LETTER Communicated by Klaus-Robert Müller

Robust Loss Functions for Boosting

Takafumi Kanamori
kanamori@is.titech.ac.jp
Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo 152-8552, Japan

Takashi Takenouchi
ttakashi@is.naist.jp
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

Shinto Eguchi
eguchi@ism.ac.jp
Institute of Statistical Mathematics, and Department of Statistical Science, Graduate University of Advanced Studies, Minato-ku, Tokyo 106-8569, Japan

Noboru Murata
noboru.murata@eb.waseda.ac.jp
School of Science and Engineering, Waseda University, Shinjuku, Tokyo 169-8555, Japan

Boosting is known as a gradient descent algorithm over loss functions. It is often pointed out that the typical boosting algorithm, Adaboost, is highly affected by outliers. In this letter, loss functions for robust boosting are studied. Based on the concept of robust statistics, we propose a transformation of loss functions that makes boosting algorithms robust against extreme outliers. Next, the truncation of loss functions is applied to contamination models that describe the occurrence of mislabels near decision boundaries. Numerical experiments illustrate that the proposed loss functions derived from the contamination models are useful for handling highly noisy data in comparison with other loss functions.

1 Introduction

Boosting is a learning method to construct a good predictor by combining so-called weak hypotheses. A typical implementation of boosting algorithms is Adaboost (Freund & Schapire, 1997), whose ability has been demonstrated from practical and theoretical viewpoints. Some statistical properties have been clarified based on the argument of the uniform convergence of empirical processes (Schapire, Freund, Bartlett, & Lee, 1998; Bartlett, Jordan, & McAuliffe, 2003). In addition, the relation between

boosting algorithms and statistical estimators such as the maximum likelihood estimator has been shown from the viewpoint of information geometry (Lebanon & Lafferty, 2002; Murata, Takenouchi, Kanamori, & Eguchi, 2004).

Friedman, Hastie, and Tibshirani (2000) pointed out that boosting algorithms can be uniformly understood as coordinate descent methods derived from some classes of loss functions. In the past decade, some useful loss functions for classification problems have been proposed, for example, the hinge loss for support vector machine (Cortes & Vapnik, 1995; Schölkopf & Smola, 2001) and the exponential loss for Adaboost (Friedman et al., 2000). These typical loss functions are convex, and therefore highly developed optimization techniques can be applied to attain the global optima of loss functions. The relation between loss functions and prediction performance is widely studied in statistics and machine learning communities, since loss functions play an important role in statistical inference.

In this letter, we discuss the robustness of boosting algorithms. It is pointed out that in the Adaboost procedure, excessively heavy weights are often assigned to solitary examples, even though these examples are outliers that should not be learned. Several proposals to improve robustness have already been made (Rätsch, Onoda, & Müller, 2001; Servedio, 2003). Kalai and Servedio (2003) proposed a theoretical foundation of boosting in the presence of noisy data from the viewpoint of probably approximately correct (PAC) learning. On the other hand, we deal with those problems by applying theoretical techniques in robust statistics. We focus on two typical contamination situations in binary classification problems: outliers and mislabels.

Outliers are a small number of observations that lie outside the scope of the assumption for data. Outliers may occur in recording data. It is often indicated that outliers seriously degrade the generalization performance of Adaboost, even though the ratio of outliers is small (Rätsch et al., 2001). There are some criteria to measure the robustness of estimators. For example, *gross error sensitivity* (Hampel, Rousseeuw, Ronchetti, & Stahel, 1986) evaluates the worst case of estimator fluctuations against outliers. An estimator that minimizes the gross error sensitivity is called the most B-robust estimator, which is expected to be resistant to outliers. Here, "B" denotes "bias." Moreover, robust estimators for observations with several sorts of outliers are introduced (see, e.g., Victoria-Feser, 2002). In this letter, we investigate the condition of loss functions that derive the most B-robust estimator when there are outliers and apply these loss functions to construct boosting algorithms.

One of the other types of contamination is mislabeling, which may occur independent of the feature vector or may occur near the decision boundary of the feature space. A typical example of mislabels is the false diagnosis that usually occurs near the decision boundary. In the context of binary regression problems, statistical models describing mislabels, which are

called *contamination models*, have been studied (Copas, 1988; Takenouchi & Eguchi, 2004). We use contamination models to deal with a certain number of mislabels.

Our main objective is to propose boosting algorithms that are robust against both outliers and mislabels at the same time. This letter is organized as follows. In section 2, we briefly introduce boosting algorithms from the viewpoint of optimizing loss functions. In section 3, we expound on the relation between loss functions and statistical models. In section 4, we explain some concepts in robust statistics and derive robust loss functions. Consequently, we propose a transformation formula of loss functions. The transformation of loss functions makes boosting algorithms robust against outliers. The results in section 4 are applied to contamination models in section 5, and their validity is illustrated with numerical experiments in section 6. The last section is devoted to concluding remarks. The appendices provide some proofs of theorems. In this letter, mainly statistical properties of boosting in the case of the limit of infinitely many large training sets are studied theoretically. The performance in finite sample size is assessed by numerical experiments.

