Machine Learning manuscript No.
(will be inserted by the editor)

Random Classification Noise Defeats All Convex Potential Boosters

Philip M. Long · Rocco A. Servedio

Received: date / Accepted: date

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 η, 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 η. This holds even if the booster regularizes using early stopping or a bound on the L_1 norm of the voting weights. 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.

Keywords boosting · learning theory · noise-tolerant learning · misclassification noise · convex loss · potential boosting

Mathematics Subject Classification (2000) 68Q32 · 68T05

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, 13, 22, 8, 21, 2]. In this framework every potential

Philip M. Long
Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043
E-mail: plong@google.com

Rocco A. Servedio
Computer Science Department, Columbia University, New York, NY 10027
E-mail: rocco@cs.columbia.edu

function ϕ defines an algorithm that may possibly be a boosting algorithm; we denote the algorithm corresponding to ϕ by $\mathcal{B}_\phi$. For example, AdaBoost [12] and its confidence-rated generalization [23] 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 [6] may be viewed as the algorithm $\mathcal{B}_\phi$ corresponding to

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

(We give a more detailed description of exactly what the algorithm $\mathcal{B}_\phi$ is for a given potential function ϕ in Section 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. [11, 18, 5]. 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 [6] and LogitBoost [13] 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 precisely, we exhibit a fixed natural set of base classifiers $h_1, \dots, h_n$ and show that for every convex function ϕ 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 $\text{sgn}(\alpha_1 h_1 + \cdots + \alpha_n h_n)$ over the base classifiers $h_1, \dots, h_n$ 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 η, i.e. labels are independently flipped with probability η.

We also show that convex potential boosters are not saved by regularization through early stopping [20,26] or a bound on the L_1 norm of the voting weights (see [22,17]).

These results show that random classification noise can cause convex potential function boosters to fail in a rather strong sense. We note that as discussed in Section 9, there do exist known boosting algorithms [14,15] that can tolerate random classification noise, and in particular can efficiently achieve perfect accuracy on S, after at most poly$(1/\gamma)$ stages of boosting, when run on $\mathcal{D}_{\eta,S}$ in the scenario described above.

A number of recent results have established the statistical consistency of boosting algorithms [4,19,25,17,26,1] under various assumptions on a random source generating the data. Our analysis does not contradict theirs roughly for the following reason. The output of a boosting classifier takes the form $\text{sign}(f(x))$, where the unthresholded $f(x)$ can be thought of as incorporating a confidence rating – usually, this is how much more weight votes for one class than the other. The analyses that establish the consistency of boosting algorithms typically require a linear f to have "potential" as good as any f (see e.g. Condition 1 from [1]). In this paper, we exploit the fact that convex potential boosters choose linear hypotheses to force the choice between many "cheap" errors and few "expensive" ones. If any f is allowed, then an algorithm can make all errors equally cheap by making all classifications with equally low confidence.

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 ϕ 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 8.

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.

For each convex potential function ϕ, we will consider three kinds of convex potential boosters: global-minimizing convex potential boosters, L_1-regularized convex potential boosters, and early-stopping convex potential boosters. First, we will define a convex potential function, then each kind of boosting algorithm in turn.

2.1 Convex potential functions

We adopt the following natural definition which, as we discuss in Section 7, 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 ϕ satisfies the following properties:

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

2.2 Convex potential boosters

Let ϕ 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.

All the boosting algorithms will choose voting weights $\alpha_1, ..., \alpha_n$ and output the classifier

$$\operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i h_i(x)\right)$$

obtained by taking the resulting vote over the base classifier predictions. Let

$$F(x; \alpha_1, ..., \alpha_n) = \sum_{i=1}^{n} \alpha_i h_i(x)$$

denote the quantity whose sign is the outcome of the vote, and whose magnitude reflects how close the vote was.

2.3 Global-minimizing convex potential boosters

The most basic kind of convex potential booster is the idealized algorithm that chooses voting weights $\alpha_1, ..., \alpha_n$ to minimize the "global" potential function over S:

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

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 $\mathbf{R}$. We will denote this booster by B_ϕ^{ideal}.

