Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: http://www.tandfonline.com/loi/uasa20

Boosting in the Presence of Outliers: Adaptive Classification With Nonconvex Loss Functions

Alexander Hanbo Li & Jelena Bradic

To cite this article: Alexander Hanbo Li & Jelena Bradic (2018) Boosting in the Presence of Outliers: Adaptive Classification With Nonconvex Loss Functions, Journal of the American Statistical Association, 113:522, 660-674, DOI: 10.1080/01621459.2016.1273116

To link to this article: https://doi.org/10.1080/01621459.2016.1273116

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2018, VOL. 113, NO. 522, 660–674, Theory and Methods
https://doi.org/10.1080/01621459.2016.1273116

Boosting in the Presence of Outliers: Adaptive Classification With Nonconvex Loss Functions

Alexander Hanbo Li and Jelena Bradic

Department of Mathematics, University of California at San Diego, La Jolla, CA

ABSTRACT

This article examines the role and the efficiency of nonconvex loss functions for binary classification problems. In particular, we investigate how to design adaptive and effective boosting algorithms that are robust to the presence of outliers in the data or to the presence of errors in the observed data labels. We demonstrate that nonconvex losses play an important role for prediction accuracy because of the diminishing gradient properties—the ability of the losses to efficiently adapt to the outlying data. We propose a new boosting framework called *ArchBoost* that uses diminishing gradient property directly and leads to boosting algorithms that are provably robust. Along with the ArchBoost framework, a family of nonconvex losses is proposed, which leads to the new robust boosting algorithms, named *adaptive robust boosting* (ARB). Furthermore, we develop a new breakdown point analysis and a new influence function analysis that demonstrate gains in robustness. Moreover, based only on local curvatures, we establish statistical and optimization properties of the proposed ArchBoost algorithms with highly nonconvex losses. Extensive numerical and real data examples illustrate theoretical properties and reveal advantages over the existing boosting methods when data are perturbed by an adversary or otherwise. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received November 2015
Accepted November 2016

KEYWORDS
Boosting; Breakdown point; Classification; Influence function; Nonconvexity; Robustness

1. Introduction

Recent advances in technologies for cheaper and faster data acquisition and storage have led to an explosive growth of data complexity in a variety of scientific areas such as high-throughput genomics, biomedical imaging, high-energy physics, astronomy and economics. As a result, noise accumulation, experimental variation, and data inhomogeneity have become substantial. However, classification in such settings is known to pose many statistical challenges and hence calls for new methods and theories.

ArchBoost contributes to the literature of binary classification algorithms and boosting algorithms in particular. It applies to a wide range of loss functions including nonconvex losses and is specifically designed to be robust and efficient whenever the labels are recorded with an error or whenever the data are contaminated with outliers. ArchBoost tilts or *arches* down the loss function to adapt to the unknown and unobserved noise in the data by exploring nonconvexity efficiently.

To design the new framework, we will amend the drawbacks of the AdaBoost algorithm (Freund and Schapire 1997) in the contaminated data setting. AdaBoost algorithm is based on an iterative scheme, in which at each stage data are reweighed, and a new weak classifier is found by minimizing the exponential loss. A final estimate of the classification boundary is found by summing up weak classifiers throughout all iterations. AdaBoost's sensitivity to outliers comes from the unbounded weight assignment on the misclassified observations. As outliers are more likely to be misclassified, they are very likely to be assigned large weights and will be repeatedly refitted in the following iterations. This refitting will deteriorate seriously the generalization performance, as the algorithm "learns" incorrect data distribution. To achieve robustness, the algorithm should be able to abandon observations that are on the extreme, incorrect side of the classification boundary. Here, we theoretically and computationally investigate the applicability of nonconvex loss functions for this purpose. We illustrate that the best weight updating rule is to assign a weight of $-\phi'(y_iF(x_i))$ to each data point (x_i, y_i) with $F(x_i)$ denoting the current estimate of the classification boundary. This assignment is only efficient if the loss function ϕ is a nonconvex loss function. We develop a tilting argument for the nonconvex losses. It is shown that, if we use a nonconvex loss, sufficiently tilted, that is, $-\phi'(v)$ is small for all $v \ll 0$, then the outliers are eliminated successively. Hence, a constant "trimming"—typically used in robust statistics—is not sufficient for outlier removal in classification setting. In tilting or "arching" the loss function, we are effectively preserving as much fidelity to the data as possible, while redistributing emphasis to different observations. We propose a new *ArchBoost* framework that implements the above tilting method and adjusts for optimality by a new search of the optimal weak hypothesis. Instead of relying only on gradient descent rules (like LogitBoost or GradientBoost; Friedman 2001), ArchBoost chooses the optimal weak hypothesis that is most orthogonal

CONTACT Jelena Bradic jbradic@ucsd.edu Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JASA.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

to the previous weak hypothesis, therefore improving the most the accuracy of the next iteration.

We propose a sufficient set of conditions needed for a loss function to allow for good properties of the ArchBoost. We show that not every nonconvex function satisfies such conditions; an example is the sigmoid loss. However, we propose a family of loss functions, γ-loss, that balances both the benefits of nonconvexity and the empirical risk interpretation of boosting. Finally, the proposed family of ARB-γ algorithms is widely applicable to a wide variety of problems related to non-Gaussian observations and data that are mislabeled (maliciously or otherwise). We address its robustness and statistical efficiency with details. Although it is straightforward to provide such analysis for parametric linear models, computations for the nonparametric and classification boundaries are far more challenging. We provide novel influence function (Hampel 1974) and finite-sample breakdown point theory (Hampel 1968) that fill in the gap in the existing literature on robustness of the boosting algorithms.

In essence, this article designs a new boosting framework that improves the prediction accuracy for the data observed with errors. In Section 2, we present the new ArchBoost framework. Section 3 outlines a new family of nonconvex losses and presents sufficient conditions for the nonconvex loss to be provably robust. Theoretical analysis of Section 4 contains the numerical and statistical convergence in 4.1 and 4.3, respectively. Moreover, it contains robustness properties in Section 4.2—the influence function in Section 4.2.1 and the new breakdown point in Section 4.2.2 . In Section 5, we demonstrate how to use these methods in practice and illustrate the superiority to the state-of-the art alternatives (see Section 5.1). Section 5.4 deals with the famous Long/Servedio problem for which we show that our ArchBoost method outperforms the RobustBoost (Freund 2009). We also discuss three real datasets in Section 5.6.

2. Methodology of the ArchBoost

Let $\mathcal{X}$ denote a p-dimensional domain, $\mathcal{Y}$ denote the class label set $\{-1, 1\}$, $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be iid data points ($p \le n$), ϕ be a differentiable loss function, and $\mathcal{F}$ be a class of functions from $\mathbb{R}^p$ to $\mathbb{R}$. For any distribution $\mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$, we wish to find $F \in \mathcal{F}$ that minimizes $\mathbb{E}_{\mathbb{P}}[\mathbb{1}\{YF(X) < 0\}]$. With Bartlett, Jordan, and McAuliffe (2006)'s classification-calibration condition on ϕ, this problem is equivalent to finding $F^* \in \mathcal{F}$ that minimizes the ϕ-risk $R_\phi(F) = \mathbb{E}_{\mathbb{P}}[\phi(YF(X))]$. We summarize $F^*(x) = \operatorname{argmin}_{F \in \mathcal{F}} \Phi(F(x)), x \in \mathcal{X}$ where $\Phi(F(x)) := \mathbb{E}[\phi(YF(X))|X = x]$ in Table 1.

AdaBoost (Freund and Schapire 1997) minimizes the empirical ϕ-risk $\hat{R}_{\phi,n}(F) = \frac{1}{n}\sum_{i=1}^n \phi(Y_i F(X_i))$ with the exponential loss, $\phi(v) = e^{-v}$, in a stagewise manner. It approximates the unknown Bayes classifier with a combination of weak classifiers, h_t, obtained by employing a weak learner at each iteration t. It is critical to observe that minimization of the exponential loss by itself is not sufficient to guarantee low generalization error of the AdaBoost (Shapire 2013). Its excellent performance is based on the premise that at each iteration of the algorithm, the method is forced to infer something new about the observations. This amounts to reweighing the observations by a weight vector w, so that the misclassified points gain more weight in the next iteration. However, in the presence of outliers, such methodology will iteratively attempt to refit the outliers to one of the classes and hence effectively pull the decision boundary away from the ground truth. Unfortunately, all convex loss functions will inevitably keep upweighting the persistently misclassified points, and as pointed out by Long and Servedio (2010), they all lead to nonrobust boosting methods. Therefore, new boosting principles need to be designed that allow the loss to be *nonconvex*. *ArchBoost* method, which we propose below, is such a framework that, equipped with nonconvex losses, leads to adaptive and robust algorithms that have provable guarantees. By exploring the nonconvexity, ArchBoost is gradually dropping out the persistent observations from the refitting procedure at each new iteration of the algorithm. In this way, if the observations are consistently being misfit, they are suspected of being outliers and are steadily assigned less importance in the risk minimization procedure. Thus, ArchBoost tilts (i.e., arches) the weight distribution to the nonoutlying observations. As an example of a weight updating rule that is effective at arching, we consider the loss function and the weight function, respectively, as

$$\phi(v) = 4/(1+e^v)^2, \quad w(v) = e^v/(1+e^v)^3 \tag{1}$$

with $v = yF(x)$. To further illustrate this idea, we present graphically (1) in Figure 1, together with the losses and weight distributions of AdaBoost and LogitBoost.

The novel boosting framework *ArchBoost* is presented in Algorithm 1. It iteratively builds an additive model $F_T(x) = \sum_{t=1}^T \alpha_t h_t(x)$ where h_t belongs to some space of weak classifiers denoted by $\mathcal{H}$ (e.g., decision trees). Different from Gradient boost and AdaBoost, ArchBoost finds the optimal weak learner h_t, the step size α_t, and the weight updating vector w_t by exploring the *hardness condition* defined as

$$\mathbb{E}_{w_{t+1}}[Y h_t(X)|X = x] = 0, \tag{2}$$

where $\mathbb{E}_w[g(X,Y)|X=x] := \mathbb{E}[w(X,Y)g(X,Y)|X=x]/\mathbb{E}[w(X,Y)|X=x]$. This condition means that, from iteration t to $t+1$, the weights on $\mathcal{X}$ are updated from w_t to w_{t+1} such that $h_t(X)$ is orthogonal to Y with respect to the inner product

Table 1. The list of commonly used loss functions and its corresponding F^*.

	Population parameters	
Classification method	Loss function $\phi(v)$	Optimal minimizer $F^*(x)$
Logistic	$\log(1+e^{-v})$	$(\log \mathbb{P}(y=1\|x) - \log \mathbb{P}(y=-1\|x))$
Exponential	e^{-v}	$\frac{1}{2}(\log \mathbb{P}(y=1\|x) - \log \mathbb{P}(y=-1\|x))$
Least squares	$(v-1)^2$	$\mathbb{P}(y=1\|x) - \mathbb{P}(y=-1\|x)$
Modified least squares	$[(1-v)_+]^2$	$\mathbb{P}(y=1\|x) - \mathbb{P}(y=-1\|x)$

Algorithm 1 ArchBoost (ϕ)

Given training sample:$(x_1, y_1), \ldots, (x_n, y_n)$ initialize the weights $w_0(x_i, y_i) = 1/n$
for $t = 1, \ldots, T$ **do**
(a) Normalize the weight by assigning $w_t = w_t / \sum_i w_t(x_i, y_i)$
(b) Fit the classifier to obtain a class probability estimate $\mathbb{P}_{w_t}(Y = 1|x) \in [0, 1]$
using current weights w_t on the training data.
(c) Set $h_t(x)$ to be the solution of estimating equation (6).
(d) Find α_t by solving the empirical counterpart of (7).
(e) Set $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$.
(f) Update the weights $w_t = -\phi'(yF_t(x))$.
end for
Output the classifier: sign $(F_T(x))$.

defined on the reweighed data. Thus, the weak hypothesis h_t behaves like a random guess on the reweighed data, and hence, the h_{t+1} will be a good supplement to h_t.

