such booster and any nonzero random classification noise rate $\eta$, there is a simple data set of examples which is efficiently learnable by such a booster if there is no noise, but which cannot be learned to accuracy better than 1/2 if there is random classification noise at rate $\eta$. This negative result is in contrast with known branching program based boosters which do not fall into the convex potential function framework and which can provably learn to high accuracy in the presence of random classification noise.

## 1. Introduction

### 1.1. Background

Much work has been done on viewing boosting algorithms as greedy iterative algorithms that perform a coordinate-wise gradient descent to minimize a potential function of the margin of the examples, see e.g. [3, 12, 19, 7, 18, 2]. In this framework every potential function $\phi$ defines an algorithm that may possibly be a boosting algorithm; we denote the algorithm corresponding to $\phi$ by $\mathcal{B}_\phi$. For example, AdaBoost [11] and its confidence-rated generalization [20] may be viewed as the algorithm $\mathcal{B}_\phi$ corresponding to the potential function $\phi(z) = e^{-z}$. The MadaBoost algorithm of Domingo and Watanabe [5] may be viewed as the algorithm $\mathcal{B}_\phi$ corresponding to

$$\phi(z) = \begin{cases} 1 - z & \text{if } z \leq 0 \\ e^{-z} & \text{if } z > 0. \end{cases} \quad (1)$$

(We give a more detailed description of exactly what the algorithm $\mathcal{B}_\phi$ is for a given potential function $\phi$ in Section 2.2.)

### 1.2. Motivation: noise-tolerant boosters?

It has been widely observed that AdaBoost can suffer poor performance when run on noisy data, see e.g. [10, 17, 4]. The most commonly given explanation for this is that the exponential reweighting of examples which it performs (a consequence of the exponential potential function) can cause the algorithm to invest too much "effort" on correctly classifying noisy examples. Boosting algorithms such as MadaBoost [5] and LogitBoost [12] based on a range of other potential functions have subsequently been provided, sometimes with an explicitly stated motivation of rectifying AdaBoost's poor noise tolerance. However, we are not aware of rigorous results establishing provable noise tolerance for any boosting algorithms that fit into the potential functions framework, even for mild forms of noise such as random classification noise (henceforth abbreviated RCN) at low noise rates. This motivates the following question: are Adaboost's difficulties in dealing with noise due solely to its exponential weighting scheme, or are these difficulties inherent in the potential function approach to boosting?

### 1.3. Our results: convex potential boosters cannot withstand random classification noise

This paper shows that the potential function boosting approach provably cannot yield learning algorithms that tolerate even low levels of random classification noise when convex potential functions are used. More

Appearing in *Proceedings of the 25$^{th}$ International Conference on Machine Learning*, Helsinki, Finland, 2008. 

precisely, we exhibit a fixed natural set of base classifiers $h_1, \ldots, h_n$ and show that for every convex function $\phi$ satisfying some very mild conditions and every noise rate $\eta > 0$, there is a multiset $S$ of labeled examples such that the following holds:

- There is a linear separator $\mathrm{sgn}(\alpha_1 h_1 + \cdots + \alpha_n h_n)$ over the base classifiers $h_1, \ldots, h_n$ that correctly labels every example in $S$ with margin $\gamma > 0$ (and hence it is easy for a boosting algorithm trained on $S$ to efficiently construct a final hypothesis that correctly classifies all examples in $S$). However,
- When the algorithm $\mathcal{B}_\phi$ is run on the distribution $\mathcal{D}_{\eta,S}$, it constructs a classifier that has error rate $1/2$ on the examples in $S$. Here $\mathcal{D}_{\eta,S}$ is the uniform distribution over $S$ but where examples are corrupted with random classification noise at rate $\eta$, i.e. labels are independently flipped with probability $\eta$.

This result shows that random classification noise can cause convex potential function boosters to fail in a rather strong sense. We note that as discussed in Section 7, there do exist known boosting algorithms [13, 16] that can tolerate random classification noise, and in particular can efficiently achieve perfect accuracy on $S$, after at most $\mathrm{poly}(1/\gamma)$ stages of boosting, when run on $\mathcal{D}_{\eta,S}$ in the scenario described above.

Recently Bartlett and Traskin proved that the AdaBoost algorithm is consistent if it is stopped after a suitable number of iterations, given certain conditions on a random source generating the data [1]. Our analysis does not contradict theirs because the source in our construction does not satisfy Condition 1 of their paper. To see why this is the case it is useful, as has become customary, to think of the contribution that a given example makes to the potential as a "loss" paid by the learning algorithm. Informally, Condition 1 from [1] requires linear combinations of base classifier predictions to have total loss arbitrarily close to the best possible loss for any measurable function. Our analysis takes advantage of the fact that, for linear combinations of base classifiers with a convex loss function, large-margin errors are especially egregious: we present the learner with a choice between a lot of cheap errors and relatively few expensive errors. If optimization were to be performed over all measurable functions, roughly speaking, it would be possible to make all errors cheap.