2.4 L_1-regularized boosters

For any $C > 0$, the L_1-regularized booster minimizes $P_{\phi,S}$ subject to the constraint that $\sum_{i=1}^{n} |\alpha_i| \le C$. We will denote this booster by $B_{\phi,C}^{L_1}$; see [22,17] for algorithms of this sort.

2.5 Early-stopping regularized boosters

To analyze regularization by early stopping, we must consider how the optimization is performed. Similarly to Duffy and Helmbold [7,8], 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 α_{i_T}. If α_{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).

In stage T the algorithm $\mathcal{B}_\phi$ first chooses a base classifier by chooses 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 α_{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 [7] we consider "un-normalized" algorithms which preserve the original weighting factors α_1, α_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 7. 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.

For each K, let $B_{\phi,K}^{\text{early}}$ be the algorithm that performs K iterations of $\mathcal{B}_\phi$, and then halts and outputs the resulting classifier.

2.6 Distributions with noise

In our analysis, we will consider the case in which the boosters are 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 each type of booster to apply to probability distributions over $X \times \{-1, 1\}$ by extending the definition of potential in (2) as follows:

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

For rational values of η, 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 $\mathcal{D}$.

2.7 Boosting

Fix a classifier $c : X \to \{-1, 1\}$ and a multiset $S = (x^1, y^1), \ldots, (x^m, y^m)$ of examples labeled according to c. 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

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

If $\gamma > 0$ is such that

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

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

It is well known that if $\mathcal{H}$ is boostable w.r.t. c and S with margin γ, 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 [12,23] for details.

2.8 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 η. 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 η.

We say that an algorithm $\mathcal{B}$ is a *boosting algorithm which tolerates RCN at rate* η 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 [14] 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 η. As we discuss in Section 9, there are known boosting algorithms [14,15] which can tolerate RCN at rate η 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 η if $\mathcal{H}$ is boostable w.r.t. c and S with margin γ.

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 1 *Fix any convex potential function ϕ and any noise rate $0 < \eta < 1/2$. Then*

(i) The global-minimizing booster $B_{\phi}^{\mathrm{ideal}}$ does not tolerate RCN at rate η;
(ii) For any number K of rounds, the early-stopping regularized booster $B_{\phi,K}^{\mathrm{early}}$ does not tolerate RCN at rate η; and
(iii) For any $C > 0$, the L_1-regularized booster $B_{\phi,C}^{L_1}$ does not tolerate RCN at rate η.

Our first analysis holds for the global optimization and early-stopping convex potential boosters. It establishes parts (i) and (ii) of Theorem 1 through the following stronger statement, which shows that there is a simple RCN learning problem for which B_ϕ^{ideal} and $B_{\phi,K}^{\text{early}}$ will in fact misclassify half the examples in S.

Theorem 2 *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.*

There is a target classifier c such that for any noise rate $0 < \eta < 1/2$ and any convex potential function ϕ, there is 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 γ, but (b) when B_ϕ^{ideal} or $B_{\phi,K}^{\text{early}}$ is run on the distribution $\mathcal{D}_{\eta,S}$, it constructs a classifier which misclassifies two of the four examples in S.

Our theorem about L_1 which establishes part (iii) is as follows.

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.*

There is a target classifier c such that for any noise rate $0 < \eta < 1/2$ and any convex potential function ϕ, any $C > 0$ and any $\beta > 0$, there is a value $\gamma > 0$ and a multiset S of examples such that (a) $\mathcal{H}$ is boostable w.r.t. c and S with margin γ, but (b) when the L_1-regularized potential booster $B_{\phi,C}^{L_1}$ is run on the distribution $\mathcal{D}_{\eta,S}$, it constructs a classifier which misclassifies $\frac{1}{2} - \beta$ fraction of the examples in S.

Section 4 contains our analysis for the global optimization booster B_ϕ^{ideal}; the early stopping and L_1 regularization boosters are dealt with in Sections 5 and 6 respectively.

4 Analysis of the global optimization booster

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

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

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

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 (α_1^*, α_2^*) of the corresponding $P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2)$ for which the corresponding classifier $g(x) = \text{sgn}(\alpha_1^* x_1 + \alpha_2^* x_2)$ misclassifies two of the points in S (and thus has error rate 1/2).