Provided that $\mathcal{F}$ includes all measurable functions, we observe that $F^*(x)$ can be defined by the first-order optimality condition $\mathbb{E}[Y\phi'(YF^*(X))|X = x] = 0$, where ϕ' is defined as the first-order derivative $\frac{d}{dv}\phi(v)$. In classification problems, the parameter v of loss function ϕ is $v = YF(X)$—that is, the margin of a classifier F applied to a data point (X, Y). Rewriting the expectation in terms of the class probabilities, we obtain the following representation of the first-order optimality conditions:

$$\frac{\phi'(-F^*(x))}{\phi'(F^*(x))} = \frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = -1|X = x)}. \tag{3}$$

We aim to mimic equation above in each of the iteration steps of the proposed framework. In more details, at iteration t, with the current estimate $F_{t-1}(x) = h_1(x) + \cdots + h_{t-1}(x)$ at hand, we wish to find a new weak hypothesis $h_t \in \mathcal{H}$, such that $F_t(x) = F_{t-1}(x) + h_t(x)$ with $h_t(x)$ solving the following equation:

$$\frac{\phi'(-F_{t-1}(x) - h_t(x))}{\phi'(F_{t-1}(x) + h_t(x))} = \frac{\mathbb{P}(Y = 1|x)}{\mathbb{P}(Y = -1|x)}. \tag{4}$$

Next, we aim to explore (4) and build an estimating equation to find the optimal h_t. The method of estimating equations is a way of specifying how the optimal h_t should be estimated. This can be thought of as a generalization of many classical methods including the framework of M-estimation. Estimating Equation (4) involves an unknown quantity $\mathbb{P}(Y = 1|x)$. One may substitute $\mathbb{P}$ with $\mathbb{P}_{w_t}$, but this coarse estimation could be very biased, especially when the data have outliers. Therefore, we propose to estimate the right-hand side of (4) by introducing a *bias correction function* $\mathbb{C}_{t-1}(x)$ that depends on both the current estimate F_{t-1} and x, and is such that

$$\frac{\mathbb{P}(Y = 1|x)}{\mathbb{P}(Y = -1|x)} = \mathbb{C}_{t-1}(x)\,\frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)}. \tag{5}$$

Here, the conditional probability $\mathbb{P}_{w_t}(Y = 1|x) := \mathbb{E}_{w_t}[\mathbb{1}_{[Y=1]}|X = x]$. Now, we observe that $\mathbb{P}(Y = 1|x)$ and $\mathbb{P}_{w_t}(Y = 1|x)$ satisfy

$$\begin{aligned}\frac{\phi'(F_{t-1}(x))}{\phi'(-F_{t-1}(x))}\frac{\mathbb{P}(Y = 1|x)}{\mathbb{P}(Y = -1|x)} &= \frac{\mathbb{E}[\mathbb{1}_{[Y=1]}\phi'(YF_{t-1}(X))|x]}{\mathbb{E}[\mathbb{1}_{[Y=-1]}\phi'(YF_{t-1}(X))|x]} \\ &= \frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)}.\end{aligned}$$

Hence, with the bias correction function defined as $\mathbb{C}_{t-1}(x) = \phi'(-F_{t-1}(x))/\phi'(F_{t-1}(x))$, Equations (5) and (4) lead to

$$\frac{\phi'(-F_{t-1}(x) - h_t(x))}{\phi'(F_{t-1}(x) + h_t(x))} = \frac{\phi'(-F_{t-1}(x))}{\phi'(F_{t-1}(x))}\frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)}. \tag{6}$$

Therefore, the estimating equation principle of ArchBoost selects the optimal h_t as a solution to the estimating Equation (6). For the loss function (1), for example, $\mathbb{C}_{t-1}(x) = e^{F_{t-1}(x)}$. Additionally, note that $\mathbb{P}_w$ can always be estimated as long as we use a weak learner that is capable to give class probabilities. One example is decision tree in which case in each terminal region R_j, one can estimate $\mathbb{P}_{w_t}(Y = 1|x)$ by $\sum_{x_i \in R_j, y_i = 1} w(x_i, y_i) / \sum_{x_i \in R_j} w(x_i, y_i)$.

Observe that we can explicitly solve Equation (6) for many commonly used loss functions. For the robust loss (1) in Figure 1, (6) becomes

$$e^{F_{t-1}(x)+h_t(x)} = e^{F_{t-1}(x)}\frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)},$$

leading to $h_t = \log \mathbb{P}_w(Y = 1|x) - \log \mathbb{P}_w(Y = -1|x)$. The results for existing losses are summarized in Table 2. Observe

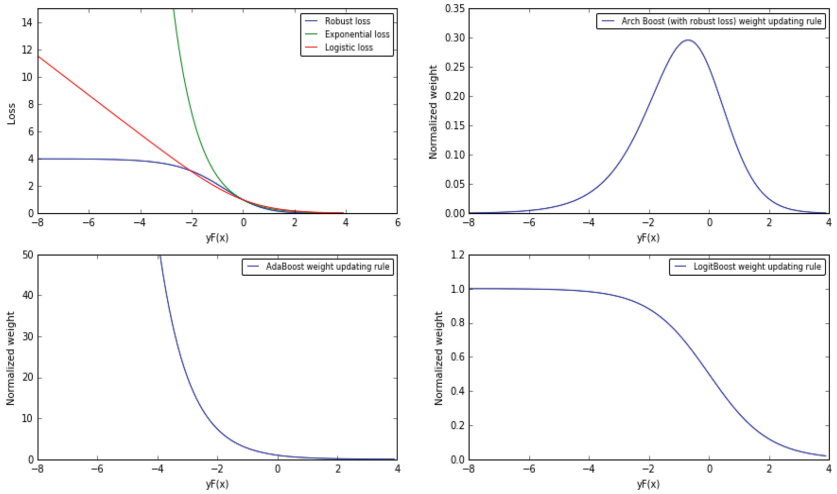


Figure 1. AdaBoost, LogitBoost, ArchBoost loss functions, and the corresponding normalized weight updating rules.

Table 2. The list of commonly used loss functions and their weak hypotheses h.

Classification method	Population parameters	
	Loss function $\phi(v)$	Optimal weak hypotheses $h(x)$
Logistic	$\log(1+e^{-v})$	$\log \mathbb{P}_w(Y=1\|x) - \log \mathbb{P}_w(Y=-1\|x)$
Exponential	e^{-v}	$\frac{1}{2}(\log \mathbb{P}_w(Y=1\|x) - \log \mathbb{P}_w(Y=-1\|x))$
Least squares	$(v-1)^2$	$C(1-F(x))(1+F(x))/(CF(x)+1)$
Modified least squares	$[(1-v)_+]^2$	$C(1-F(x))(1+F(x))/(CF(x)+1)$

NOTE: *$C = \mathbb{P}_w(Y=1|x) - \mathbb{P}_w(Y=-1|x)$.

that for different choices of the weight vector w_t, the resulting h_t changes. The hardness condition works as the guideline of updating the weights w_t. To ensure that $\alpha_t h_t$ indeed decreases the ϕ-risk, we consider an additional line search step

$$\alpha_t = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}}\, \mathbb{E}\Big[\phi\Big(YF_{t-1}(X) + Y\alpha h_t(X)\Big)\Big]. \tag{7}$$

We observe that for

$$w_{t+1}(X,Y) := -\phi'(YF_t(X)), \tag{8}$$

the α_t, (7), satisfies $\mathbb{E}_{w_{t+1}}[Y\alpha_t h_t(X)] \propto \mathbb{E}[-\phi'(YF_{t-1}(X) + Y\alpha_t h_t(X)) \cdot Y\alpha_t h_t(X)] = 0$. For the robust loss (1), $w(v) = e^v/(1+e^v)^3$ is proportional to $-\phi'(v) = 8e^v(1+e^v)^{-3}$ up to a constant. Therefore, by updating weights according to (8), the hardness condition (2) is satisfied.

Finally, we emphasize that throughout the above derivation, we did not put any convexity restriction on the loss function. The only assumption we made is that $\Phi(F(x))$ has only one critical point that is the global minimum, a condition satisfied by many *nonconvex* functions, for example, invex functions of Ben-Israel and Mond (1986). In this way, the ArchBoost algorithm can be applied to a broad family of nonconvex loss functions (see Section 3). Moreover, note that the weak hypotheses of the least-square loss and modified least-square loss (Table 2) depend on the current estimate $F(x)$ and the weighted conditional probability $\mathbb{P}_{w_t}(Y=1|x)$, which is different from that of Gradient boosting (Friedman 2001). Observe that the Gradient boosting effectively fits a least-square method on pseudo-responses (see step 4 of Gradient boost that approximates Equation (9) therein), and hence the optimal weak learner is not chosen robustly. ArchBoost is an improvement as it designs a fully robust algorithm. Moreover, Gradient boost does not define the weights w and hence has a very different viewpoint. Although it can be applied to nonconvex losses using the simple steepest descent, the solution is unstable and the corresponding algorithms using our nonconvex losses (Section 3) behave even worse than LogitBoost.

3. Robust Nonconvex Loss Functions

Not every nonconvex function is a valid candidate for the developed *ArchBoost* method. Any binary classification problem can be written as

$$\min_{v\in\mathbb{R}}\Big[\mathbb{P}(Y=1|x)\phi(v) + \mathbb{P}(Y=-1|x)\phi(-v)\Big], \tag{9}$$

where $v := YF(x)$ is the margin. We assume that (9) has a unique optimal solution in $\mathbb{R}$ for every $x \in \mathcal{X}$. Note that this condition is not equivalent to the convexity of ϕ but rather to the local convexity around the true parameter of interest.

Definition 1. A function ϕ is an *ArchBoosting loss function* if it is differentiable and (i) $\phi(v) \geq 0$ for all $v \in \mathbb{R}$ and $\inf_{v\in\mathbb{R}} \phi(v) = 0$; (ii) for any $0 < \alpha < 1$, $\alpha\phi(v) + (1-\alpha)\phi(-v)$ has only one critical point v^*, which is the global minimum; (iii) for any $0 \leq \alpha \leq 1$ and $\alpha \neq \frac{1}{2}$, $\inf\{\alpha\phi(v) + (1-\alpha)\phi(-v) : v(2\alpha-1) \leq 0\} > \inf\{\alpha\phi(v) + (1-\alpha)\phi(-v) : v \in \mathbb{R}\}$.

Conditions (i) and (iii) together imply that ϕ is an upper bound of the 0–1 loss up to a constant scaling. Condition (iii) is called "classification calibration" (Bartlett, Jordan, and McAuliffe 2006) and is satisfied as long as ϕ is convex, differentiable and $\phi'(0) < 0$. It is considered the weakest possible condition for the resulting classifier to be Bayes-consistent. However, when considering nonconvex losses, the set of regularity conditions does not exists in the current literature.

Lemma 1. All continuously differentiable convex functions $\phi : \mathbb{R} \to \mathbb{R}_+$ such that ϕ' is not a constant satisfy Condition (ii). Moreover, all positive, continuously differentiable functions ϕ such that $\phi'(v) \neq 0$ for all $v \in \mathbb{R}$, satisfy Condition (ii) as long as the function $g : (0,\infty) \to (0,1)$, defined as $g(v) := \phi'(-v)/\phi'(v)$ is strictly increasing and surjective.

By Lemma 1, the logistic, exponential, least-square, and modified least-square losses are all valid ArchBoosting losses. Differentiability of the loss is a noncrucial, technical condition and the hinge loss can be shown to satisfy Conditions (i)–(iii). However, the sigmoid loss $\phi_{\text{sig}}(v) = (1+e^v)^{-1}$ does not satisfy Condition (ii).

Observe that the right-hand side of (4) does not depend on the loss function ϕ and can take values in the positive real line $\mathbb{R}_+$. Hence, we can parameterize it with any strictly increasing surjective function $g : \mathbb{R} \to \mathbb{R}_+$, that is, $\phi'(-v)/\phi'(v) = g(v)$. The classical motivation for reparameterization (McCullagh and Nelder 1989)—often called link functions—is that one uses a parametric representation that has a natural scale matching the desired one. One such function satisfying second part of Lemma 1 is $g(v) = e^{(\gamma-1)v}$ with constant $\gamma > 1$. This parameterization is not unique but it admits a solution to the differential equation $\phi'(-v)/\phi'(v) = e^{(\gamma-1)v}$. The solution (see supplement) is a family of nonconvex losses, which we name γ*-robust losses*,

$$\phi_\gamma(v) = 2^\gamma(1+e^v)^{-\gamma},\ \gamma > 1. \tag{10}$$

We plot the γ-robust losses and the corresponding normalized weight updating functions in Figure 2. Parameter γ is not a tuning parameter, but rather an index of a family of nonconvex losses much like Huber and Tukey's biweight losses. All ϕ_γ are bounded functions ($\leq 2^\gamma$) and hence the effects of the outliers are necessarily bounded. Moreover, the weight updating rules down-weights the largely misclassified data points. When $\gamma = 1$, the weight updating curve is equivalent to the

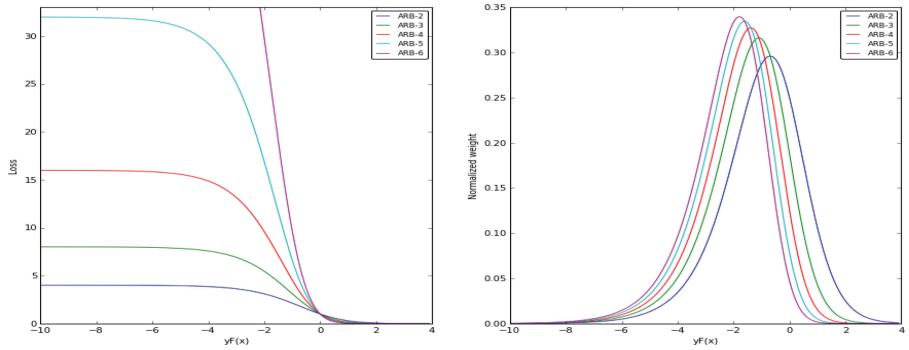


Figure 2. γ-robust losses, ϕ_γ, and the corresponding normalized weight updating rules.

sigmoid loss $\phi(v) = 1 - \tanh(\lambda v)$ when $\lambda = 1/2$ (Mason et al. 1999). Moreover, for $\gamma = 2$, the loss ϕ_2 is similar to the Savage loss $\phi(v) = (1 + e^{2v})^{-2}$ of Mesnadi-Shirazi and Vasconcelos (2009), in which they used the probability elicitation technique. The following Lemma 2 allows us to use ϕ_γ together with the ArchBoost method. The resulting family of robust boosting algorithms, named *Adaptive Robust Boost-γ* (*ARB-γ*), are presented in Algorithm 2.