Though the analysis required to establish our main result is somewhat delicate, the actual construction is quite simple and admits an intuitive explanation (see Section 4.2). For every convex potential function $\phi$ we use the same set of only $n = 2$ base classifiers (these are confidence-rated base classifiers which output real values in the range $[-1, 1]$), and the multiset $S$ contains only three distinct labeled examples; one of these occurs twice in $S$, for a total multiset size of four. We expect that many other constructions which similarly show the brittleness of convex potential boosters to random classification noise can be given. We describe experiments with one such construction that uses Boolean-valued weak classifiers rather than confidence-rated ones in Section 6.

## 2. Background and Notation

Throughout the paper $X$ will denote the instance space. $\mathcal{H} = \{h_1, \ldots, h_n\}$ will denote a fixed finite collection of *base classifiers* over $X$, where each base classifier is a function $h_i : X \to [-1, 1]$; i.e. we shall work with confidence-rated base classifiers. $S = (x^1, y^1), \ldots, (x^m, y^m) \in (X \times \{-1, 1\})^m$ will denote a multiset of $m$ examples with binary labels.

### 2.1. Convex potential functions

We adopt the following natural definition which, as we discuss in Section 5, captures a broad range of different potential functions that have been studied.

**Definition 1** *We say that $\phi : \mathbf{R} \to \mathbf{R}$ is a* convex potential function *if $\phi$ satisfies the following properties:*

1. *$\phi$ is convex and nonincreasing and $\phi \in C^1$ (i.e. $\phi$ is differentiable and $\phi'$ is continuous);*
2. *$\phi'(0) < 0$ and $\lim_{x \to +\infty} \phi(x) = 0$.*

### 2.2. Convex potential boosters

Let $\phi$ be a convex potential function, $\mathcal{H} = \{h_1, \ldots, h_n\}$ a fixed set of base classifiers, and $S = (x^1, y^1), \ldots, (x^m, y^m)$ a multiset of labeled examples.

Similarly to Duffy and Helmbold [6, 7], we consider an iterative algorithm which we denote $\mathcal{B}_\phi$. The algorithm performs a coordinatewise gradient descent through the space of all possible coefficient vectors for the weak hypotheses, in an attempt to minimize the convex potential function of the margins of the examples. We now give a more precise description of how $\mathcal{B}_\phi$ works when run with $\mathcal{H}$ on $S$.

Algorithm $\mathcal{B}_\phi$ maintains a vector $(\alpha_1, ..., \alpha_n)$ of voting weights for the base classifiers $h_1, ..., h_n$. The weights are initialized to 0. In a given round $T$, the algorithm chooses an index $i_T$ of a base classifier, and modifies

the value of $\alpha_{i_T}$. If $\alpha_{i_T}$ had previously been zero, this can be thought of as adding base classifier number $i_T$ to a pool of voters, and choosing a voting weight.

Let $F(x;\alpha_1,...,\alpha_n) = \sum_{i=1}^n \alpha_i h_i(x)$ be the master hypothesis that the algorithm has constructed prior to stage $T$ (so at stage $T = 1$ the hypothesis $F$ is identically zero.) We write $P_{\phi,S}$ to denote the "global" potential function over $S$

$$P_{\phi,S}(\alpha_1,...,\alpha_n) = \sum_{i=1}^m \phi(y^i F(x^i;\alpha_1,...,\alpha_n)) \qquad (2)$$

which represents the overall potential of a hypothesis vector $(\alpha_1,\ldots,\alpha_n)$ on the sample $S$. It is easy to check that this is a convex function from $\mathbf{R}^n$ (the space of all possible $(\alpha_1,\ldots,\alpha_n)$ coefficient vectors for $F$) to $\mathbf{R}$.

In stage $T$ the algorithm $\mathcal{B}_\phi$ first chooses a base classifier by choosing $i_T$ to be the index $i \in [n]$ which maximizes

$$-\frac{\partial}{\partial \alpha_i} P_{\phi,S}(\alpha_1,...,\alpha_n),$$

and then choosing a new value of $\alpha_{i_T}$ in order to minimize $P_{\phi,S}(\alpha_1,...,\alpha_n)$ for the resulting $\alpha_1,...,\alpha_n$. Thus, in the terminology of [6] we consider "un-normalized" algorithms which preserve the original weighting factors $\alpha_1, \alpha_2$, etc. The AdaBoost algorithm is an example of an algorithm that falls into this framework, as are the other algorithms we discuss in Section 5. Note that the fact that $\mathcal{B}_\phi$ can determine the exactly optimal weak classifier to add in each round errs on the side of pessimism in our analysis.

In our analysis, we will consider the case in which $\mathcal{B}_\phi$ as being run on a distribution $\mathcal{D}_{\eta,S}$ obtained by starting with a finite multiset of examples, and adding independent misclassification noise. One can naturally extend the definition of $\mathcal{B}_\phi$ to apply to probability distributions over $X \times \{-1,1\}$ by extending the definition of potential in (2) as follows

$$P_{\phi,\mathcal{D}}(\alpha_1,...,\alpha_n) = \mathbf{E}_{(x,y)\sim\mathcal{D}}(\phi(yF(x;\alpha_1,...,\alpha_n))). \qquad (3)$$

For rational values of $\eta$, running $\mathcal{B}_\phi$ on (3) for $\mathcal{D} = \mathcal{D}_{\eta,S}$ is equivalent to running $\mathcal{B}_\phi$ over a finite multiset in which each element of $S$ occurs a number of times proportional to its weight under $\mathcal{D}$.