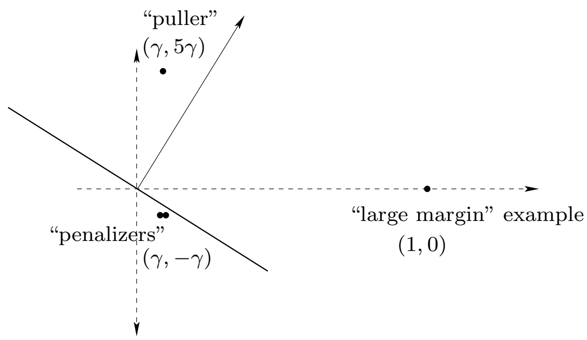

Fig. 1 The sample S. All four examples are positive. We show that for a suitable $0 < \gamma < 1/6$ (based on the convex potential function ϕ), the "puller" example at $(\gamma, 5\gamma)$ causes the optimal hypothesis vector to incorrectly label the two "penalizer" examples as negative.

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 γ 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.)

Thus all examples in S are positive. It is immediately clear that the classifier $c(x) = \text{sgn}(x_1)$ 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 γ. 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. 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 5γ) 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 ϕ. See Figure 1.

4.3 Proof of Theorem 2 for the B_ϕ^{ideal} booster

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

We have that

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

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

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

Let $P_1(\alpha_1,\alpha_2)$ and $P_2(\alpha_1,\alpha_2)$ be defined as follows:

$$\begin{aligned}
P_1(\alpha_1,\alpha_2) &\stackrel{\text{def}}{=} \frac{\partial}{\partial\alpha_1} 4(N+1)P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2) \quad \text{and} \\
P_2(\alpha_1,\alpha_2) &\stackrel{\text{def}}{=} \frac{\partial}{\partial\alpha_2} 4(N+1)P_{\phi,\mathcal{D}}(\alpha_1,\alpha_2).
\end{aligned}$$

Differentiating by α_1 and α_2 respectively, we have

$$\begin{aligned}
P_1(\alpha_1,\alpha_2) = {} & N\phi'(\alpha_1) - \phi'(-\alpha_1) \\
& +2\gamma N\phi'(\alpha_1\gamma - \alpha_2\gamma) - 2\gamma\phi'(-\alpha_1\gamma + \alpha_2\gamma) \\
& +N\gamma\phi'(\alpha_1\gamma + 5\alpha_2\gamma) - \gamma\phi'(-\alpha_1\gamma - 5\alpha_2\gamma)
\end{aligned}$$

and

$$\begin{aligned}
P_2(\alpha_1,\alpha_2) = {} & -2\gamma N\phi'(\alpha_1\gamma - \alpha_2\gamma) + 2\gamma\phi'(-\alpha_1\gamma + \alpha_2\gamma) \\
& +5\gamma N\phi'(\alpha_1\gamma + 5\alpha_2\gamma) - 5\gamma\phi'(-\alpha_1\gamma - 5\alpha_2\gamma).
\end{aligned}$$

Some expressions will be simplified if we reparameterize by setting $\alpha_1 = \alpha$ and $\alpha_2 = B\alpha$. It is helpful to think of $B > 1$ as being fixed (its value will be chosen later). Now, let us write $P_1(\alpha)$ to denote $P_1(\alpha, B\alpha)$ and similarly write $P_2(\alpha)$ to denote $P_2(\alpha, B\alpha)$, so that

$$\begin{aligned}
P_1(\alpha) = {} & N\phi'(\alpha) - \phi'(-\alpha) + 2\gamma N\phi'(-(B-1)\alpha\gamma) \\
& -2\gamma\phi'((B-1)\alpha\gamma) + N\gamma\phi'((5B+1)\alpha\gamma) \\
& -\gamma\phi'(-(5B+1)\alpha\gamma)
\end{aligned}$$

and

$$\begin{aligned}
P_2(\alpha) = {} & -2\gamma N\phi'(-(B-1)\alpha\gamma) + 2\gamma\phi'((B-1)\alpha\gamma) \\
& +5\gamma N\phi'((5B+1)\alpha\gamma) - 5\gamma\phi'(-(5B+1)\alpha\gamma).
\end{aligned}$$