Lemma 2. For all $\gamma > 1$, ϕ_γ is an ArchBoosting loss function.

Algorithm 2 Adaptive Robust Boost (ARB)-γ

Given: $(x_1, y_1), \ldots, (x_n, y_n)$, initialize the weight vector w_0, for example, $w_0(x_i, y_i) = 1/n$

for $t = 1, \ldots, T$ **do**

(a) Normalize the weight vector $w_t = w_t / \sum_i w_t(x_i, y_i)$

(b) Compute the weak classifier to obtain a class probability estimate $\mathbb{P}_{w_t}(Y = 1|x) \in [0, 1]$, using weights w_t on the training data.

(c) Set $h_t(x) = \log \frac{\mathbb{P}_{w_t}(Y=1|x)}{\mathbb{P}_{w_t}(Y=-1|x)} \in \bar{\mathbb{R}}$.

(d) Find α_t by solving empirical counterpart of (7).

(e) Set $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$

(f) Set $w_{t+1} = e^{yF_t(x)}(1 + e^{yF_t(x)})^{-\gamma-1}$

end for

Output the classifier: $sign\,(F_T(x))$

4. Theoretical Considerations

Despite the substantial body of existing work on boosting classifiers (e.g., Freund 1995; Friedman, Hastie, and Tibshirani 2000; Koltchinskii and Panchenko 2002; Breiman 2004; Zhang and Yu 2005; Bartlett, Jordan, and McAuliffe 2006), research on robust boosting has been limited to methodological proposals with little supporting theory (e.g., Littlestone 1991; Kearns and Li 1993; Gentile and Littlestone 1999; Nock and Lefaucheur 2002; Kalai and Servedio 2005; Rosset 2005; Lutz, Kalisch, and Bühlmann 2008; Bootkrajang and Kaban 2013; Miao et al. 2015; Martinez and Gray 2016).

4.1. Numerical Convergence

In this section, we discuss the numerical convergence of the ArchBoost algorithm whenever the loss ϕ belongs to the class of ArchBoosting loss functions. The main difference from the existing work (e.g., Koltchinskii and Panchenko 2002; Zhang and Yu 2005) is that they used the gradient descent rule in the first article or an approximate minimization in the second one, while we only use the hardness condition to select the weak hypothesis h. Here, $\mathcal{F}^T$ is a set of T-combinations of functions in $\mathcal{H}$, more precisely, $\mathcal{F}^T = \{F : F = \sum_{t=1}^T \alpha_t h_t, \alpha_t \in \mathbb{R}, h_t \in \mathcal{H}\}$. Then every $f \in \cup_{T=1}^\infty \mathcal{F}^T$ can be represented as $\sum_{h \in H_f} \alpha^h h$ for an appropriate subset $\mathcal{H}_f \subset \mathcal{H}$, and its l_1-norm is defined as $\sum_{h \in H_f} |\alpha^{(h)}|$, and its l_2-norm as $\sum_{h \in H_f} \sqrt{|\alpha^{(h)}|^2}$. Finally, let $\{\bar{f}_t\}$ be a sequence of reference functions with empirical risk converging to $R^*_{\phi,n} = \inf_{F \in \cup_{T=1}^\infty \mathcal{F}^T} \hat{R}_{\phi,n}(F)$.

Condition 1. (i) ϕ is Lipschitz differentiable; (ii) $\hat{\mu}(h_t, w_t) = (1/n) \sum_{i=1}^n Y_i h_t(X_i) w_t(X_i, Y_i) \to 0$ as $t \to \infty$; (iii) the step sizes α_t satisfy

$$\sum_{t=1}^\infty \alpha_t = \infty, \quad \sum_{t=1}^\infty \alpha_t^2 < \infty, \quad \sum_{t=1}^\infty \frac{\alpha_{t+1} \xi_t \log t}{t^{c_t}} < \infty,$$

for some $\xi_t = o(1)$, $\xi_t \geq 0$; (iv) $\bar{f}_t$ satisfies $||\bar{f}_t - F_t||_1 = o(\log t)$, $||\bar{f}_t - F_t||_2^2 \leq \frac{||\bar{f}_t - F_t||_1^2}{t^{c_t}}$ where $c_t \to 0$ and $t^{c_t} \to 1$ as $t \to \infty$.

Theorem 1. Let ϕ be an ArchBoosting loss function and assume the weak learner is able to divide the domain $\mathcal{X}$ into disjoint regions and give the class probability estimations (e.g., decision tree). Let F_T be the ArchBoost classifier, then $\hat{R}_{\phi,n}(F_T)$ will converge in $\mathbb{R}$ as $T \to \infty$. In addition, under Condition 1, $\hat{R}_{\phi,n}(F_T) \to R^*_{\phi,n}$ as $T \to \infty$.

Unlike existing results, Theorem 1 does not require any additional algorithmic tuning parameters (see Theorem 3.1 of Zhang and Yu 2005 and choices of ε_t, Λ_t). It is worth mentioning again that the proof techniques in the existing literatures do not extend to nonconvex losses. We bridge the gap by developing new analysis. Results in Bartlett and Traskin (2007) (e.g., Theorem 6) hold under an assumption of a positive lower bound on the Hessian of the empirical risk, which is strictly violated by any nonconvex loss. Furthermore, Theorem 1 allows the approximate minimization step (7) to be inexact (by contrast, see Theorem 6 of Bartlett and Traskin 2007).

Remark 1. The reference sequence $\{\bar{f}_t\}$ needs to be in a local neighborhood of F_t. For all $\bar{f}_t$ such that $||\bar{f}_t||_1 = o(\log(t))$, the condition further reduces to $||\bar{f}_t - F_t||_1 \leq ||\bar{f}_t - 0||_1$, that is, the distance between $\bar{f}_t$ and F_t is smaller than the distance between

$\bar{f}_t$ and a random guess. This can be achieved by shrinking the step sizes α_t at a constant rate over every iteration. Moreover, the effects of the second constraint regarding $\bar{f}_t$ can be explained as a nonsparsity assumption on the difference between F_t and $\bar{f}_t$, and is asymptotically negligible because $t^{c_t} \to 1$ when $t \to \infty$, which leads to the trivial inequality between l_1 and l_2 norms.

Remark 2. The classical conditions that are guarding against infinitely small step sizes are now supplemented with an additional constraint $\sum_{t=1}^{\infty} t^{-c_t}\alpha_{t+1}\xi_t \log t < \infty$. For example, if $\xi_t = O(t^{-1})$, then we can choose $\alpha_t = O(t^{-b-c_t})$ where b is any positive constant and c_t can converge to 0 at any speed. However, if $\xi_t = O((\log t)^{-1})$, we need $c_t \to 0$ slowly (e.g., $O((\log\log t)^{-1})$) and α_t can be chosen as $O(t^{-1})$. The additional constraint on the step size choice acts as a penalty on allowing nonconvex loss functions (Zhang and Yu 2005).

4.2. Robustness

In this section, we quantify and justify the robustness of ArchBoost Algorithm 1 through the point of view of the influence function, as well as that of the finite-sample breakdown point.

4.2.1. Influence Function

The richest quantitative robustness measure is provided by the influence function (Hampel 1974) $u \to IF(u; T, G)$ of T at G. It is defined as the first Gâteaux derivative of a functional T at a distribution $\mathbb{P}$, that is, $IF(z; T, \mathbb{P}) = \lim_{\epsilon\to 0^+}[T((1-\epsilon)\mathbb{P} + \epsilon\Delta_z) - T(\mathbb{P})]/\epsilon$, where Δ_z is the Dirac distribution at the point z such that $\Delta_z(\{z\}) = 1$. It gives the effect that an outlying observation may have on an estimator. To simplify the analysis, we consider a subclass of binary classification models, in which the true boundary F^* is assumed to belong to a class of functions H. Here, H is defined as a reproducing kernel Hilbert space (RKHS) with a bounded kernel k and the induced norm $||\cdot||_H$. Observe that ArchBoost is consistent only if it is properly regularized (stopped after a certain number of steps; see Theorem 5). Hence, to study its robustness properties we consider a regularized criterion

$$f_{\mathbb{P},\lambda} = \operatorname*{argmin}_{f\in H}\left\{\mathbb{E}_{\mathbb{P}}\left[\phi(Y, f(X))\right] + \lambda||f||_H^2\right\}.$$

The loss ϕ is a function of tuple $(Y, f(X))$ only for convenience of analysis.

The feature map is $\Psi : \mathcal{X} \to H$ with $\Psi(x) = k(x, \cdot)$.

Theorem 2. The influence function of $f_{\mathbb{P},\lambda}$ takes the form $IF(z; T, \mathbb{P}) = -S^{-1} \circ J$, where $\circ$ is defined to mean S^{-1} acting on J and operators $S : H \to H$ and $J \in H$ are defined as $S = \mathbb{E}_{\mathbb{P}}[\phi''(Y, f_{\mathbb{P},\lambda}(X))\langle\Psi(X), \cdot\rangle\Psi(X)] + 2\lambda id_H$, $J = \phi'(z_y, f_{\mathbb{P},\lambda}(z_x))\Psi(z_x) - \mathbb{E}_{\mathbb{P}}[\phi'(Y, f_{\mathbb{P},\lambda}(X))\Psi(X)]$, where $id_H : H \to H$ is the identity mapping and $z = (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$ is the contamination point. In the above display, the derivative is defined as $\phi'(u, v) := \frac{\partial}{\partial v}\phi(u, v)$.

For a nonconvex loss function ϕ, ϕ'' is not guaranteed to be nonnegative. However, we show that it is sufficient to have the nonnegativity of the expectation (locally around F^*) rather than of the second derivative itself.

Lemma 3. For a binary classification problem, given any distribution $\mathbb{P}$, whenever ϕ is a twice continuous differentiable ArchBoosting loss function, then $\mathbb{E}_{\mathbb{P}}[\phi''(Y, F^*(X))q^2(X)] \geq 0$ for any measurable function $q : \mathcal{X} \to \mathbb{R}$. Furthermore, if $\mathbb{P}$ and $\mathcal{X}$ are such that $\mathbb{P}(Y = 1|X = x) \in [\delta, 1-\delta]$ for some $0 < \delta < \frac{1}{2}$, and if $p\phi''(1, v_p^*) + (1-p)\phi''(-1, v_p^*) > 0$ at the global minimum v_p^* for all $p \in [\delta, 1-\delta]$, then there exists $r > 0$ such that $\mathbb{E}_{\mathbb{P}}[\phi''(Y, G(X))q^2(X)] \geq 0$ for all measurable function G with $||G - F^*||_\infty < r$.

Conditions of the above lemma are satisfied for all γ-robust loss function. With $\gamma = 2$ and any x, $\mathbb{E}_Y[\phi''(Y, F^*(X))q^2(X)|X = x] = 2p_x^2(1-p_x)^2q^2(x) \geq 0$ where $p_x = \mathbb{P}(Y = 1|X = x)$. Thus, $\mathbb{E}_{\mathbb{P}}\phi''(Y, F^*(X))q^2(X) \geq 0$. Furthermore, if $p_x \in [\delta, 1-\delta]$ for some $\delta \in (0, \frac{1}{2})$, then $p_x\phi''(1, F^*(x)) + (1-p_x)\phi''(-1, F^*(x)) = 2p_x^2(1-p_x)^2 \geq 2\delta^2(1-\delta)^2 > 0$ for all $p_x \in [\delta, 1-\delta]$. (Observe that the condition of $p_x \in [\delta, 1-\delta]$ for some $\delta \in (0, \frac{1}{2})$ restricts our setting to the "low-noise" setting where the true probability of the class membership is bounded away from 0 or 1.)

Theorem 3. For a binary classification problem, let $\phi : \mathbb{R} \to [0, \infty)$ be a twice continuously differentiable ArchBoosting loss function and let H be an RKHS with bounded kernel k. Assume $\mathbb{P}$ is a distribution on $\mathcal{X} \times \mathcal{Y}$ such that for all $x \in \mathcal{X}$, $\mathbb{P}(Y = 1|X = x) \in [\delta, 1-\delta]$ for some $0 < \delta < \frac{1}{2}$, and $p\phi''(1, v_p^*) + (1-p)\phi''(-1, v_p^*) > 0$ at the global minimum v_p^* for all $p \in [\delta, 1-\delta]$. Then there exists $r > 0$ such that for all $||f_{\mathbb{P},\lambda} - F^*||_\infty < r$,

$$||IF(z; f_{\mathbb{P},\lambda}, \mathbb{P})||_H \leq \sqrt{\frac{C_\phi}{\lambda}} + \frac{M_k|\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))|}{2\lambda}, \qquad (11)$$

where M_k is the upper bound of the kernel k and $C_\phi = \phi(0, 0)$.

Theorem 3 shows that the robustness mainly comes from the diminishing property of $|\phi'|$. In fact, for any nonconvex ArchBoosting loss function, due to Assumption 2, we have $|\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \to 0$ when $|z_y f(z_x)| \to \infty$. If we plot $\|IF(z; f_{\mathbb{P},\lambda}, \mathbb{P})\|_H$ versus $z_y f_{\mathbb{P},\lambda}(z_x)$, then it will decrease toward a constant far from the origin, much alike the redescending M-estimators. Moreover, Theorem 3 implies that $\|IF(z; f_{\mathbb{P},\lambda}, \mathbb{P})\|_H$ is unbounded for the exponential loss (AdaBoost), bounded but not diminishing for the logistic loss (LogitBoost) and diminishing for the γ-robust losses (ArchBoost).