### 2.3. Boosting

Fix a classifier $c : X \to \{-1,1\}$ and a multiset $S = (x^1,y^1),\ldots,(x^m,y^m)$ of labeled examples. We say that a set of base classifiers $\mathcal{H} = \{h_1,\ldots,h_n\}$ is *boostable with respect to $c$ and $S$* if there is a vector $\alpha \in \mathbf{R}^n$ such that for all $i = 1,\ldots,m$, we have

$$\mathrm{sgn}[\alpha_1 h_1(x^i) + \cdots + \alpha_n h_n(x^i)] = y^i.$$

If $\gamma > 0$ is such that

$$\frac{y^i \cdot \big(\alpha_1 h_1(x^i) + \cdots + \alpha_n h_n(x^i)\big)}{\sqrt{\alpha_1^2 + \cdots + \alpha_n^2}} \geq \gamma$$

for all $i$, we say that $\mathcal{H}$ is *boostable w.r.t. $c$ and $S$ with margin $\gamma$*.

It is well known that if $\mathcal{H}$ is boostable w.r.t. $c$ and $S$ with margin $\gamma$, then a range of different boosting algorithms (such as AdaBoost) can be run on the noise-free data set $S$ to efficiently construct a final classifier that correctly labels every example in $S$. As one concrete example, after $O(\frac{\log m}{\gamma^2})$ stages of boosting AdaBoost will construct a linear combination $F(x) = \sum_{i=1}^n \gamma_i h_i(x)$ of the base classifiers such that $\mathrm{sgn}(F(x^i)) = y^i$ for all $i = 1,\ldots,m$; see [11, 20] for details.

### 2.4. Random classification noise and noise-tolerant boosting

Random classification noise is a simple, natural, and well-studied model of how benign (nonadversarial) noise can affect data. Given a multiset $S$ of labeled examples and a value $0 < \eta < \frac{1}{2}$, we write $\mathcal{D}_{\eta,S}$ to denote the distribution corresponding to $S$ corrupted with random classification noise at rate $\eta$. A draw from $\mathcal{D}_{\eta,S}$ is obtained by drawing $(x,y)$ uniformly at random from $S$ and independently flipping the binary label $y$ with probability $\eta$.

We say that an algorithm $\mathcal{B}$ is a *boosting algorithm which tolerates RCN at rate $\eta$* if $\mathcal{B}$ has the following property. Let $c$ be a target classifier, $S$ be a multiset of $m$ examples, and $\mathcal{H}$ be a set of base classifiers such that $\mathcal{H}$ is boostable w.r.t. $c$ and $S$. Then for any $\epsilon > 0$, if $\mathcal{B}$ is run with $\mathcal{H}$ as the set of base classifiers on $\mathcal{D}_{\eta,S}$, at some stage of boosting $\mathcal{B}$ constructs a classifier $g$ which has accuracy

$$\frac{|\{(x^i,y^i) \in S : g(x^i) = y^i\}|}{m} \geq 1 - \eta - \epsilon.$$

The accuracy rate above is in some sense optimal, since known results [13] show that no "black-box" boosting algorithm can be guaranteed to construct a classifier $g$ whose accuracy exceeds $1-\eta$ in the presence of RCN at rate $\eta$. As we discuss in Section 7, there are known boosting algorithms [13, 16] which can tolerate RCN at rate $\eta$ for any $0 < \eta < 1/2$. These algorithms, which do not follow the convex potential function approach but instead build a branching program over the base classifiers, use $\mathrm{poly}(1/\gamma, \log(1/\epsilon))$ stages to achieve accuracy $1-\eta-\epsilon$ in the presence of RCN at rate $\eta$ if $\mathcal{H}$ is boostable w.r.t. $c$ and $S$ with margin $\gamma$.

## 3. Main Result

As was just noted, there do exist boosting algorithms (based on branching programs) that can tolerate RCN. Our main result is that no convex potential function booster can have this property:

**Theorem 2** *Fix any convex potential function $\phi$. For any noise rate $0 < \eta < 1/2$, the algorithm $\mathcal{B}_\phi$ does not tolerate RCN at rate $\eta$.*

We obtain Theorem 2 as a direct consequence of the following stronger result, which shows that there is a simple RCN learning problem for which $\mathcal{B}_\phi$ will in fact misclassify half the examples in $S$.

**Theorem 3** *Fix the instance space $X = [-1,1]^2 \subset \mathbf{R}^2$ and the set $\mathcal{H} = \{h_1(x) = x_1, h_2(x) = x_2\}$ of confidence-rated base classifiers over $X$.*

*For any noise rate $0 < \eta < 1/2$ and any convex potential function $\phi$, there is a target classifier $c$, a value $\gamma > 0$, and a multiset $S$ of four labeled examples (three of which are distinct) such that (a) $\mathcal{H}$ is boostable w.r.t. $c$ and $S$ with margin $\gamma$, but (b) when $\mathcal{B}_\phi$ is run on the distribution $\mathcal{D}_{\eta,S}$, it constructs a classifier which misclassifies two of the four examples in $S$.*

## 4. Proof of Theorem 3

We are given an RCN noise rate $0 < \eta < 1/2$ and a convex potential function $\phi$.