We introduce the following function to help in the analysis of $P_1(\alpha)$ and $P_2(\alpha)$:

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

Let us establish some basic properties of this function. Since ϕ is differentiable and convex, we have that ϕ' is a non-decreasing function. Since $N > 1$, this implies 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 nondecreasing function and $\phi'(0) < 0$. Since ϕ' 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 $P_1(\alpha)$ and $P_2(\alpha)$ as

$$P_1(\alpha) = Z(\alpha) + 2\gamma Z(-(B-1)\alpha\gamma) + \gamma Z((5B+1)\gamma\alpha) \tag{6}$$

and

$$P_2(\alpha) = -2\gamma Z(-(B-1)\alpha\gamma) + 5\gamma Z((5B+1)\gamma\alpha). \tag{7}$$

In the rest of this section we shall show that there are values $\alpha > 0$, $0 < \gamma < 1/6$, $B > 1$ such that $P_1(\alpha) = P_2(\alpha) = 0$. 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 γ, as required.

Let us begin with the following claim which will be useful in establishing $P_2(\alpha) = 0$.

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

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

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 ϵ 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 ϵ 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 4 *The function $\epsilon(B)$ is a continuous and nonincreasing function of B for $B \in [1, \infty)$.*

Proof The larger $B \geq 1$ is, the faster $-(B-1)\epsilon$ decreases as a function of ϵ and the faster $(5B+1)\epsilon$ increases as a function of ϵ. 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 γ 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 γ, we shall take $\alpha = \frac{\epsilon(1+\gamma)}{\gamma}$. For any such α, γ we have

$$P_2(\alpha) = (7) \;=\; \gamma[-2Z(-(B-1)\epsilon(1+\gamma)) + 5Z((5B+1)\epsilon(1+\gamma))] = 0$$

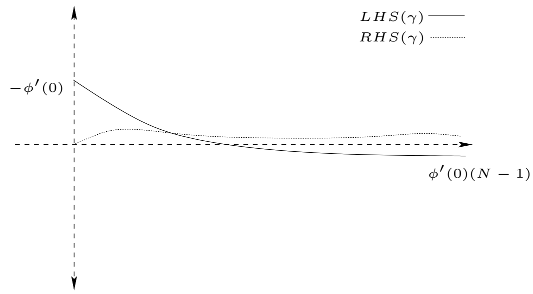

Fig. 2 The LHS of (9) (solid line) and RHS of (9) (dotted line), plotted as a function of γ. As γ ranges through $(0, \infty)$ the LHS decreases through all values between $-\phi'(0)$ (a positive value) and $\phi'(0)(N-1)$ (a negative value). The RHS is 0 at $\gamma = 0$ and is positive for all $\gamma > 0$. Since both LHS and RHS are continuous, there must be some value of $\gamma > 0$ at which the LHS and RHS are equal.

where the last equality is by Proposition 1. Now let us consider (6); our goal is to show that for some $\gamma > 0$ it is also 0. For any (α, γ) pair with $\alpha\gamma = \epsilon(1+\gamma)$, we have by Proposition 1 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 Proposition 1. Plugging this into (6), we have that for $\alpha = \frac{\epsilon(1+\gamma)}{\gamma}$, the quantity $P_1(\alpha)$ equals 0 if and only if

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

Let us analyze (9). We first note that Observation 4 implies that $\epsilon(1+\gamma)$ is a nonincreasing function of γ for $\gamma \in [0, \infty)$. Consequently $\frac{\epsilon(1+\gamma)}{\gamma}$ is a decreasing function of γ, and since Z is a nondecreasing function, the LHS is a nonincreasing function of γ. Recall that at $\gamma = 0$ we have $\epsilon(1+\gamma) = \epsilon(1)$ which is some fixed finite positive value by Proposition 1. So we have $\lim_{\gamma\to 0^+} \text{LHS} = \lim_{x\to+\infty} Z(x) \geq -\phi'(0)$. 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 γ 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 Proposition 1. 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. (See Figure 2.) So we have shown that there are values $\alpha > 0$, $\gamma > 0$, $B = 1+\gamma$ such that $P_1(\alpha) = P_2(\alpha) = 0$.