4.2.2. Breakdown Point

Empirical robustness property defined as breakdown point by Donoho and Huber (1983) has proved most successful in the context of location, scale, and regression problems (e.g., Rousseeuw 1984; Stromberg and Ruppert 1992; Tyler 1994). This success has sparked many attempts to extend the concept to other situations (e.g., Ruckstuhl and Welsh 2001; Genton and Lucas 2003; Davies and Gather 2005). However, very little work has been done in the classification context. The breakdown point, as defined by Hampel (1968), is roughly the smallest amount of contamination that may cause an estimator to take on arbitrarily large aberrant values. The breakdown points of $1/n$ for the mean and $1/2$ for the median

do reflect their finite-sample behavior. However, an alternative view is desired in the classification context as the magnitude of an estimator may not relate to necessarily bad classification—that is, the size of the weak hypothesis is not crucially related to the classification boundary. Instead, in the context of boosting, we look for the estimator that keeps the gradient of the risk minimization in the *oracle direction*. The meaning of oracle direction will be further explained in Remark 3. To that end, let $\mathcal{S}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be a set of observed, contaminated samples among which $\mathcal{O}_{m:n} = \{(X_{m+1}, Y_{m+1}), \ldots, (X_n, Y_n)\}$ being a set of outliers. Let h_t be the weak hypothesis and denote the vectors $\mathbf{h}_t = (h_t(X_1), \ldots, h_t(X_n))$. Let $-\mathbf{g}_t = (-g_t(X_1), \ldots, -g_t(X_n))$ stands for the negative gradient of the empirical risk $\hat{R}_{\phi,n}$ on $\mathcal{S}_n$, whereas $-\mathbf{g}_o = (-g_t(X_1), \ldots, -g_t(X_m), 0, \ldots, 0)$ is the embedding of the negative gradient of the empirical risk on the sample without outliers $\mathcal{S}_n \setminus \mathcal{O}_{m:n}$ into $\mathbb{R}^n$.

Theorem 4. For every region R^j, define $\eta_j := |p_j - \frac{1}{2}|/\min(p_j, 1-p_j)$, where $p_j \in (0,1)$ and $p_j \neq \frac{1}{2}$. Then at iteration t, if any ArchBoost algorithm, conditional on the realizations $\{(X_i, Y_i) = (x_i, y_i)\}_{i=1}^n$, satisfies that for all R_j,

$$\sum_{i:x_i \in \mathcal{O}_{m:n} \cap R^j} w_t(x_i, y_i) \leq \eta_j \sum_{i:x_i \in R^j \setminus \mathcal{O}_{m:n}} w_t(x_i, y_i), \tag{12}$$

then the gradient descent direction is preserved, that is, $-\langle \mathbf{g}_o, \mathbf{h}_t \rangle \geq 0$.

Conditions of the above theorem are very mild. Theorem 4 suggests that any ArchBoost algorithm that satisfies the above conditions preserves the descending direction of the noncontaminated empirical ϕ-risk, hence it minimizes the oracle risk while disregarding the outliers.

Remark 3. Theorem 4 establishes that whenever (12) holds $\mathbf{h}$ will have a direction along which the oracle empirical risk of the noncontaminated data decreases. Figure 2 clearly illustrates that (12) is more likely to be satisfied for the ARB-γ than for the AdaBoost or the LogitBoost algorithm. For example, if $y_i = -1$ and $\mathbb{P}(Y = -1|X = x_i) = 0.001$, then for Real AdaBoost, $w(x_i, y_i)/w_b \simeq 32$, and for ARB-2, $w(x_i, y_i)/w_b \simeq 0.008$ where w_b is the weight for a data point (x_b, y_b) such that $F^*(x_b) = 0$. It can be seen that AdaBoost puts 4000 times more weight on this outlier data than ARB-2, and hence violates (12).

4.3. Statistical Consistency

Consistency of gradient boosting algorithms has been established in the case of convex losses (Jiang 2004; Zhang and Yu 2005; Bartlett, Jordan, and McAuliffe 2006). We develop the consistency for bounded nonconvex loss functions by exploring local curvatures. By dropping convexity requirement, we add the boundedness condition on the loss functions, and hence our theory is exclusively for bounded nonconvex functions.

Condition 2. Let the class of weak hypothesis $\mathcal{H}$ satisfy $\lim_{T\to\infty} \inf_{f\in\mathcal{F}^T} R_\phi(f) = R_\phi^*$ for a VC-dimension $d_{\mathrm{VC}}\{\mathcal{H}\} < \infty$. Moreover, the function ϕ is a decreasing ArchBoosting loss function that is also bounded and Lipschitz.

For a rich class $\mathcal{H}$, the first part of Condition 2 is true (Bartlett and Traskin 2007). The class $\mathcal{T}$ of binary trees with the number of terminal nodes larger or equal to $d+1$, where d is the dimension of $\mathcal{X}$ (Breiman 2004) satisfies it. If a loss function ϕ satisfies the second part of this condition, then both $\lim_{v\to\infty} \phi(v)$ and $\lim_{v\to-\infty} \phi(v)$ exist in $\mathbb{R}$, and the first derivative converges to zero away from the origin. This lessens the effect of gross outliers and in turn leads to good robust properties of the resulting estimator.

Theorem 5. Let L_ϕ and M_ϕ be the Lipschitz constant and the maximum value of ϕ, respectively. Let $V = d_{\mathrm{VC}}(\mathcal{H})$, $c = 24\int_0^1 \sqrt{\log\frac{8e}{\mu^2}}\,d\mu$. Then, under Condition 2, (a) for sequences $T_n, \zeta_n \to \infty$ and $\delta_n \to 0$ as $n \to \infty$, there exists a sequence $E_n(\zeta_n) \to 0$ such that, with probability at least $1-\delta_n$,

$$\sup_{f\in\mathcal{F}^{T_n}} |\hat{R}_{\phi,n}(f) - R_\phi(f)| \leq c\zeta_n L_\phi \sqrt{\frac{(V+1)(T_n+1)\log_2(\frac{2(T_n+1)}{\log 2})}{n}} + M_\phi\sqrt{\frac{\log\frac{1}{\delta_n}}{2n}} + E_n;$$

(b) $\sup_{f\in\mathcal{F}^{T_n}} |\hat{R}_{\phi,n}(f) - R_\phi(f)| \to 0$ a.s. if $T_n = n^{1-\varepsilon}$, $\varepsilon \in (0,1)$; (c) with the same T_n, $R_\phi(f_n^*) \to R_\phi^*$ a.s. where $f_n^* = \operatorname{argmin}_{f\in\mathcal{F}^{T_n}} R_{\phi,n}(f)$.

Theorem 5 illustrates the uniform deviation between the ϕ-risk and the empirical ϕ-risk. Note that we want $T_n \to \infty$ as $n \to \infty$ but not too fast (slower than $\mathcal{O}(n)$). Moreover, from part (b), there exists a sequence of samples $\{S_n^*\}_{n=1}^\infty$ such that $R_\phi(\tilde{f}_n) \to R_\phi^*$ as $n \to \infty$. Here, $\tilde{f}_n$ is the optimal classifier obtained by minimizing the empirical risk on S_n^*. Given any sample $\mathcal{S}_n$, the misclassification error of any classifier f on $\mathcal{S}_n$ is $L(f) = \mathbb{P}(f(X) \neq Y|\mathcal{S}_n)$. The Bayes risk is then defined as $L^* = \inf_{f\in\mathcal{M}} L(f) = \mathbb{E}_X[\min(\eta(X), 1-\eta(X))]$, where $\eta(X) = \mathbb{P}(Y = 1|X)$ and $\mathcal{M}$ stands for the family of all measurable functions. Next we state the intermediary lemma that connects the reference sequence $\tilde{f}_n$ to the ArchBoost estimator F_{T_n}.

Lemma 4. For the above reference sequence $\{\tilde{f}_n\}_{n=1}^\infty$ and a nonnegative sequences $T_n = n^{1-\varepsilon}$, $\varepsilon \in (0,1)$, and with the choice of α_t as in Theorem 1, we have as $n \to \infty$, (a) $(\hat{R}_{\phi,n}(\tilde{f}_n) - R_\phi(\tilde{f}_n))_+ \to 0$ a.s. and (b) $(\hat{R}_{\phi,n}(F_{T_n}) - \hat{R}_{\phi,n}(\tilde{f}_n))_+ \to 0$ a.s.

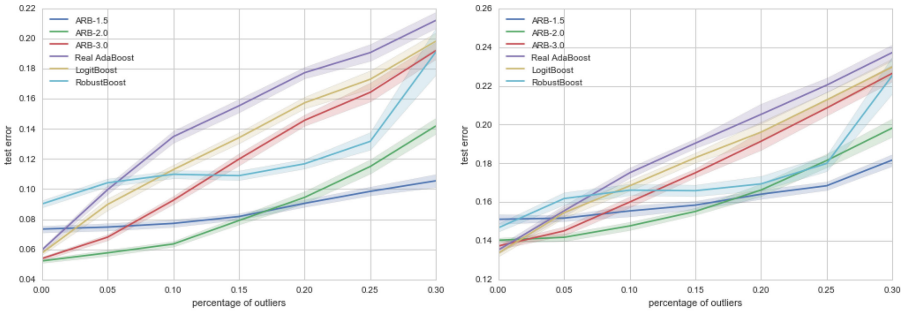
Theorem 6. Assuming conditions of Theorem 5 hold. Then, with the stopping time T_n as in Theorem 5 and the step size α_t as in Theorem 1, the ArchBoost classifier F_{T_n} satisfies $L(\operatorname{sign}(F_{T_n})) \to L^*$ a.s. as $n \to \infty$.

5. Numerical Experiments

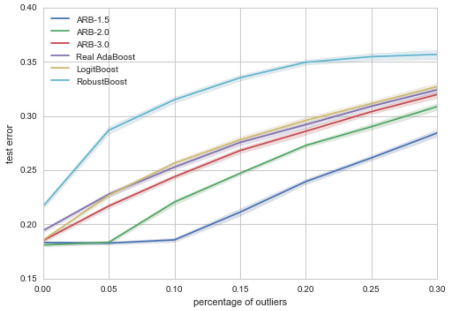
In this section, we provide an extensive simulation and real data analysis illustrating superior performance of the ArchBoost framework and ARB-γ algorithms in particular.

5.1. Gaussian–Student Mixture

In this section, design $X \sim \mathcal{N}(0, \Sigma_p)$, and we define the elliptical boundary according to the median of $\|X\|_2^2$, that is, $Y = 1$



(a) $p = 10$, $\Sigma = I_{10}$ (b) $p = 10$, $\Sigma =$ Toeplitz



(c) $p = 100$, $\Sigma =$ Toeplitz

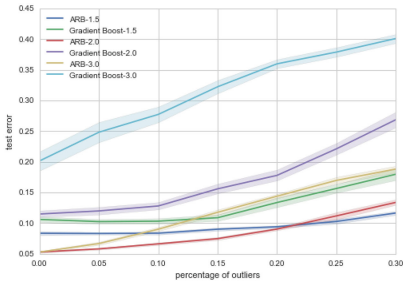
Figure 3. Comparison of average test errors of ARB-γ, AdaBoost, and LogitBoost.

if and only if $X^T\Sigma^{-1}X \geq \text{median}(\mathcal{X}_p^2)$. In the first example, $p = 10$ and $\Sigma_{10} = I_{10}$ with $n = 12{,}000$ and 2000 of them are used as a training sample (Hastie et al. 2005). In the second and third example, we let $[\Sigma_p]_{ij} = (0.3)^{|i-j|}$ be a Toeplitz matrix with $p = 10$ and $p = 100$, respectively. In the third example, $n = 36{,}000$ with 6000 used for training. In all experiments, we use fivefold cross-validation and use decision tree as the weak learner with the tree depth set to be 1 (decision tree stump) for $p = 10$ and tuned to be 3 when $p = 100$. For RobustBoost, the maximum stopping times are set to be 1000 when $p = 10$, and 3000 when $p = 100$. Additional noise in observations is generated from t-distribution with 4 degrees of freedom and with correlation structure that parallels the one of X.

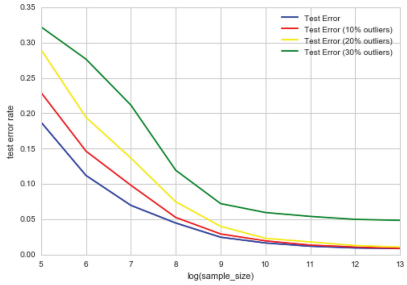
Figure 3 implies several observations. First, the test errors of the ARB-γ algorithms are all less than that of the Real AdaBoost or LogitBoost for correlated and uncorrelated feature space and low and higher dimensional problems. Second, when the percentage of outliers is small, the performances of ARB-2 are the best. When the noise level is higher, ARB-1.5 behaves the best. Hence, if we were to "tune" γ for ARB-γ algorithms, for example, choose ARB-2 when noise level is less than 25% and ARB-1.5 otherwise in Figure 3(a), then ARB-γ is uniformly better than both AdaBoost and LogitBoost. At last, the performances of ARB-γ is very similar to the performance of AdaBoost or LogisticBoost when γ gets larger, allowing certain flexibility in the hardness of the robustness belief. If one is more certain of the cleanliness of the data, larger γ may provide a compromise between robustness and nonrobustness. Therefore, in practice, we recommend to choose γ to be 1.5 or 2. Choosing γ too large will depress the robustness of the algorithm, and choosing γ too close to 1 will lead to unnecessary instability.