### 4.1. The basic idea

Before specifying the sample $S$ we explain the high-level structure of our argument. Recall from (3) that $P_{\phi,\mathcal{D}}$ is defined as

$$P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2) = \sum_{(x,y)} \mathcal{D}_{\eta,S}(x,y)\phi(y(\alpha_1 x_1 + \alpha_2 x_2)). \quad (4)$$

As noted in Section 2.2 the function $P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2)$ is convex. It follows immediately from the definition of a convex potential function that $P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2) \geq 0$ for all $(\alpha_1,\alpha_2) \in \mathbf{R}^2$.

The high-level idea of our proof is as follows. We shall construct a multiset $S$ of four labeled examples in $[-1,1]^2$ (actually in the unit disc $\{x : \|x\| \leq 1\} \subset \mathbf{R}^2$) such that there is a global minimum $(\alpha_1^*,\alpha_2^*)$ of the corresponding $P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2)$ which has the following two properties:

1. **("high error")** The corresponding classifier $g(x) = \mathrm{sgn}(\alpha_1^* x_1 + \alpha_2^* x_2)$ misclassifies two of the points in $S$ (and thus has error rate 1/2); and
2. **("steep slope")** At the point $(0,0)$, the directional derivative of $P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2)$ in any direction orthogonal to $(\alpha_1^*,\alpha_2^*)$ is not as steep as the directional derivative toward $(\alpha_1^*,\alpha_2^*)$.

We now show that it suffices to establish these two properties to prove part (b) of Theorem 3.[1] Suppose we have such an $S$. Since $P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2)$ depends only on the inner product between $(\alpha_1,\alpha_2)$ and the (normalized) example vectors $(yx_1, yx_2)$, it follows that rotating the set $S$ around the origin by any fixed angle induces a corresponding rotation of the function $P_{\phi,\mathcal{D}}$, and in particular of its minima. (Note that we have used here the fact that every example point in $S$ lies within the unit disc; this ensures that for any rotation of $S$ each weak hypothesis $x_i$ will always give outputs in $[-1,1]$ as required.) Consequently a suitable rotation of $S$ to $S'$ will result in the corresponding rotated function $P_{\phi,\mathcal{D}}$ having a global minimum at a vector which lies on one of the two coordinate axes (say a vector of the form $(0,\tau)$). If this is the case, then the "steep slope" property (2) ensures that the directional derivative at $(0,0)$ in this direction will be steepest, so the convex potential booster $\mathcal{B}_\phi$ will pick a base classifier corresponding to this direction (in this case $h_2$). Since a globally optimal weight vector is available in this direction (the vector of length $\sqrt{(\alpha_1^*)^2 + (\alpha_2^*)^2}$ is such a vector), $\mathcal{B}_\phi$ will select such a vector. Once it has achieved such a global optimum it will not change its hypothesis in any subsequent stage, and thus $\mathcal{B}_\phi$'s hypothesis will have error rate 1/2 on the points in the rotated set $S'$ by the "high error" property (1).

### 4.2. The sample $S$

Now let us define the multiset $S$ of examples. $S$ consists of three distinct examples, one of which is repeated twice. (We shall specify the value of $\gamma$ later and show that $0 < \gamma < \frac{1}{6}$.)

- $S$ contains one copy of the example $x = (1,0)$ with label $y = +1$. (We call this the "large margin" example.)
- $S$ contains two copies of the example $x = (\gamma, -\gamma)$ with label $y = +1$. (We call these examples the "penalizers" since they are the points that $\mathcal{B}_\phi$ will misclassify.)
- $S$ contains one copy of the example $x = (\gamma, 5\gamma)$ with label $y = +1$. (We call this example the "puller" for reasons described below.)

[1] To prove part (a) we need to show that $\mathcal{H}$ is boostable w.r.t. some classifier $c$ and $S$ with margin $\gamma$, but as we shall see this is easy to achieve.

Thus all examples in $S$ are positive. It is immediately clear that the classifier $c(x) = \mathrm{sgn}(x_1)$ correctly classifies all examples in $S$ with margin $\gamma > 0$, so the set $\mathcal{H} = \{h_1(x) = x_1, h_2(x) = x_2\}$ of base classifiers is boostable w.r.t. $c$ and $S$ with margin $\gamma$. We further note that since $\gamma < \frac{1}{6}$, each example in $S$ does indeed lie in the unit disc $\{x : \|x\| \leq 1\}$.

Let us give some intuition for why this set $S$ has the "high error" property. The halfspace whose normal vector is $(1, 0)$ classifies all examples correctly, but the noisy (negative labeled) version of the "large margin" example causes a convex potential function to incur a very large cost for this hypothesis vector. Consequently a lower cost hypothesis can be obtained with a vector that points rather far away from $(1, 0)$. The "puller" example (whose $y$-coordinate is $5\gamma$) outweights the two "penalizer" examples (whose $y$-coordinates are $-\gamma$), so it "pulls" the minimum cost hypothesis vector to point up into the first quadrant – in fact, so far up that the two "penalizer" examples are misclassified by the optimal hypothesis vector for the potential function $\phi$.