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 γ, 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.

This concludes the proof of Theorem 2 for the B_ϕ^{ideal} booster.

5 Early stopping

In this section, we show that early stopping cannot save a boosting algorithm: it is possible that the global optimum analyzed in the preceding section can be reached after the first iteration.

Since $P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2)$ depends only on the inner product between (α_1, α_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 $(0, \tau)$). The weight vector $(1, 0)$ achieved a margin γ for the original construction: since rotating this weight vector can only increase its L_1 norm, the rotated weight vector also achieves a margin γ.

Now, all that remains is to show that $B_{\phi,K}^{\mathrm{early}}$ will choose the ultimately optimal direction during the first round of boosting. For this to be the case after rotating, all we need before rotating is that at the point $(0, 0)$, the directional derivative of $P_{\phi,\mathcal{D}}(\alpha_1, \alpha_2)$ in any direction orthogonal to (α_1^*, α_2^*) is not as steep as the directional derivative toward (α_1^*, α_2^*), which we will now prove.

In Section 4, we established that $(\alpha, B\alpha) = (\alpha, (1 + \gamma)\alpha)$ is a global minimum for the data set as constructed there. The directional derivative at $(0, 0)$ in the direction of this optimum is $\frac{P_1(0)+BP_2(0)}{\sqrt{1+B^2}}$.

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

$$\begin{aligned}
P_1(0) &= (1 + 3\gamma)\phi'(0)(N - 1) < 0 \\
P_2(0) &= 3\gamma\phi'(0)(N - 1) < 0.
\end{aligned}$$

This implies that $P_1(0) < P_2(0) < 0$, which, since $B > 1$, implies $BP_1(0) - P_2(0) < 0$. 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 inequalities:

$$\begin{aligned}
B &< 1 + 6\gamma = \frac{(1 + 3\gamma) + 3\gamma}{(1 + 3\gamma) - 3\gamma} \\
B &< \frac{-P_1(0) - P_2(0)}{-P_1(0) + P_2(0)} \qquad (10)\\
B(-P_1(0) + P_2(0)) &< -P_1(0) - P_2(0) \qquad (11)\\
P_1(0) + BP_2(0) &< BP_1(0) - P_2(0) < 0, \qquad (12)
\end{aligned}$$

where (11) follows from (10) using $P_1(0) < P_2(0) < 0$. So the directional derivative in the optimal direction $(1, B)$ is steeper than in $(B, -1)$ □

6 L_1 regularization

Our treatment of L_1 regularization relies on the following intuition. One way to think of the beneficial effect of regularizing a convex potential booster is that regularization controls the impact of the convexity – limiting the weights limits the size of the margins, and thus the extremity of the losses on large-margin errors. But the trouble with regularization is that the convexity is sometimes needed to encourage the boosting algorithm to classify examples correctly: if the potential function is effectively a linear function of the margin, then the booster "cares" as much about enlarging the margins of already correctly classified examples as it does about correcting examples that are classified incorrectly.

In the absence of noise, our construction for regularized boosters concentrates the weight on two examples:

- one positive example at $(2\gamma, -\gamma)$ with weight $1+\epsilon$ (where $\epsilon > 0$), and
- one positive example at $(-\gamma, 2\gamma)$ with weight 1.