5.2. Comparison with Nonconvex Gradient Boost

To illustrate that nonconvexity is not the only feature that enables ArchBoost to have great performance, we showcase that it behaves much better than the Gradient boost with a 1.5-robust loss function (10). It is worth pointing out that such Gradient boosting must be implemented using steepest descent methods and that nonconvexity of the loss leads to high instability of estimates over iterations. We contrast



(a) ARB-γ and Gradient Boost with γ-robust loss.



(b) Consistency of ARB-γ

Figure 4. Comparisons with nonconvex gradient boost and consistency.

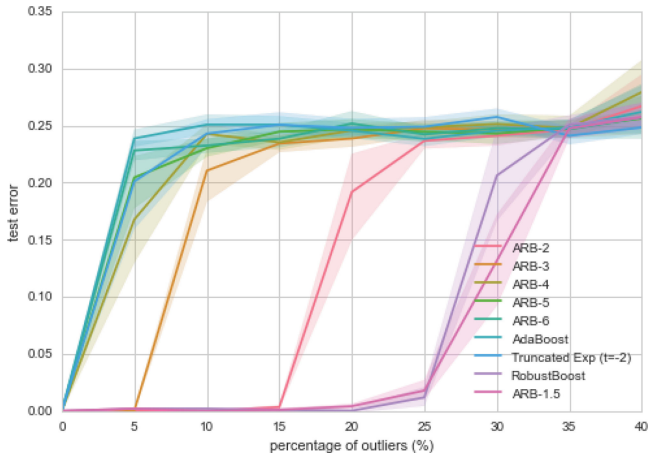


Figure 5. Comparison of ARB-γ on Long/Servedio problem with different ϵ.

the methods by generating samples from the model as in Figure 3(a).

From Figure 5(a), we immediately observe that for every choice of γ, the ARB-γ achieves lower test error than the corresponding Gradient boost with γ-robust loss with the difference being larger for larger number of outliers and larger γ. We observe that similarly as before ARB-2 achieves smallest error (5%,9%) if the percentage of outliers is smaller than 20% whereas ARB-1.5 achieves smallest error (9%,11%) if the percentage of outliers is larger than 20%. The corresponding test errors for Gradient boost with 2 and 1.5-robust loss are much higher (ranging from (11%,19%) to (21%,35%), respectively).

5.3. Consistency

To show consistency of the proposed ArchBoost algorithms, we generate iid data from the model as in Figure 4(a) but now varying sample sizes $\exp(k) + 20{,}000$, for $k = 5, 6, \ldots, 13$. Then we use $\exp(k)$ data for training and the rest 20,000 for testing. In Figure 4(b), we can see that the test error is indeed decreasing to 0 for various percentages of outliers. The higher the number of outliers, the larger the sample size n should be for the algorithm to converge. This is not unexpected as the outliers are effectively eating up (shrinking) the sample size (the algorithm is discarding them successively in each iteration).

5.4. The Long/Servedio Problem

Long and Servedio (2010) constructed a challenging experiment with $X \in \mathbb{R}^{21}$ with binary features $X_i \in \{-1, +1\}$ and label $y_i \in \{-1, +1\}$. First, the label y is chosen to be -1 or $+1$ with equal probability. Then for any given y, the features X_i are generated according to the following mixture distribution: *Large margin:* With probability $\frac{1}{4}$, set $X_i = y$ for all $1 \le i \le 21$. *Pullers:* With probability $\frac{1}{4}$, set $X_i = y$ for $1 \le i \le 11$ and $X_i = -y$ for $12 \le i \le 21$. *Penalizers:* With probability $\frac{1}{2}$, randomly choose 5 coordinates from the first 11 features and 6 from the last 10 to be equal to y. The remaining features are set to $-y$. We generate 800 samples and flip each label with probability $\epsilon \in [0, 0.5)$. The data from this distribution can be perfectly classified by $\text{sign}(\sum_i X_i)$. The classifiers are trained using the noisy data and tested on the original clean data (Freund 2009). In total, 20 datasets are generated, and on each of them, 10% of the labels were flipped. Stopping times of the algorithms are $T \le 800$. The average test errors and sample deviations are reported in Table 3, from which we conclude that the ARB-2 outperforms Real AdaBoost and LogitBoost, and is even better than RobustBoost (target parameter $\theta = 0.15$).

Figure 5 shows the average test errors and the 95% confidence intervals of different ARB-γ algorithms. The conclusion is that ARB-1.5 behaves uniformly better than all the other algorithms. The breakdown point will get higher when $\gamma \to 1^+$, implying that smaller γ leads to better robustness properties.

5.5. Outlier Detection

In this experiment, we generate 2000 data points as in Section 5.1, and add noise to the first ϵ percentage. After 800 iterations, we record the times that each data point is misclassified, and count how many of the points that are misclassified more than 600 times (denoted as T) actually belong to the noisy set (denoted as T_o). The ratio T_o/T and the results are shown in Table 4. When the percentage of outliers is less than 15%, for the ARB-2, more than 99% of the points that have been misclassified for more than 600 times are indeed the outliers, but for the Real AdaBoost, this number is only around 31%. Informally, for ARB-2, when $\epsilon \le 15\%$, we have more than 99% "confidence" to conclude that a data point, which is misclassified for more than 600 times, is indeed an outlier.

5.6. Real Data Application

We consider the Wisconsin (diagnostic) breast cancer data (*https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)*) with ten real-valued features computed for each cell nucleus (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension) for 569 individuals, with 357 benign and 212 malignant cells.

The training set has 150 benign samples and 150 malignant samples, randomly obtained. The maximum stopping time is set to be 200. We use tree stump as the weak learner in all three problems. Results are reported in Table 5 and in Figure 6. Observe that ARB-2 behaves the best on the original dataset, and ARB-1.5 outperforms others in the presence of noise. Compared to Stefanski, Wu, and White (2014) and their test error rate of 4%, our methods uniformly achieve smaller and comparable test error rates on the clean and perturbed datasets.

Next, we consider a dataset that is part of the "MicroArray quality control II" project with accession number GSE20194 (*http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE20194*).

The dataset contains 278 newly diagnosed breast cancer patients, aged from 26 to 79 years spanning all three major races and their mixtures. Estrogen-receptor status helps guide treatment for breast cancer patients. Of 278 patients, 164 had positive estrogen-receptor status (PERS) and 114 have negative estrogen-receptor status (NERS). Each sample includes 22,283 biomarker probe-sets.

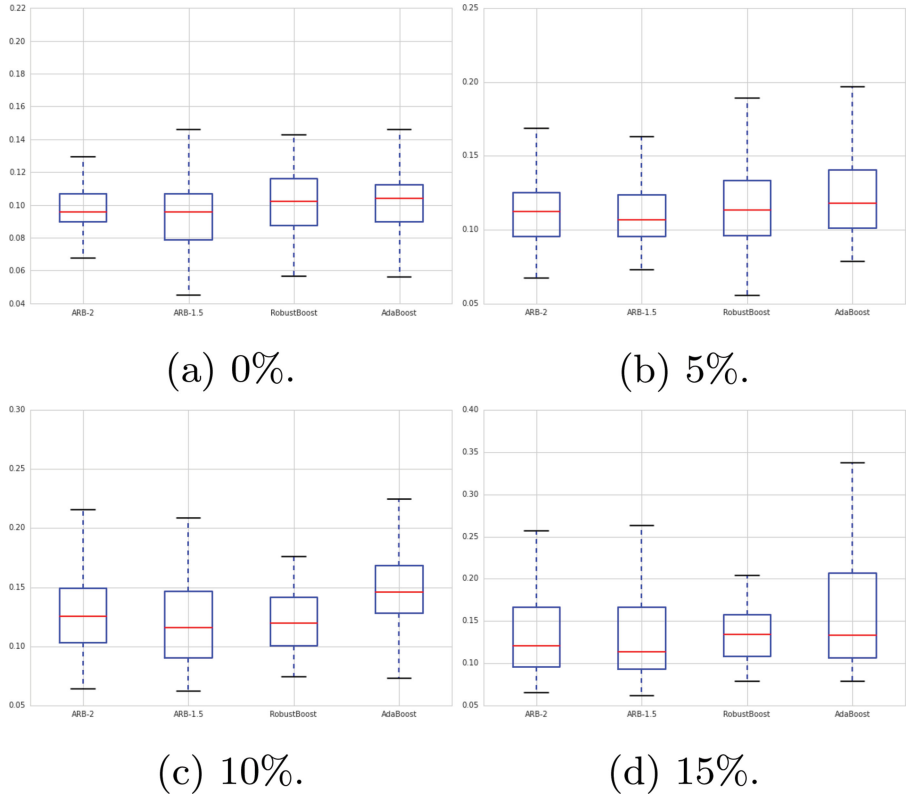


Figure 6. Comparison of ARB-2, ARB-1.5, RobustBoost, and Real AdaBoost on the GSE20194 gene dataset, from left to right are the boxplots for the test errors of ARB-2, ARB-1.5, RobustBoost, and AdaBoost.

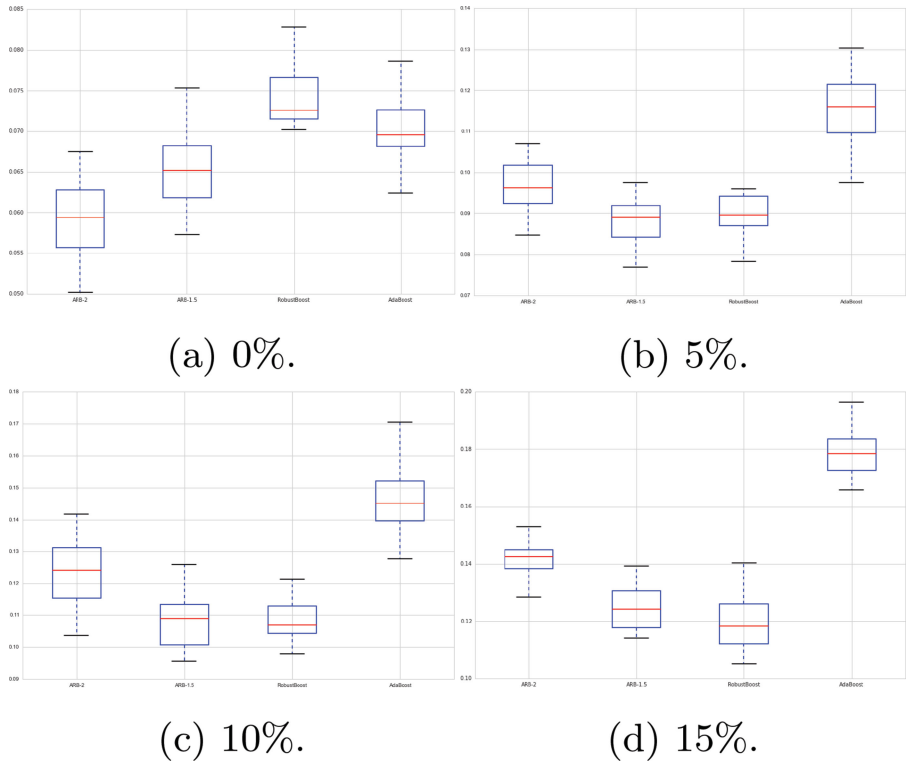
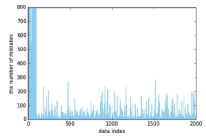
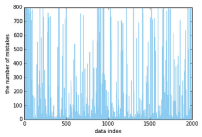
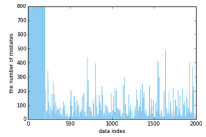
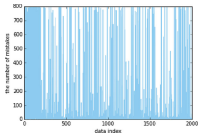
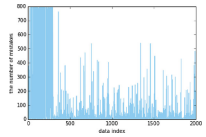
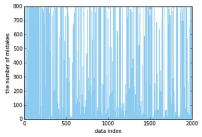
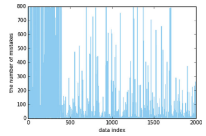
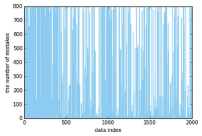


Figure 7. Comparison of ARB-2, ARB-1.5, RobustBoost, and Real AdaBoost on the UCI sensorless drive diagnosis dataset. In each subfigure, from left to right are the boxplots for the test errors of ARB-2, ARB-1.5, RobustBoost, and AdaBoost.

Table 3. Long/Servedio problem.

Data type	Real AdaBoost	LogitBoost	RobustBoost ($\theta = 0.15$)	ARB-2
noise($\epsilon = 0.1$)	28.24%(1.53%)	26.61%(1.51%)	11.04%(0.67%)	**9.82**%(0.43%)
clean	25.07%(1.92%)	22.59%(1.74%)	0.21%(0.35%)	**0.02**%(0.04%)

Table 4. Outliers detection. The *x*-axis stands for the index of the training points ranging from 1 to 2000, and the *y*-axis stands for the times a point is misclassified, ranging from 0 to 800.

ϵ	ARB-2	AdaBoost
0.05		
T_o/T	100%	30.49%
0.15		
T_o/T	99.04%	37.38%
0.1		
T_o/T	100%	32.22%
0.2		
T_o/T	85.48%	44.40%

We choose 3000 probe-sets with the smallest p-values in the two-sample t-test (e.g., Zhang et al. 2014). We randomly choose 50 samples with PERS and 50 samples with NERS for a training set. Then the labels of the training samples are randomly flipped. The stopping time is set to be at most 100. Results are summarized in Table 6 and in Figure 6. The best test errors of 15% and 9% were achieved by Deshwar and Morris (2014) and Zhang et al. (2014), respectively. However, our methods achieve errors comparable to those even when the labels were randomly perturbed.