In Section 4.3 below we make this intuition precise and show that there is a global minimum $(\alpha_1^*, \alpha_2^*)$ of $P_{\phi,\mathcal{D}}$ for which $\alpha_1^* < \alpha_2^*$. This immediately implies that the corresponding classifier $g(x) = \mathrm{sgn}(\alpha_1^* x_1 + \alpha_2^* x_2)$ misclassifies the two copies of $(\gamma, -\gamma)$ in $S$ and gives us the "high error" property (1). In Section 4.4 we show that this $(\alpha_1^*, \alpha_2^*)$ moreover has the "steep slope" property (2).

### 4.3. The "high error" property: analyzing a global minimum of $P_{\phi,\mathcal{D}}$

Let $1 < N < \infty$ be such that $\eta = \frac{1}{N+1}$, so $1-\eta = \frac{N}{N+1}$.

We have that

$$\begin{aligned} P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2) &= \sum_{(x,y)} \mathcal{D}_{\eta,S}(x, y)\phi(y(\alpha_1 x_1 + \alpha_2 x_2)) \\ &= \frac{1}{4} \sum_{(x,y)\in S} [(1-\eta)\phi(\alpha_1 x_1 + \alpha_2 x_2) \\ &\qquad + \eta\phi(-\alpha_1 x_1 - \alpha_2 x_2)]\,. \end{aligned}$$

It is clear that minimizing $4(N+1)P_{\phi,\mathcal{D}}$ is the same as minimizing $P_{\phi,\mathcal{D}}$ so we shall henceforth work with $4(N+1)P_{\phi,\mathcal{D}}$ since it gives rise to cleaner expressions. We have that $4(N+1)P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2)$ equals

$$\begin{aligned} &\sum_{(x,y)\in S} [N\phi(\alpha_1 x_1 + \alpha_2 x_2) + \phi(-\alpha_1 x_1 - \alpha_2 x_2)] \\ &= N\phi(\alpha_1) + \phi(-\alpha_1) \\ &\quad +2N\phi(\alpha_1\gamma - \alpha_2\gamma) + 2\phi(-\alpha_1\gamma + \alpha_2\gamma) \\ &\quad +N\phi(\alpha_1\gamma + 5\alpha_2\gamma) + \phi(-\alpha_1\gamma - 5\alpha_2\gamma). \quad (5) \end{aligned}$$

Let $L_1(\alpha_1, \alpha_2)$ and $L_2(\alpha_1, \alpha_2)$ be defined as follows:

$$\begin{aligned} L_1(\alpha_1, \alpha_2) &\stackrel{\text{def}}{=} \frac{\partial}{\partial\alpha_1} 4(N+1)P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2) \text{ and} \\ L_2(\alpha_1, \alpha_2) &\stackrel{\text{def}}{=} \frac{\partial}{\partial\alpha_2} 4(N+1)P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2). \end{aligned}$$

For $B > 1$ to be fixed later, let us write $L_1(\alpha)$ to denote $L_1(\alpha, B\alpha)$ and similarly write $L_2(\alpha)$ to denote $L_2(\alpha, B\alpha)$. It is easy to verify that we have

$$\begin{aligned} L_1(\alpha) \;=\; & N\phi'(\alpha) - \phi'(-\alpha) + 2\gamma N\phi'(-(B-1)\alpha\gamma) \\ & -2\gamma\phi'((B-1)\alpha\gamma) + N\gamma\phi'((5B+1)\alpha\gamma) \\ & -\gamma\phi'(-(5B+1)\alpha\gamma) \end{aligned}$$

and

$$\begin{aligned} L_2(\alpha) \;=\; & -2\gamma N\phi'(-(B-1)\alpha\gamma) + 2\gamma\phi'((B-1)\alpha\gamma) \\ & +5\gamma N\phi'((5B+1)\alpha\gamma) - 5\gamma\phi'(-(5B+1)\alpha\gamma). \end{aligned}$$

We introduce the following function to help in the analysis of $L_1(\alpha)$ and $L_2(\alpha)$:

$$\text{for } \alpha \in \mathbf{R}, \qquad Z(\alpha) \stackrel{\text{def}}{=} N\phi'(\alpha) - \phi'(-\alpha).$$

Let us establish some basic properties of this function. Since $\phi$ is differentiable and convex, we have that $\phi'$ is a non-decreasing function. This is easily seen to imply that $Z(\cdot)$ is a non-decreasing function. We moreover have $Z(0) = \phi'(0)(N-1) < 0$. The definition of a convex potential function implies that as $\alpha \to +\infty$ we have $\phi'(\alpha) \to 0^-$, and consequently we have

$$\lim_{\alpha\to+\infty} Z(\alpha) = 0 + \lim_{\alpha\to+\infty} -\phi'(-\alpha) > 0,$$

where the inequality holds since $\phi'(\alpha)$ is a nonincreasing function and $\phi'(0) < 0$. Since $\phi'$ and hence $Z$ is continuous, we have that over the interval $[0, +\infty)$ the function $Z(\alpha)$ assumes every value in the range $[\phi'(0)(N-1), -\phi'(0))$.