As in Section 2.3, when there is noise, for $N > 1$, each clean example will have weight N, and each noisy example weight 1. (Note once again that if N and ϵ are rational, these can be realized with a finite multiset of examples.) Thus, the L_1-regularized convex potential boosting algorithm will solve the following optimization problem:

$$\begin{aligned}
&\min_{\alpha_1,\alpha_2} Q(\alpha_1,\alpha_2), \text{s.t. } |\alpha_1| + |\alpha_2| \le C,\\
&\text{where } Q(\alpha_1,\alpha_2) = (1+\epsilon)N\phi(2\alpha_1\gamma - \alpha_2\gamma) + N\phi(-\alpha_1\gamma + 2\alpha_2\gamma)\\
&\qquad\qquad +(1+\epsilon)\phi(-2\alpha_1\gamma + \alpha_2\gamma) + \phi(\alpha_1\gamma - 2\alpha_2\gamma).
\end{aligned} \tag{13}$$

Let us redefine $P_1(\alpha_1, \alpha_2)$ and $P_2(\alpha_1, \alpha_2)$ to be the partial derivatives with respect to Q:

$$\begin{aligned}
P_1(\alpha_1,\alpha_2) = \frac{\partial Q(\alpha_1,\alpha_2)}{\partial \alpha_1} &= 2\gamma(1+\epsilon)N\phi'(2\alpha_1\gamma - \alpha_2\gamma) - \gamma N\phi'(-\alpha_1\gamma + 2\alpha_2\gamma)\\
&\quad -2\gamma(1+\epsilon)\phi'(-2\alpha_1\gamma + \alpha_2\gamma) + \gamma\phi'(\alpha_1\gamma - 2\alpha_2\gamma)\\
P_2(\alpha_1,\alpha_2) = \frac{\partial Q(\alpha_1,\alpha_2)}{\partial \alpha_2} &= -\gamma(1+\epsilon)N\phi'(2\alpha_1\gamma - \alpha_2\gamma) + 2\gamma N\phi'(-\alpha_1\gamma + 2\alpha_2\gamma)\\
&\quad +\gamma(1+\epsilon)\phi'(-2\alpha_1\gamma + \alpha_2\gamma) - 2\gamma\phi'(\alpha_1\gamma - 2\alpha_2\gamma).
\end{aligned}$$

The following key lemma characterizes the consequences of changing the weights when γ is small enough.

Lemma 1 *For all $C > 0$, $N > 1$, $1 > \epsilon > 0$, there is a $\gamma > 0$ such that, for all α_1, α_2 for which*

$$|\alpha_1| + |\alpha_2| \le C,$$

we have

$$P_1(\alpha_1, \alpha_2) < P_2(\alpha_1, \alpha_2) < 0.$$

Proof If $|\alpha_1| + |\alpha_2| \leq C$, then $|2\alpha_1 - \alpha_2| \leq 3C$ and $|2\alpha_2 - \alpha_1| \leq 3C$. Thus, by making $\gamma > 0$ arbitrarily small, we can make $|2\alpha_1\gamma - \alpha_2\gamma|$ and $|2\alpha_2\gamma - \alpha_1\gamma|$ arbitrarily close to 0. Since ϕ' is continuous, this means that for any $\tau > 0$, there is a $\gamma > 0$ such that, whenever $|\alpha_1| + |\alpha_2| \leq C$, we have

$$\begin{aligned}
|\phi'(2\alpha_1\gamma - \alpha_2\gamma) - \phi'(0)| &< \tau \\
|\phi'(2\alpha_2\gamma - \alpha_1\gamma) - \phi'(0)| &< \tau \\
|\phi'(-2\alpha_1\gamma + \alpha_2\gamma) - \phi'(0)| &< \tau \\
|\phi'(-2\alpha_2\gamma + \alpha_1\gamma) - \phi'(0)| &< \tau.
\end{aligned}$$

For such a γ, we have

$$\begin{aligned}
\frac{P_1(\alpha_1,\alpha_2) - P_2(\alpha_1,\alpha_2)}{\gamma} &= 3(1+\epsilon)N\phi'(2\alpha_1\gamma - \alpha_2\gamma) - 3N\phi'(-\alpha_1\gamma + 2\alpha_2\gamma) \\
&\quad -3(1+\epsilon)\phi'(-2\alpha_1\gamma + \alpha_2\gamma) + 3\phi'(-2\alpha_2\gamma + \alpha_1\gamma) \\
&< 3((1+\epsilon)N + 1)(\phi'(0) + \tau) - 3(1 + \epsilon + N)(\phi'(0) - \tau) \\
&= 3\epsilon(N-1)\phi'(0) + 3(2(N+1) + \epsilon(N+1))\tau.
\end{aligned}$$