Finally, we compare ARB-2, ARB-1.5, RobustBoost, and Real AdaBoost on the sensorless drive diagnosis dataset (*https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis*).

We have 58,509 samples and each with 49 features and 11 different classes; 14,000 points are chosen and then from these, 2000 are used for training and 2000 for validation. The stopping times are set ≤ 3000. The test errors on clean data are summarized in Table 7 and Figure 7. RobustBoost behaves worse and the best for 10% or 15% and 0% of the labels flipped, respectively. With higher levels of the noise, the test errors of ARB-1.5 and RobustBoost are very close with ARB-1.5 not needing to fine tune any target parameters.

5.7. Literature Review

There have been considerable efforts focused on designing methods that adapt to the error in the data: outliers and/or mislabeling of the observations. In the existing work, algorithms of Grünwald and Dawid (2004) achieve provable guarantees (Kanamori et al. 2007; Natarajan et al. 2013) when contamination model (Blanchard et al. 2016) is known or when multiple

Table 5. Comparison of the average test errors and sample deviation (over 100 repetitions and using five-fold cross-validation) of four algorithms on the Wisconsin breast cancer dataset.

Percentage of flipped labels	Methods		
	ARB-2	ARB-1.5	RobustBoost
0%	3.47%(1.41%)	**3.43**%(1.34%)	4.71%(1.70%)
5%	4.80%(1.79%)	**4.47**%(1.75%)	4.82%(1.66%)
10%	5.85%(1.82%)	**5.11**%(1.79%)	5.44%(1.81%)
15%	6.67%(2.18%)	**5.92**%(2.22%)	6.53%(2.20%)
	GradientBoost-1.5	LogitBoost	AdaBoost
0%	5.44%(1.76%)	4.82%(1.85%)	4.06%(1.58%)
5%	6.29%(1.81%)	5.64%(1.97%)	5.43%(2.04%)
10%	7.34%(1.99%)	6.19%(1.81%)	6.33%(1.85%)
15%	8.11%(2.46%)	6.83%(2.28%)	7.07%(2.37%)

Table 6. Comparison of the average test errors and sample deviation (over 100 repetitions and using five-fold cross-validation) of four algorithms on the GSE20194 gene dataset.

Percentage of flipped labels	Methods		
	ARB-2	ARB-1.5	RobustBoost
0%	9.40%(1.89%)	**9.31**%(1.96%)	10.19%(2.05%)
5%	10.02%(2.64%)	**9.88**%(2.67%)	11.21%(2.89%)
10%	12.04%(4.92%)	**11.97**%(4.67%)	12.39%(4.11%)
15%	15.72%(6.91%)	15.70%(6.56%)	**14.58**%(5.93%)
	GradientBoost-1.5	LogitBoost	AdaBoost
0%	9.87%(1.91%)	9.54%(2.31%)	9.63%(2.22%)
5%	10.16%(2.40%)	10.21%(3.32%)	10.17%(3.07%)
10%	12.31%(3.35%)	12.14%(5.16%)	12.34%(5.07%)
15%	16.12%(5.94%)	16.32%(7.20%)	16.79%(7.07%)

Table 7. Comparison of the average test errors and sample deviation (over 100 repetitions) of four algorithms on the Sensorless drive diagnosis dataset.

Percentage of flipped labels	Methods		
	ARB-2	ARB-1.5	RobustBoost
0%	5.79%(0.50%)	**5.21**%(0.41%)	6.82%(0.42%)
5%	9.49%(0.69%)	**8.06**%(0.83%)	8.74%(0.67%)
10%	12.21%(0.79%)	10.80%(0.91%)	**10.69**%(0.85%)
15%	14.34%(1.01%)	12.85%(0.89%)	**12.81**%(1.10%)
	GradientBoost-1.5	LogitBoost	AdaBoost
0%	12.52%(1.45%)	6.18%(0.48%)	6.77%(0.50%)
5%	13.98%(1.30%)	10.30%(0.67%)	11.86%(0.79%)
10%	16.00%(1.41%)	12.10%(0.72%)	13.99%(0.80%)
15%	19.31%(1.61%)	14.97%(0.88%)	17.34%(0.89%)

noisy copies of the data are available (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2011), good generalization errors in the test set are by no means guaranteed. This problem is compounded when the contamination model is unknown, where outliers need to be detected automatically. Despite progress on outlier-removing algorithms, significant practical challenges (due to exceedingly restrictive conditions imposed therein) remain. Hence, a classification method that does not rely on the specified model of the corruption in the observations is still unavailable.

As boosting algorithms use observed data distribution over iterations, they may provide a robust alternative to the existing classification methods. Among the boosting algorithms, the most famous one is AdaBoost (Freund and Schapire 1997) that averages simple estimators (classifiers) from reweighed data over a sequence of iterations. It is the first adaptive boosting algorithm because the update at each iteration is a direct function of the classification error of the previous step. AdaBoost then attracted much attention from statistics community, and has proven to be simple and effective (Zhang and Yu 2005). Breiman (1998, 1999) showed that AdaBoost is a gradient descent method in function space and Friedman, Hastie, and Tibshirani (2000) viewed AdaBoost as a gradient-based incremental search for an additive model using the exponential loss function. By observing that the exponential criterion is equivalent to the binomial log-likelihood criterion to the second order, Friedman, Hastie, and Tibshirani (2000) also proposed the LogitBoost algorithm. All these algorithms depend on standard convex optimization techniques like the Newton method. The descent method viewpoint then extends the usage of boosting to the context other than classification. For example, Friedman (2001) developed gradient boosting method for regression using squared error loss, and Mason et al. (1999) generalized the boosting idea to wider families of loss functions.

Nevertheless, in the presence of the label noise and/or outliers, the existing methods face significant challenges (Dietterich 2000). AdaBoost is known to be very sensitive to noise (Freund and Schapire 1996; Maclin and Opitz 1997; Dietterich 2000) because of the exponential criterion it uses. The weights on repeatedly misclassified data increase exponentially fast, which leads AdaBoost to overfit the noises. Algorithms like LogitBoost, MadaBoost (Domingo and Watanabe 2000), LogLossBoost (Collins, Schapire, and Singer 2002) are able to better tolerate noise than AdaBoost because they use loss functions that give much slower weights growth rate than e^x. However, they are still not insensitive to outliers or provably robust. In fact, any boosting algorithm with convex loss is highly susceptible to a random label noise as pointed out by Long and Servedio (2010).

Boost by majority (BBM; Freund 1995) follows a very different mechanism and can give up on repeatedly misclassified observations because it has a preassigned number of boosting iterations. Hence, the weights updating rule of BBM is nonconvex. However, the nonadaptiveness of BBM prevents its practical usage because the uniform bound $1/2 - \gamma$ ($\gamma > 0$) on the errors of weak learners are hard to achieve. BrownBoost (Freund 2001) combines the nonconvexity of BBM and the adaptiveness of AdaBoost, and RobustBoost (Freund 2009) is developed based on BrownBoost and further adapts to the idea of margin maximization which is believed to be the reason for the good generalization performance of AdaBoost (Freund and Schapire 1999; Rätsch, Onoda, and Müller 2001; Shapire 2003). However, BrownBoost hinders upon an extra tuning parameter, target error ϵ, and RobustBoost depends on both target error ϵ and maximum margin θ. These tuning parameters make both algorithms highly inconsistent with respect to minor changes in the population parameter settings. Furthermore, both BrownBoost and RobustBoost do not fit in the mainstream boosting algorithms that analytically minimize a convex loss function. They solve two differential equations for two unknowns at each iteration, and the loss function (which they call potential function) changes after every iteration and converges to the 0–1 loss. Although stable in simulations, the statistical properties and robustness are unknown. Therefore, a natural question is: how do we formally develop an adaptive, mainstream and robust boosting algorithm that has a nonconvex loss function and has provable robustness properties? In this article, we address this question and propose a fully automatic estimator, *ArchBoost*, with no tuning parameters, that has provable robustness guarantees. Since ArchBoost does not require the knowledge of the erroneous labels, or the knowledge of the errors themselves, one can probe the utilities of the algorithm in the extremely wide scope of heterogenous problems.

ArchBoost keeps the initial motivation of the boost by majority method in that the algorithms gives up on repeatedly misclassified observations. However, unlike BBM or RobutBoost it does so without requiring any pretuning of the error or maximum margin. ArchBoost adaptively learns which data to give up on without a priori intervention. Additionally, ArchBoost keeps the reweighing flavor of the AdaBoost or GradientBoost algorithms but it differs in the way it minimizes the empirical risk function as it allows for nonconvex losses. While GradientBoost uses least-square and Newton criterions for finding the optimal classifier, ArchBoost uses the hardness condition to define an estimating equations and solves the equations directly (not approximately). Because of that, ArchBoost does not reduce to the existing methods when the loss function of choice is a recognized convex loss, for example, ArchBoost does not reduce to the L_2Boost when the loss is the least-square loss.

5.8. Discussion

We showed that ArchBoost algorithms with nonconvex losses are robust alternatives to the popular gradient boosting type algorithms. We illustrated that the algorithm works for a very general class of loss functions (see Definition 1). Additionally, the robustness, summarized in Theorems 3 and 4, holds for an arbitrary Lipschitz loss function. Hence, it presents novel proof of why is LogitBoost more robust than the AdaBoost, a folklore observation made by many experts in the field. The statistical consistency proof is centered around "tilted" loss functions that are nonconvex in particular. We believe that nonconvex losses have great and unexplored potential for robust high-dimensional statistics. The framework of "tilted" loss functions is very general and can very well be explored for robust variable selection and estimation through an appropriate penalization scheme. Moreover, it is very well known that the

impact of outliers is multiplied in case of inferential problems, such as confidence intervals and testing. By screening out many large outliers, "tilted" losses may significantly improve upon asymptotic efficiency.

Appendix A: Proof of Theorem 1

Proof of Theorem 1(a). First, we show that at each iteration t, as long as the empirical margin $\hat{\mu}(w_t, h_t)$ is positive, the empirical risk decreases by adding the weak hypotheses h_t to the current estimate. Second, we show that the weak hypothesis returned by our ArchBoost algorithm always has a positive empirical margin before convergence.

Step 1. On the sample $\mathcal{S}_n$, denote $\mathbf{F}_{t-1} = (F_{t-1}(x_1), \ldots, F_{t-1}(x_n))$. Denote the partial derivative w.r.t. $F(X_i)$ as $g_t(X_i) = [\frac{\partial \hat{R}_{\phi,n}(\mathbf{F})}{\partial F(X_i)}]_{F(X_i)=F_{t-1}(X_i)} = \frac{1}{n} Y_i \phi^{'}(Y_i F_{t-1}(X_i))$. Then the gradient of $\hat{R}_{\phi,n}$ at $\mathbf{F}_{t-1}$ is $\nabla \hat{R}_{\phi,n}(\mathbf{F}_{t-1}) = \frac{1}{n}\mathbf{g}$ for $\mathbf{g} = (g_t(X_1), \ldots, g_t(X_n))^\top$. Recall that $w_t(X_i, Y_i) = -\phi^{'}(Y_i F_{t-1}(X_i))$ for each $i = 1, \ldots, n$. Suppose that we choose a weak hypothesis h_t with positive empirical margin w.r.t. weights w_t, that is, $\hat{\mu}(h_t, w_t) > 0$, and denote $\mathbf{h}_t = (h_t(X_1), \ldots, h_t(X_n))$. Then $\langle -\nabla \hat{R}_{\phi,n}(\mathbf{F}_{t-1}), \mathbf{h}_t \rangle = \frac{1}{n}\sum_{i=1}^n Y_i h_t(X_i) w_t(X_i, Y_i) = \hat{\mu}(h_t, w_t) > 0$, where $\langle \cdot, \cdot \rangle$ is the standard inner product in $\mathbb{R}^n$. Therefore, we have $\langle -\nabla \hat{R}_{\phi,n}(\mathbf{F}_{t-1}), \mathbf{h}_t \rangle > 0 \iff \hat{\mu}(w_t, h_t) > 0$. Next, observe that if $\langle -\nabla \hat{R}_{\phi,n}(\mathbf{F}_{t-1}), \mathbf{h}_t \rangle > 0$, then $\mathbf{h}_t$ is a descending direction of $\hat{R}_{\phi,n}(\mathbf{F})$ at $\mathbf{F}_{t-1}$, therefore $\hat{R}_{\phi,n}[\mathbf{F}_t] = \hat{R}_{\phi,n}[\mathbf{F}_{t-1} + \alpha_t \mathbf{h}_t] < \hat{R}_{\phi,n}[\mathbf{F}_{t-1}]$. Note that an appropriate step size α_t can be found by the line search $\alpha_t = \text{argmin}_\alpha \hat{R}_{\phi,n}[\mathbf{F}_{t-1} + \alpha \mathbf{h}_t]$. In summary, if at step t, we choose a base learner h_t such that $\hat{\mu}(w_t, h_t) > 0$ and choose a suitable step size α_t either by line search or set to be appropriately small, then

$$\hat{R}_{\phi,n}(F_t) < \hat{R}_{\phi,n}(F_{t-1}). \tag{A.1}$$

Step 2. In any region R_t^j, $h_t \equiv \gamma_t^j$. Then, $\langle -\mathbf{g}_t, \mathbf{h}_t \rangle = \sum_{j=1}^{J_t} \sum_{i \in R_t^j} Y_i w_t(X_i, Y_i) \gamma_t^j = \sum_{j=1}^{J_t} \gamma_t^j (\mathbb{P}_{w_t}(Y = 1 | X \in R_t^j) - \mathbb{P}_{w_t}(Y = -1 | X \in R_t^j)) \sum_{i \in R_t^j} w_t(X_i, Y_i)$.