Next observe that we may rewrite $L_1(\alpha)$ and $L_2(\alpha)$ as

$$L_1(\alpha) = Z(\alpha) + 2\gamma Z(-(B-1)\alpha\gamma) + \gamma Z((5B+1)\gamma\alpha) \quad (6)$$

and

$$L_2(\alpha) = -2\gamma Z(-(B-1)\alpha\gamma) + 5\gamma Z((5B+1)\gamma\alpha). \quad (7)$$

In the rest of this section we shall show that there are values $\alpha > 0$, $0 < \gamma < 1/6$, $B > 1$ such that $L_1(\alpha) = L_2(\alpha) = 0$. Since $P_{\phi,\mathcal{D}}$ is convex, this will imply that $(\alpha_1^*, \alpha_2^*) \stackrel{\text{def}}{=} (\alpha, B\alpha)$ is a global minimum for the dataset constructed using this $\gamma$, as required.

Let us begin with the following claim which will be useful in establishing $L_2(\alpha) = 0$.

**Claim 4** *For any $B \geq 1$ there is a finite value $\epsilon(B) > 0$ such that*

$$2Z(-(B-1)\epsilon(B)) = 5Z((5B+1)\epsilon(B)) < 0 \quad (8)$$

**Proof:** Fix any value $B \geq 1$. Recalling that $Z(0) = \phi'(0)(N-1) < 0$, at $\epsilon = 0$ the quantity $2Z(-(B-1)\epsilon)$ equals $2\phi'(0)(N-1) < 0$, and as $\epsilon$ increases this quantity does not increase. On the other hand, at $\epsilon = 0$ the quantity $5Z((5B+1)\epsilon)$ equals $5\phi'(0)(N-1) < 2\phi'(0)(N-1)$, and as $\epsilon$ increases this quantity increases to a limit, as $\epsilon \to +\infty$, which is at least $5(-\phi'(0))$. Since $Z$ is continuous, there must be some $\epsilon > 0$ at which the two quantities are equal and are each at most $2\phi'(0)(N-1) < 0$. □

**Observation 5** *The function $\epsilon(B)$ is a continuous and nonincreasing function of $B$ for $B \in [0, \infty)$.*

**Proof:** The larger $B \geq 1$ is, the faster $-(B-1)\epsilon)$ decreases as a function of $\epsilon$ and the faster $(5B+1)\epsilon$ increases as a function of $\epsilon$. Continuity of $\epsilon(\cdot)$ follows from continuity of $Z(\cdot)$. □

We now fix the value of $B$ to be $B \stackrel{\text{def}}{=} 1+\gamma$, where the parameter $\gamma$ will be fixed later. We shall only consider settings of $\alpha, \gamma > 0$ such that $\alpha\gamma = \epsilon(B) = \epsilon(1+\gamma)$; i.e. given a setting of $\gamma$, we shall take $\alpha = \frac{\epsilon(1+\gamma)}{\gamma}$. For any such $\alpha, \gamma$ we have

$$\begin{aligned} L_2(\alpha) &= (7) = \gamma[-2Z(-(B-1)\epsilon(1+\gamma)) \\ &\qquad +5Z((5B+1)\epsilon(1+\gamma))] = 0 \end{aligned}$$

where the last equality is by Claim 4. Now let us consider (6); our goal is to show that for some $\gamma > 0$ it is also 0. For any $(\alpha, \gamma)$ pair with $\alpha\gamma = \epsilon(1+\gamma)$, we have by Claim 4 that

$$\begin{aligned} &2\gamma Z(-(B-1)\gamma\alpha) + \gamma Z((5B+1)\gamma\alpha) \\ &= 2\gamma Z(-(B-1)\epsilon(1+\gamma)) + \gamma Z((5B+1)\epsilon(1+\gamma)) \\ &= 6\gamma Z((5B+1)\epsilon(1+\gamma)) \end{aligned}$$

where the second equality is by Claim 4. Plugging this into (6), we have that for $\alpha = \frac{\epsilon(1+\gamma)}{\gamma}$, the quantity $L_1(\alpha)$ equals 0 if and only if

$$\begin{aligned} Z\left(\frac{\epsilon(1+\gamma)}{\gamma}\right) &= -6\gamma Z((5B+1)\epsilon(1+\gamma)) \\ &= 6\gamma \cdot (-Z((6+5\gamma)\cdot\epsilon(1+\gamma))). \quad (9) \end{aligned}$$

Let us analyze (9). We first note that Observation 5 implies that $\epsilon(1+\gamma)$ is a nonincreasing function of $\gamma$ for $\gamma \in [0, \infty)$. Consequently $\frac{\epsilon(1+\gamma)}{\gamma}$ is a decreasing function of $\gamma$, and since $Z$ is a nonincreasing function, the LHS is a nonincreasing function of $\gamma$. Recall that at $\gamma = 0$ we have $\epsilon(1+\gamma) = \epsilon(1)$ which is some fixed finite positive value by Claim 4. So we have $\lim_{\gamma\to 0^+}$ LHS $= \lim_{x\to+\infty} Z(x) \geq -\phi'(0)$. On the other extreme, since $\epsilon(\cdot)$ is nonincreasing, we have

$$\lim_{\gamma\to+\infty} \text{LHS} \leq \lim_{\gamma\to+\infty} Z\left(\frac{\epsilon(1)}{\gamma}\right) = Z(0) = \phi'(0)(N-1) < 0.$$

So as $\gamma$ varies through $(0, \infty)$, the LHS decreases through all values between $-\phi'(0)$ and 0.