Since $\phi'(0) < 0$, $\epsilon > 0$, and $N > 1$, for sufficiently small τ, we have

$$\frac{P_1(\alpha_1,\alpha_2) - P_2(\alpha_1,\alpha_2)}{\gamma} < 0$$

and since $\gamma > 0$, this means

$$P_1(\alpha_1,\alpha_2) < P_2(\alpha_1,\alpha_2).$$

Furthermore

$$\begin{aligned}
P_2(\alpha_1,\alpha_2) &< -\gamma(1+\epsilon)N(\phi'(0) - \tau) + 2\gamma N(\phi'(0) + \tau) \\
&\quad +\gamma(1+\epsilon)(\phi'(0) + \tau) - 2\gamma(\phi'(0) - \tau) \\
&= \gamma(1-\epsilon)(N-1)\phi'(0) + \gamma(3+\epsilon)(N+1)\tau
\end{aligned}$$

so

$$\frac{P_2(\alpha_1,\alpha_2)}{\gamma} < (1-\epsilon)(N-1)\phi'(0) + (3+\epsilon)(N+1)\tau.$$

Again, since $\phi'(0) < 0$, $1 > \epsilon > 0$, and $N > 0$, this means that when τ gets small enough

$$\frac{P_2(\alpha_1,\alpha_2)}{\gamma} < 0$$

and thus $P_2(\alpha_1,\alpha_2) < 0$. □

Lemma 2 *For all $C > 0$, $N > 1$, and $1 > \epsilon > 0$ there is a $\gamma > 0$ such that the output (α_1^*, α_2^*) of the L_1-regularized potential booster for ϕ satisfies $\alpha_1^* > 0$, $\alpha_2^* = 0$.*

Proof By Lemma 1, there is a $\gamma > 0$ such that whenever $|\alpha_1| + |\alpha_2| \leq C$,

$$P_1(\alpha_1,\alpha_2) < P_2(\alpha_1,\alpha_2) < 0.$$

For such a γ, if either of the coordinates of the optimal solution were negative, we could improve the solution while reducing the L_1 norm of the solution by making the negative component less so, a contradiction. Also, if α_2^* were strictly positive, then we could improve the solution without affecting the L_1 norm of the solution by transferring a small amount of the weight from α_2^* to α_1^*, again, a contradiction. □

Looking at the proof of Lemma 1 it is easy to see that the lemma actually holds for all sufficiently small $\gamma > 0$, and thus we may suppose that the instances $(2\gamma, -\gamma)$ and $(-\gamma, 2\gamma)$ lie in the unit square $[-1, 1]^2$. Lemma 2 thus implies Theorem 3 because if $\alpha_1^* > 0$ and $\alpha_2^* = 0$, the positive example $(-\gamma, 2\gamma)$ is classified incorrectly.

7 Consequences for Known Boosting Algorithms

A wide range of well-studied boosting algorithms are based on potential functions ϕ that satisfy our Definition 1. Theorem 1 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 1 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 [7,21] the Adaboost algorithm [12] is the algorithm $\mathcal{B}_\phi$ obtained by taking the convex potential function to be $\phi(x) = \exp(-x)$. Similarly the MadaBoost algorithm [6] 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 [7,21,2], the LogitBoost algorithm of [13] 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.

8 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%.

9 Discussion

We have shown that a range of different types of boosting algorithms that 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 [9] and BrownBoost [10] 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.

The L_1 regularized boosting algorithms considered here fix a bound on the norm of the voting weights before seeing any data. This leaves open the possibility that an algorithm that adapts this bound to the data may still tolerate random misclassification noise. We suspect that this type of adaptiveness in fact cannot confer noise-tolerance; it would be interesting to show this.

There are efficient boosting algorithms (which do not follow the potential function approach) that can provably tolerate random classification noise [14, 15]. These noise-tolerant boosters work by constructing a branching program over the weak classifiers; the original algorithms of [14, 15] were presented only for binary-valued weak classifiers, but recent work [16] extends the algorithm from [15] to work with confidence-rated base classifiers. A standard analysis 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 2.8. In particular, for any noise rate η 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.