From (6), we have $\frac{\phi^{'}(-F_t(x))}{\phi^{'}(F_t(x))} = \frac{\mathbb{P}_{w_t}(Y=1|x)}{\mathbb{P}_{w_t}(Y=-1|x)}$. $\frac{\phi^{'}(-F_t(x))}{\phi^{'}(F_{t-1}(x))}$. Observe that if $\mathbb{P}_{w_t}(Y = 1|x) > \mathbb{P}_{w_t}(Y = -1|x)$, then $\frac{\phi^{'}(-F_t(x))}{\phi^{'}(F_t(x))} > \frac{\phi^{'}(-F_{t-1}(x))}{\phi^{'}(F_{t-1}(x))}$. By second part of Lemma 1, $F_t(x) > F_{t-1}(x)$, that is, $h_t(x) > 0$. Therefore, there exists a strictly increasing function θ with the only root at 1/2 such that $\gamma_t^j = \theta(\mathbb{P}_{w_t}(Y = 1|X \in R_t^j))$. Hence, $\langle -\mathbf{g}_t, \mathbf{h}_t \rangle = \sum_{j=1}^{J_t} \theta(\mathbb{P}_{w_t}(Y = 1|X \in R_t^j))(2\mathbb{P}_{w_t}(Y = 1|X \in R_t^j) - 1) \sum_{i \in R_t^j} w_t(X_i, Y_i) \geq 0$. The last inequality is because $\theta(\mathbb{P}_{w_t}(Y = 1|X \in R_t^j))$ always has the same sign as $2\mathbb{P}_{w_t}(Y = 1|X \in R_t^j) - 1$, and "=" holds if and only if $\mathbb{P}_{w_t}(Y = 1|X \in R_t^j) = \frac{1}{2}$ for all $j = 1, \ldots, J_t$. □

Proof of Theorem 1(b). Here, we develop ideas similar to the proof of Lemma 4.1 and Lemma 4.2 in Zhang and Yu (2005). There are two differences here in comparison to Zhang and Yu (2005). First, the loss is nonconvex function and second, the optimal hypothesis is chosen differently. For $f \in \cup_{T=1}^{\infty} \mathcal{F}^T$, let $H_f \subset \mathcal{H}$ be the set that contains all weak hypotheses in f. For example, $f_1 = \sum_{h \in H_f} \alpha_1^h h$ and $f_2 = \sum_{h \in H_f} \alpha_2^h h$. Moreover, denote $\bar{f}_t = \sum_{h \in H_t} \omega_t^h h$, $F_t = \sum_{h \in H_t} \alpha_t^h h$. For notation simplicity, we denote $R = \hat{R}_{\phi,n}$ since we have fixed a loss function ϕ and sample size n. Let $s^h = sign(\omega_t^h - \alpha_t^h)$. By Taylor expansion, we have $R(F_t + \alpha_{t+1} s^h h) \leq R(F_t) + \alpha_{t+1} s^h \langle \nabla R(F_t), h \rangle + \frac{\alpha_{t+1}^2}{2} \sup_{\xi \in [0,1]} R^{''}_{F_t,h}(\xi \alpha_{t+1} s^h)$, where $R_{F_t,h}(\alpha) := R(F_t + \alpha h)$. Since the Hessian of R is bounded, there exists $M > 0$ s.t. $\sup_{\xi \in [0,1]} R^{''}_{F_t,h}(\xi \alpha_{t+1} s^h) < M$. Therefore,

$$R(F_t + \alpha_{t+1} s^h h) \leq R(F_t) + \alpha_{t+1} s^h \langle \nabla R(F_t), h \rangle + \frac{\alpha_{t+1}^2}{2} M.$$

By Algorithm 1 that $R(F_{t+1}) = R(F_t + \alpha_{t+1} h_{t+1})$. Moreover, by (6), h_{t+1} is chosen as the $\arg\min_{h \in \mathcal{H}_t} \mathbb{E}_w[R(F_t + \alpha_{t+1} h)]$. Hence, for any $h \in \mathcal{H}_t$, $\mathbb{E}_w[R(F_t + \alpha_{t+1} h_{t+1})] \leq \mathbb{E}_w[R(F_t + \alpha_{t+1} h)]$. Moreover, for any bounded random variable Z, $|\mathbb{E}_w[Z] - \mathbb{E}[Z]| \leq K$ for a positive constant K. Combining the above, we have $R(F_{t+1}) \leq R(F_t + \alpha_{t+1} s^h h) + 2\epsilon_t + 2K$, for $\epsilon_t = \sup_{h \in \mathcal{H}_t} |R(F_t + \alpha_{t+1} s^h h) - \mathbb{E}[R(F_t + \alpha_{t+1} s^h h)]|$. By the arguments very much similar to Lemmas S1 and S2 of the supplement, it is easy to obtain $\epsilon_t = o_P(1)$. Since $||\bar{f}_t - F_t||_1 = o(\log t)$, and $||\bar{f}_t - F_T||_2^2 \leq \frac{||\bar{f}_t - F_t||_1^2}{t^{c_t}}$ where $c_t \in (0, 1)$ and $c_t \to 0$ as $t \to \infty$, we have $\frac{||\bar{f}_t - F_t||_1^2}{t^{c_t}} = o(\frac{\log t}{t^{c_t}} ||\bar{f}_t - F_t||_1)$. Hence,

$$\begin{aligned}
&||\bar{f}_t - F_t||_2^2 (R(F_{t+1}) - 2\epsilon_t - 2K) \\
&= o\left[\frac{\log t}{t^{c_t}} \sum_{h \in H_t} |\alpha_t^h - \omega_t^h| R(F_t + \alpha_{t+1} s^h h) \right] \\
&= o\left[\frac{\log t}{t^{c_t}} \sum_{h \in H_t} |\alpha_t^h - \omega_t^h| \right. \\
&\qquad \left. \times \left(R(F_t) + \alpha_{t+1} s^h \langle \nabla R(F_t), h \rangle + \frac{\alpha_{t+1}^2}{2} M \right) \right] \\
&= o\left[\frac{\log t}{t^{c_t}} ||\bar{f}_t - F_t||_1 R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t}} \langle \nabla R(F_t), \bar{f}_t - F_t \rangle \right. \\
&\qquad \left. + \frac{M \alpha_{t+1}^2 \log t}{2 t^{c_t}} ||\bar{f}_t - F_t||_1 \right].
\end{aligned} \tag{A.2}$$

Now we look at the situation when $\hat{\mu}(h_k, w_k) = 0$. From part (a), this happens if and only if $\mathbb{P}_{w_k}(Y = 1|X \in R_k^j) = \frac{1}{2}$, $\forall j$, that is, $\nabla R(F_k) \perp \mathcal{H}$. Since $\hat{\mu}(h_t, w_t) \to 0$, $\nabla R(F_t)$ is perpendicular to $\cup_{T=1}^{\infty} \mathcal{F}^T$, and $\langle \nabla R(F_t) - \nabla R(\bar{f}_t), \bar{f}_t - F_t \rangle \to 0$ since $\bar{f}_t - F_t \in \cup_{T=1}^{\infty} \mathcal{F}^T$. Since ϕ is Lipschitz differentiable, there exists $L > 0$ s.t. $R(F_t) - R(\bar{f}_t) \leq \langle \nabla R(\bar{f}_t), F_t - \bar{f}_t \rangle + \frac{L}{2} ||\bar{f}_t - F_t||_2^2$. Then $\langle \nabla R(\bar{f}_t), \bar{f}_t - F_t \rangle \leq R(\bar{f}_t) - R(F_t) + \frac{L}{2} ||\bar{f}_t - F_t||_2^2$. When t is large enough, there exists sequence $\tilde{\epsilon}_t \to 0$ s.t. $\langle \nabla R(F_t), \bar{f}_t - F_t \rangle \leq R(\bar{f}_t) - R(F_t) + \frac{L}{2} ||\bar{f}_t - F_t||_2^2 + \tilde{\epsilon}_t$. Then, by (A.2),

$$\begin{aligned}
&||\bar{f}_t - F_t||_2^2 (R(F_{t+1}) - 2\epsilon_t - 2K) \\
&= o\left[\frac{\log t}{t^{c_t}} ||\bar{f}_t - F_t||_1 R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t}} \langle \nabla R(F_t), \bar{f}_t - F_t \rangle \right.
\end{aligned}$$

$$
\begin{aligned}
&+\frac{\alpha_{t+1}^2\log t}{2t^{c_t}}||\bar{f}_t-F_t||_1M\Bigg]\\
&=o\Bigg[\frac{\log t}{t^{c_t}}||\bar{f}_t-F_t||_1R(F_t)+\frac{\alpha_{t+1}\log t}{t^{c_t}}\\
&\times\left(R(\bar{f}_t)-R(F_t)+\frac{L}{2}||\bar{f}_t-F_t||_2^2+\tilde{\epsilon}_t\right)\\
&+\frac{\alpha_{t+1}^2\log t}{2t^{c_t}}||\bar{f}_t-F_t||_1M\Bigg]\\
&=o\left[\frac{\log t}{t^{c_t}}||\bar{f}_t-F_t||_1R(F_t)+\frac{\alpha_{t+1}\log t}{t^{c_t}}(R(\bar{f}_t)-R(F_t))+\eta_t\right],
\end{aligned}
\tag{A.3}
$$

where $\eta_t:=\frac{\alpha_{t+1}\log t}{t^{c_t}}(\frac{L}{2}||\bar{f}_t-F_t||_2^2+\tilde{\epsilon}_t)+\frac{\alpha_{t+1}^2\log t}{2t^{c_t}}||\bar{f}_t-F_t||_1M$. Then by dividing $||\bar{f}_t-F_t||_2^2$ on both sides of (A.3), we get

$$
\begin{aligned}
R(F_{t+1})&=o\Bigg[\frac{\log t}{t^{c_t}}\frac{||\bar{f}_t-F_t||_1}{||\bar{f}_t-F_t||_2^2}R(F_t)+\frac{\alpha_{t+1}\log t}{t^{c_t}||\bar{f}_t-F_t||_2^2}\\
&\quad\times(R(\bar{f}_t)-R(F_t))+\bar{\eta}_t+2\epsilon_t+2K\Bigg]\\
&=o\Bigg[\frac{\log t}{t^{c_t/2}||\bar{f}_t-F_t||_2}R(F_t)+\frac{\alpha_{t+1}\log t}{t^{c_t}||\bar{f}_t-F_t||_2^2}\\
&\quad\times(R(\bar{f}_t)-R(F_t))+\bar{\eta}_t+2\epsilon_t+2K\Bigg],
\end{aligned}
$$

where $\bar{\eta}_t:=\frac{\alpha_{t+1}\log t}{t^{c_t}}(\frac{L}{2}+\frac{\tilde{\epsilon}_t}{||\bar{f}_t-F_t||_2^2})+\frac{\alpha_{t+1}^2\log t}{2t^{c_t/2}||\bar{f}_t-F_t||_2}M$. Therefore,

$$
\begin{aligned}
R(F_{t+1})-R(\bar{f}_t)&=o\Bigg[\frac{\log t}{t^{c_t}||\bar{f}_t-F_t||_2}R(F_t)+\frac{\alpha_{t+1}\log t}{t^{c_t}||\bar{f}_t-F_t||_2^2}\\
&\quad\times(R(\bar{f}_t)-R(F_t))+\bar{\eta}_t+2\epsilon_t+2K\Bigg]\\
&\le\frac{\xi_t\log t}{t^{c_t}||\bar{f}_t-F_t||_2}R(F_t)+\frac{\alpha_{t+1}\xi_t\log t}{t^{c_t}||\bar{f}_t-F_t||_2^2}\\
&\quad\times(R(\bar{f}_t)-R(F_t))+\xi_t\bar{\eta}_t+2\xi_t\epsilon_t+2K\xi_t,
\end{aligned}
$$

for some sequence $\xi_t\to 0$ as $t\to\infty$. Now, for $c_t\to 0$, and with α_t satisfying conditions in (b), and by Lemma 4.2 in Zhang and Yu (2005), we have $R(F_{t+1})-R(\bar{f}_t)\to 0$ as $t\to\infty$. □

Appendix B: Proof of Theorem 3

With $IF(z;T,\mathbb{P})=g_z\in H$, we have $2\lambda g_z+\mathbb{E}_{\mathbb{P}}\phi''(Y,f_{\mathbb{P},\lambda}(X))g_z(X)\Psi(X)=\mathbb{E}_{\mathbb{P}}\phi'(Y,f_{\mathbb{P},\lambda}(X))\Psi(X)-\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))\Psi(z_x)$. By taking inner product $\langle\cdot,\cdot\rangle_H$ with g_z itself, we have

$$
\begin{aligned}
&2\lambda||g_z||_H^2+\mathbb{E}_{\mathbb{P}}\phi''(Y,f_{\mathbb{P},\lambda}(X))g_z^2(X)\\
&=\mathbb{E}_{\mathbb{P}}\phi'(Y,f_{\mathbb{P},\lambda}(X))g_z(X)-\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))g_z(z_x).
\end{aligned}
\tag{B.1}
$$