On the other hand, at $\gamma = 0$ the RHS of (9) is clearly 0. Moreover the RHS is always positive for $\gamma > 0$ by Claim 4. Since the RHS is continuous (by continuity of $Z(\cdot)$ and $\epsilon(\cdot)$), this together with the previous paragraph implies that there must be some $\gamma > 0$ for which the LHS and RHS of (9) are the same positive value. So we have shown that there are values $\alpha > 0$, $\gamma > 0$, $B = 1+\gamma$ such that $L_1(\alpha) = L_2(\alpha) = 0$. This concludes the proof of the "high error" property (1).

We close this section by showing that the value of $\gamma > 0$ obtained above is indeed at most $1/6$ (and hence every example in $S$ lies in the unit disc as required). To see this, note that we have shown that for this $\gamma$, we have $Z((6+5\gamma)\epsilon(1+\gamma)) < 0$ and $Z\left(\frac{\epsilon(1+\gamma)}{\gamma}\right) > 0$. Since $Z$ is a nondecreasing function this implies $6+5\gamma < \frac{1}{\gamma}$ which clearly implies $\gamma < 1/6$ as desired.

### 4.4. The "steep slope" property: analyzing directional derivatives

Now we turn to proving that the directional derivative in the orthogonal direction is less steep than in the direction of the global minimum $(\alpha_1^*, \alpha_2^*)$. We have just established that $(\alpha, B\alpha) = (\alpha, (1+\gamma)\alpha)$ is a global minimum for the data set as constructed above. The directional derivative at $(0,0)$ in the direction of this optimum is $\frac{L_1(0)+BL_2(0)}{\sqrt{1+B^2}}$.

Since $\phi'(0) < 0$, by (6) and (7) we have

$$\begin{aligned} L_1(0) &= (1+3\gamma)\phi'(0)(N-1) < 0 \\ L_2(0) &= 3\gamma\phi'(0)(N-1) < 0. \end{aligned}$$

This implies that $L_1(0) < L_2(0) < 0$, which, since $B > 1$, implies $BL_1(0) - L_2(0) < 0$. This means that $(B, -1)$ rather than $(-B, 1)$ is the direction orthogonal to the optimal $(1, B)$ which has negative slope.

Recalling that $B = 1+\gamma$, we have the following in-

equalities:

$$\begin{aligned} B &< 1 + 6\gamma = \frac{(1+3\gamma)+3\gamma}{(1+3\gamma)-3\gamma} \\ B &< \frac{-L_1(0) - L_2(0)}{-L_1(0) + L_2(0)} \quad (10) \\ B(-L_1(0) + L_2(0)) &< -L_1(0) - L_2(0) \quad (11) \\ L_1(0) + BL_2(0) &< BL_1(0) - L_2(0) < 0, \quad (12) \end{aligned}$$

where (11) follows from (10) using $L_1(0) < L_2(0) < 0$. So the directional derivative in the optimal direction $(1, B)$ is steeper than in $(B, -1)$, and the proof of the "steep slope" property, and with it Theorem 3, is complete. □

## 5. Consequences for Known Boosting Algorithms

A wide range of well-studied boosting algorithms are based on potential functions $\phi$ that satisfy our Definition 1. Theorem 2 thus implies that each of the corresponding convex potential function boosters as defined in Section 2.2 cannot tolerate random classification noise at any noise rate $0 < \eta < \frac{1}{2}$. (In some cases the original versions of the algorithms discussed below are not exactly the same as the $\mathcal{B}_\phi$ algorithm as described in Section 2.2 because of small differences such as the way the step size is chosen at each update. Thus we do not claim that Theorem 2 applies directly to each of the original boosting algorithms; however we feel that our analysis strongly suggests that the original boosters may, like the corresponding $\mathcal{B}_\phi$ algorithms, be highly susceptible to random classification noise.)

**AdaBoost and MadaBoost**. As discussed in the Introduction and in [6, 18] the Adaboost algorithm [11] is the algorithm $\mathcal{B}_\phi$ obtained by taking the convex potential function to be $\phi(x) = \exp(-x)$. Similarly the MadaBoost algorithm [5] is based on the potential function $\phi(x)$ defined in Equation (1). Each of these functions clearly satisfies Definition 1.

**LogitBoost and FilterBoost.** As described in [6, 18, 2], the LogitBoost algorithm of [12] is based on the logistic potential function $\ln(1 + \exp(-x))$, which is easily seen to fit our Definition 1. Roughly, FilterBoost [2] combines a variation on the rejection sampling of MadaBoost with the reweighting scheme, and therefore the potential function, of LogitBoost.

## 6. Experiments with Binary-valued Weak Learners

The analysis of this paper leaves open the possibility that a convex potential booster could still tolerate noise if the base classifiers were restricted to be binary-valued. In this section we describe empirical evidence that this is not the case. We generated 100 datasets, applied three convex potential boosters to each, and calculated the training error.