This work thus points out a natural attractive property that some branching program boosters have, but all convex potential boosters do not. It would be interesting to further explore the relative capabilities of these classes of algorithms; some concrete goals along these lines include investigating under what conditions branching program boosters can be shown to be consistent, and working toward a characterization of the sources for which one kind of method or another is to be preferred. The fact that convex potential boosters have been shown to be consistent when applied with weak learners that use rich hypothesis spaces suggests that branching program boosters have the most promise to improve accuracy for applications in which the number of features is large enough that, for example, boosting a decision tree learner is impractical. Also, because branching program boosters divide data into disjoint bins during training, they are likely to be best suited to applications in which training data is plentiful.

The parameter γ used in our constructions is associated with the quality of the weak hypotheses available to the booster. The known noise-tolerant boosting algorithms tolerate noise at rates that do not depend on γ, and the analysis of this paper shows that potential boosters cannot achieve such a guarantee. The still leaves open the possibility that noise at rates depending on γ may still be tolerated. In fact, “smooth” boosting algorithms can tolerate even “malicious” noise at rates that depend on γ [24].

The construction using binary classifiers as weak learners that we used for the experiments in Section 8 is patterned after the simpler construction using confidence-rated weak learners that we analyzed theoretically. It may be possible to perform a theoretical analysis for a related problem with binary weak learners. The main outstanding task appears to be to get a handle on the unattractiveness of the penalizers to the boosting algorithm (for example, to prove nearly matching upper and lower bounds on their contribution to the total potential).

References

1. P. L. Bartlett and M. Traskin. Adaboost is consistent. *Journal of Machine Learning Research*, 8:2347–2368, 2007.
2. J. Bradley and R. Schapire. Filterboost: Regression and classification on large datasets. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.
3. L. Breiman. Arcing the edge. Technical report 486, Department of Statistics, University of California, Berkeley, 1997.
4. L. Breiman. Some infinity theory for predictor ensembles. *Annals of Statistics*, 32(1):1–11, 2004.
5. 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.
6. C. Domingo and O. Watanabe. MadaBoost: a modified version of AdaBoost. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT)*, pages 180–189, 2000.
7. N. Duffy and D. Helmbold. Potential boosters? In *Advances in Neural Information Processing Systems (NIPS)*, pages 258–264, 1999.
8. N. Duffy and D. Helmbold. A geometric approach to leveraging weak learners. *Theoretical Computer Science*, 284:67–108, 2002.
9. Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.
10. Y. Freund. An adaptive version of the boost-by-majority algorithm. *Machine Learning*, 43(3):293–318, 2001.
11. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.
12. Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
13. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 1998.

14. A. Kalai and R. Servedio. Boosting in the presence of noise. *Journal of Computer & System Sciences*, 71(3):266–290, 2005.
15. P. Long and R. Servedio. Martingale boosting. In *Proc. 18th Annual Conference on Learning Theory (COLT)*, pages 79–94, 2005.
16. P. Long and R. Servedio. Adaptive martingale boosting. In *Proc. 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 977–984, 2008.
17. G. Lugosi and N. Vayatis. On the bayes-risk consistency of regularized boosting methods. *Annals of Statistics*, 32(1):30–55, 2004.
18. R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI)*, pages 546–551, 1997.
19. S. Mannor, R. Meir, and T. Zhang. The consistency of greedy algorithms for classification. *Journal of Machine Learning Research*, 4:713–741, 2003.
20. Dragos D. Margineantu and Thomas G. Dietterich. Pruning adaptive boosting. In *Proc. 14th International Conference on Machine Learning*, pages 211–218. Morgan Kaufmann, 1997.
21. L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, pages 512–518, 1999.
22. G. R atsch, T. Onoda, and K.-R. M uller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.
23. R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.
24. R. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.
25. T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–85, 2004.
26. T. Zhang and B. Yu. Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, 33(4):1538–1579, 2005.

Acknowledgements We thank Shai Shalev-Shwartz, Yoram Singer, Nati Srebro and Liwei Wang for stimulating conversations.