Moreover, the Frechet derivative at $f_{\mathbb{P},\lambda}$ is a zero mapping hence,

$$
2\lambda\langle f_{\mathbb{P},\lambda},g_z\rangle_H+\mathbb{E}_{\mathbb{P}}\phi'(Y,f_{\mathbb{P},\lambda}(X))g_z(X)=0.
\tag{B.2}
$$

We also note that since $f_{\mathbb{P},\lambda}$ is the global minimum, then $\lambda||f_{\mathbb{P},\lambda}||_H^2+R_\phi(f_{\mathbb{P},\lambda})\le\lambda||0_H||_H^2+R_\phi(0_H)=C_\phi$ where $C_\phi=R_\phi(0_H)=\phi(0,0)$ is a constant, that is,

$$
\lambda||f_{\mathbb{P},\lambda}||_H^2\le\lambda||f_{\mathbb{P},\lambda}||_H^2+\mathbb{E}_{\mathbb{P}}\phi(Y,f_{\mathbb{P},\lambda}(X))\le C_\phi.
\tag{B.3}
$$

Finally, we have

$$
\begin{aligned}
2\lambda||g_z||_H^2&\le 2\lambda||g_z||_H^2+\mathbb{E}_{\mathbb{P}}\phi''(Y,f_{\mathbb{P},\lambda}(X))g_z^2(X)\\
&\overset{(i)}{=}\mathbb{E}_{\mathbb{P}}\phi'(Y,f_{\mathbb{P},\lambda}(X))g_z(X)-\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))g_z(z_x)\\
&\overset{(ii)}{=}-2\lambda\langle f_{\mathbb{P},\lambda},g_z\rangle_H-\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))g_z(z_x)\\
&\overset{(iii)}{\le}2\lambda\|f_{\mathbb{P},\lambda}\|_H\|g_z\|_H-\phi'(z_y,f_{\mathbb{P},\lambda}(z_x)))g_z(z_x)\\
&\overset{(iv)}{\le}2\sqrt{\lambda C_\phi}\|g_z\|_H+|\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))||g_z(z_x)|\\
&=2\sqrt{\lambda C_\phi}\|g_z\|_H+|\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))|\langle g_z,k(z_x,\cdot)\rangle_H\\
&\overset{(v)}{\le}2\sqrt{\lambda C_\phi}\|g_z\|_H+|\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))|\\
&\quad\times\sqrt{\langle g_z,g_z\rangle_H}\sqrt{\langle k(z_x,\cdot),k(z_x,\cdot)\rangle_H}\\
&=2\sqrt{\lambda C_\phi}\|g_z\|_H+|\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))|||g_z||_H|k(z_x,z_x)|.
\end{aligned}
$$

where (i) is due to (B.1); (ii) due to (B.2); (iii) is due to the Cauchy–Schwartz inequality; (iv) is due to (B.3); (v) is again due to the Cauchy–Schwartz inequality. Since k is a bounded kernel, $\exists\ M_k>0$ such that $|k(x_1,x_2)|\le M_k$ for all $x_1,x_2\in\mathcal{X}$. Hence, $|\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))|||g_z||_H|k(z_x,z_x)|\le M_k|\phi'(z_y,f_{\mathbb{P},\lambda}(z_x))|||g_z||_H$, which in turn leads to $2\lambda||g_z||_H^2\le 2\sqrt{\lambda C_\phi}\|g_z\|_H+M_k|\phi'(z_y,f_{\mathbb{P},\lambda})(z_x)|\|g_z\|_H$ and hence $||g_z||_H\le\sqrt{\frac{C_\phi}{\lambda}}+\frac{M_k|\phi'(z_y,f_{\mathbb{P},\lambda})(z_x)|}{2\lambda}$.

Supplementary Materials

Supplement consists of the detailed technical proofs of all intermediate results (Lemmas 1–4 and Lemmas S4–S6) together with the proof of Theorems 2, 4, 5, and 6. It also contains some additional results about both simulated and real data analysis (Figures S1–S3).

Acknowledgments

The authors thank the editors in chief, associate editor, and two reviewers for their insightful comments and suggestions that have led to the significant improvement of their work.

Funding

The authors gratefully acknowledge NSF support through the grant DMS-1205296.

References

Bartlett, P., Jordan, M., and McAuliffe, J. (2006), "Convexity, Classification, and Risk Bounds," *Journal of the American Statistical Association*, 101, 138–156. [661,663,664,666]

Bartlett, P., and Traskin, M. (2007), "Adaboost is Consistent," *Journal of Machine Learning Research*, 8, 2347–2368. [664,666]

Ben-Israel, A., and Mond, B. (1986), "What is Invexity?" *The Journal of the Australian Mathematical Society*, Series B, 28, 1–9. [663]

Blanchard, G., Flaska, M., Handy, G., Pozzi, S., and Scott, C. (2016), "Classification with Asymmetric Label Noise: Consistency and Maximal Denoising," *Electronic Journal of Statistics*, 10, 2780–2824. [670]

Bootkrajang, J., and Kaban, A. (2013), "Boosting in thePresence of Label Noise," *arXiv:1309.6818*. [664]

Breiman, L. (1998), "Arcing Classifiers" (with Discussion), *Annals of Statistics*, 26, 801–849. [671]

Breiman, L. (1999), "Prediction Games and Arcing Algorithms," *Neural Computation*, 11, 1493–1517. [671]

——— (2004), "Population Theory for Boosting Ensembles," *Annals of Statistics*, 32, 1–11. [664,666]

Cesa-Bianchi, N., Shalev-Shwartz, S., and Shamir, O. (2011), "Online Learning of Noisy Data," *IEEE Transactions on Information Theory*, 57, 7907–7931. [671]

Collins, M., Schapire, R. E., and Singer, Y. (2002), "Logistic Regression, AdaBoost and Bregman Distances," *Machine Learning*, 48, 253–285. [671]

Davies, P. L., and Gather, U. (2005), "Breakdown and Groups," *Annals of Statistics*, 38, 977–988. [665]

Deshwar, A. G., and Morris, Q. (2014), "PLIDA: Cross-Platform Gene Expression Normalization using Perturbed Topic Models," *Bioinformatics*, 30, 956–961. [670]

Dietterich, T. G. (2000), "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," *Machine Learning*, 40, 139–158. [671]

Domingo, C., and Watanabe, O. (2000), "Madaboost: A Modified Version of Adaboost," in *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pp. 180–189. [671]

Donoho, D. L., and Huber, P. J. (1983), "The Notion of Breakdown Point," in *A Festschrift forErich L. Lehmann*, eds. P. J. Bickel, K. Doksum and J. L. Hodges, Jr., Belmont, CA: Wadsworth, pp. 157–184. [665]

Freund, Y. (1995), "Boosting a Weak Learning Algorithm by Majority," Information and Computation, 121, 256–285. [664,671]

——— (2001), "An Adaptive Version of the Boost-by-Majority Algorithm," *Machine Learning*, 43, 293–318. [671]

——— (2009), "A More RobustBoosting Algorithm," *arXiv preprint* arXiv:0905.2138 [661,668,671]

Freund, Y., and Schapire, R. (1996), "Experiments with a New Boosting Algorithm," in *Proceedings of the 13th International conference on Machine Learning*, pp. 148–156. [671]

——— (1997), "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, 55, 119–139. [660,661,671]

——— (1999), "A Short Introduction to Boosting," *Journal-Japanese Society For Artificial Intelligence*, 14, 771–780. [671]

Friedman, J. (2001), "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, 29, 1189–1232. [660,663,671]

Friedman, J., Hastie, T., and Tibshirani, R. (2000), "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics*, 28, 337–407. [664,671]

Gentile, C., and Littlestone, N. (1999), "The Robustness of the p-Norm Algorithms," in *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pp. 1–11. [664]

Genton, M. G., and Lucas, A. (2003), "Comprehensive Definitions of Breakdown Points for Independent and Dependent Observations," *Journal of the Royal Statistical Society*, Series B, 65, 81–94. [665]

Grünwald, P. D., and Dawid, A. P. (2004), "Game Theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian Decision Theory," *Annals of Statistics*, 32, 1367–1433. [670]

Hampel, F. (1974), "The Influence Curve and its Role in Robust Estimation," *Journal of the American Statistical Association*, 69, 383–393. [661,665]

Hampel, F. R. (1968), "Contributions to the Theory of Robust Estimation," Ph.D. dissertation, University of California, Berkeley. [661,665]

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005), "The Elements of Statistical Learning: Data Mining, Inference and Prediction," *The Mathematical Intelligencer*, 27, 83–85. [667]

Jiang, W. (2004), "Process Consistency for AdaBoost," *The Annals of Statistics*, 32, 13–29. [666]

Kalai, A. T., and Servedio, R. A. (2005), "Boosting in the Presence of Noise," *Journal of Computer and System Sciences*, 73, 266–290. [664]

Kanamori, T., Takenouchi, T., Eguchi, S., and Murata, N. (2007), "Robust Loss Functions for Boosting," *Neural Computation*, 19, 2183–2244. [670]

Kearns, M., and Li, M. (1993), "Learning in the Presence of Malicious Errors," *SIAM Journal on Computing*, 22, 807–837. [664]

Koltchinskii, V., and Panchenko, D. (2002), "Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers," *The Annals of Statistics*, 30, 1–50. [664]

Littlestone, N. (1991), "Redundant Noisy Attributes, Attribute Errors, and Linear-Threshold Learning Using Winnow," in *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pp. 147–156. [664]

Long, P. M., and Servedio, R. A. (2010), "Random Classification Noise Defeats All Convex Potential Boosters," *Machine Learning*, 78, 287–304. [661,668,671]

Lutz, R. W., Kalisch, M., and Bühlmann, P. (2008), "Robustified L2 Boosting," *Computational Statistics & Data Analysis*, 52, 3331–3341. [664]

Maclin, R., and Opitz, D. (1997), "An Empirical Evaluation of Bagging and Boosting," in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 546–551. [671]

Martinez, W., and Gray, J. B. (2016), "Noise Peeling Methods to Improve Boosting Algorithms," *Computational Statistics & Data Analysis*, 93, 483–497. [664]

Masnadi-Shirazi, H., and Vasconcelos, N. (2009), "On the Design of Loss Functions for Classification: Theory, Robustness to Outliers, and Savageboost," *Advances in Neural Information Processing Systems*, 21, 1049–1056. [664]

Mason, L., Baxter, J., Bartlett, P., and Frean, M. (1999), "Functional Gradient Techniquesfor Combining Hypotheses," *Advances in Neural Information Processing Systems*, 221–246. [664,671]

McCullagh, P., and Nelder, J. A. (1989), *Generalized Linear Models*, Boca Raton, FL: Chapman & Hall/CRC. [663]

Miao, Q., Cao, Y., Xia, G., Gong, M., Liu, J., and Song, J. (2015), "RBoost: Label Noise-Robust Boosting Algorithm Based on a Nonconvex Loss Function andthe Numerically Stable Base Learners," *IEEE Transactions on Neural Networks and Learning Systems*, 27, 2216–2228. [664]

Natarajan, N., Dhillon, J., Ravikumar, P., and Tewari, A. (2013), "Learning with Noisy Labels," *Advances in Neural Information Processing Systems*, 26, 1196–1204. [670]

Nock, R., and Lefaucheur, P. (2002), "A Robust Boosting Algorithm," in *Machine Learning: ECML 2002: 13th European Conference on Machine Learning*, Springer, pp. 319–331. [664]

Rätsch, G., Onoda, T., and Müller, K. R. (2001), "Soft Margins for AdaBoost," *Machine Learning*, 42, 287–320. [671]

Rosset, S. (2005), "Robust Boosting and Its Relation to Bagging," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 249–255. [664]

Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880. [665]

Ruckstuhl, A. F., and Welsh, A. H. (2001), "Robust Fitting of the Binomial Model," *Annals of Statistics*, 29, 1117–1136. [665]

Schapire, R. E. (2003), "The Boosting Approach to Machine Learning: An Overview," in *Nonlinear Estimation and Classification*, eds. D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu, New York: Springer, pp. 149–171. [671]

Schapire, R. E. (2013), "Explaining AdaBoost," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, eds. B. Shoelkopf, Z. Luo, and V. Vovk, Berlin Heidelberg: Springer, pp. 37–52. [661]

Stefanski, L. A., Wu, Y., and White, K. (2014), "Variable Selection in Nonparametric Classification via Measurement Error Model Selection Likelihoods," *Journal of American Statistical Association*, 109, 574–589. [668]

Stromberg, A. J., and Ruppert, D. (1992), "Breakdown in Nonlinear Regression," *Journal of the American Statistical Association*, 87, 991–997. [665]

Tyler, D. E. (1994), "Finite Sample Breakdown Points of Projection based Multivariate Location and Scatter Statistics," *Annals of Statistics*, 22, 1024–1044. [665]

Zhang, T., and Yu, B. (2005), "Boosting with Early Stopping: Convergence and Consistency," *Annals of Statistics*, 33, 1538–1579. [664,665,666,671,672,673]

Zhang, X., Wu, Y., Wang, L., and Li, R. (2014), "Variable Selection for Support Vector Machines in Moderately High Dimensions," *Journal of the Royal Statistical Society*, Series B, 78, 53–76. [670]