**Data.** Each dataset consisted of 4000 examples, divided into three groups, 1000 large margin examples, 1000 pullers, and 2000 penalizers. The large margin examples corresponded to the example $(1, 0)$ in Section 4.2, the pullers play the role of $(\gamma, 5\gamma)$, and the penalizers collectively play the role of $(\gamma, -\gamma)$.

Each labeled example $(x, y)$ in our dataset is generated as follows. First the label $y$ is chosen randomly from $\{-1, 1\}$. There are 21 features $x_1, \ldots, x_{21}$ that take values in $\{-1, 1\}$. Each large margin example sets $x_1 = \cdots = x_{21} = y$. Each puller assigns $x_1 = \cdots = x_{11} = y$ and $x_{12} = \cdots = x_{21} = -y$. Each penalizer is chosen at random in three stages: (1) the values of a random subset of five of the first eleven features $x_1, \ldots, x_{11}$ are set equal to $y$, (2) the values of a random subset of six of the last ten features $x_{12}, \ldots, x_{21}$ are set equal to $y$, and (3) the remaining ten features are set to $-y$.

At this stage, if we associate a base classifier with each feature $x_i$, then each of the 4000 examples is classified correctly by a majority vote over these 21 base classifiers. Intuitively, when an algorithm responds to the pressure exerted by the noisy large margin examples and the pullers to move toward a hypothesis that is a majority vote over the first 11 features only, then it tends to incorrectly classify the penalizers, because in the penalizers only 5 of those first 11 features agree with the class.

Finally, each class designation $y$ is corrupted with classification noise with probability 0.1.

**Boosters.** We experimented with three boosters: AdaBoost, MadaBoost (which is arguably, loosely speaking, the least convex of the convex potential boosters), and LogitBoost. Each booster was run for 100 rounds.

**Results.** The average training error of AdaBoost over the 100 datasets was 33%. The average for LogitBoost was 30%, and for MadaBoost, 27%.

## 7. Discussion

We have shown that any boosting algorithm based on coordinate-wise gradient descent to optimize a convex potential function satisfying mild conditions cannot tolerate random classification noise. While our results imply strong limits on the noise-tolerance of algorithms that fit this framework, they do not apply to other boosting algorithms such as Freund's Boost-By-Majority algorithm [8] and BrownBoost [9] for which the corresponding potential function is non-convex. An interesting direction for future work is to extend our negative results to a broader class of potential functions, or to other types of boosters such as "regularized" boosters [19, 14].

We close by observing that there do exist efficient boosting algorithms (which do not follow the potential function approach) that can provably tolerate random classification noise [13, 16]. These noise-tolerant boosters work by constructing a branching program over the weak classifiers; the original algorithms of [13, 16] were presented only for binary-valued weak classifiers, but recent work [15] extends the algorithm from [16] to work with confidence-rated base classifiers. A standard analysis (omitted because of space constraints) shows that this boosting algorithm for confidence-rated base classifiers can tolerate random classification noise at any rate $0 < \eta < 1/2$ according to our definition from Section 2.4. In particular, for any noise rate $\eta$ bounded below $1/4$, if this booster is run on the data sets considered in this paper, it can construct a final classifier with accuracy $1-\eta-\epsilon > 3/4$ after $O(\frac{\log 1/\epsilon}{\gamma^2})$ stages of boosting. Since our set of examples $S$ is of size four, though, this means that the booster's final hypothesis will in fact have *perfect* accuracy on these data sets which thwart convex potential boosters.

## References


[1] P. L. Bartlett and M. Traskin. Adaboost is consistent. *JMLR*, 8:2347–2368, 2007.

[2] J. Bradley and R. Schapire. Filterboost: Regression and classification on large datasets. In *NIPS*, 2007.

[3] L. Breiman. Arcing the edge. Technical report 486, Department of Statistics, University of California, Berkeley, 1997.

[4] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–158, 2000.

[5] C. Domingo and O. Watanabe. Madaboost: a modified version of adaboost. In *COLT*, pages 180–189, 2000.

[6] N. Duffy and D. Helmbold. Potential boosters? In *NIPS*, pages 258–264, 1999.

[7] N. Duffy and D. Helmbold. A geometric approach to leveraging weak learners. *TCS*, 284:67–108, 2002.

[8] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[9] Y. Freund. An adaptive version of the boost-by-majority algorithm. *Machine Learning*, 43(3):293–318, 2001.

[10] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *ICML*, pages 148–156, 1996.

[11] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS*, 55(1):119–139, 1997.

[12] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 1998.

[13] A. Kalai and R. Servedio. Boosting in the presence of noise. *JCSS*, 71(3):266–290, 2005. Preliminary version in *Proc. STOC'03*.

[14] G. Lebanon and J. Lafferty. Boosting and maximum likelihood for exponential models. *NIPS*, pages 447–454, 2002.

[15] P. Long and R. Servedio. Adaptive martingale boosting. Unpublished manuscript.

[16] P. Long and R. Servedio. Martingale boosting. In *COLT*, pages 79–94, 2005.

[17] Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. In *AAAI/IAAI*, pages 546–551, 1997.

[18] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *NIPS*, pages 512–518, 1999.

[19] G. R atsch, T. Onoda, and K.-R. M uller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

[20] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.