The preliminary results in section 4 have been presented earlier (Kanamori, Takenouchi, Eguchi, & Murata, 2004), and are expanded significantly here; the results in section 5 are new.

2 Boosting Algorithms

Several studies have clarified that boosting algorithms are derived from gradient descent methods for loss functions (Friedman et al., 2000; Mason, Baxter, Bartlett, & Frean, 1999). (For gradient descent methods, refer to Bertsekas, 1999.) The derivation is illustrated in this section.

Suppose that a set of samples, $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, is observed, where x_i is an element in an input space $\mathcal{X}$, and y_i takes 1 or -1 as class labels. We denote a set of hypotheses by

$$\mathcal{H} = \{h_t : \mathcal{X} \to \{1, -1\} \mid t = 1, 2, 3, \ldots\}.$$

Each hypothesis is a class label function from input space $\mathcal{X}$. Outputs of hypothesis denote a prediction of labels for given inputs. The cardinality of $\mathcal{H}$ may be uncountably infinite. In this letter, $\mathcal{H}$ is assumed to be finite set or countably infinite set for the sake of simple explanations.

Our aim is to construct a powerful predictor from observed samples by combining hypotheses in $\mathcal{H}$. In this letter, the term *predictor* denotes functions from $\mathcal{X}$ to $\mathbb{R}$. Let predictor $H_\alpha(x)$ be a linear combination of hypotheses in $\mathcal{H}$, that is,

$$H_\alpha(x) = \sum_{t=1}^{\infty} \alpha_t h_t(x), \tag{2.1}$$

where α denotes $\alpha = \{\alpha_t\}_{t=1}^{\infty}$ and $\alpha_t \in \mathbb{R}$ for all t. The prediction of class labels for input $x \in \mathcal{X}$ is given by $\mathrm{sign}(H_\alpha(x))$, where $\mathrm{sign}(z)$ denotes the sign of z. Let $\mathcal{S}[\mathcal{H}]$ be the set of predictors,

$$\mathcal{S}[\mathcal{H}] = \{H_\alpha \mid \|\alpha\|_1 < \infty\},$$

where $\|\alpha\|_1$ denotes $\sum_{t=1}^{\infty} |\alpha_t|$. For any input $x \in \mathcal{X}$, the value of $H_\alpha(x)$ in $\mathcal{S}[\mathcal{H}]$ is finite because the absolute convergence of infinite series, equation 2.1, is ensured.

To estimate a predictor, we often use a loss function $L : \mathbb{R} \to \mathbb{R}$. Some assumptions on loss functions will be shown in the next section. Prediction accuracy of a predictor is measured by the risk defined as follows.

Definition 1 (Risk). *Let $\mathcal{Z}$ be $\mathcal{X} \times \{1, -1\}$. For a probability measure Q on $\mathcal{Z}$, risk $R_L(Q, H)$ of predictor $H : \mathcal{X} \to \mathbb{R}$ under loss function $L : \mathbb{R} \to \mathbb{R}$ is defined as*

$$R_L(Q, H) = \int_{\mathcal{Z}} Q(dz) L(-yH(x)).$$

Typically, increasing functions are employed as loss functions. The prediction accuracy of $H(x)$ tends to be higher as the value of the risk is smaller, because $yH(x)$ takes large values with a high probability. Note that the positivity of $yH(x)$ indicates the correct prediction.

In practical situations, empirical distribution of a set of samples is substituted into risk as probability measure Q. The empirical distribution $\widehat{P}$ is defined as

$$\widehat{P}(B, y) = \frac{1}{n} \sum_{i=1}^{n} I(x_i \in B, y_i = y),$$

where $I(\cdot)$ denotes the indicator function and B is any measurable subset of $\mathcal{X}$. Then the risk is equal to

$$R_L(\widehat{P}, H) = \frac{1}{n} \sum_{i=1}^{n} L(-y_i H(x_i)),$$

which is called *empirical risk*.

We show a generic boosting algorithm. In practice, input to the boosting algorithm is training data or empirical distribution. Here, for theoretical analysis, we introduce a boosting algorithm that takes any probability measure Q on $\mathcal{Z}$ as input. Thus, the boosting algorithm is described as statistical functional (Hampel et al., 1986).

The boosting algorithm is derived from the gradient method for minimizing risk $R_L(Q, H)$ with respect to H. Predictor $H(x)$ is updated to $H(x) + \alpha h(x)$ to decrease the value of risk, where $H \in \mathcal{S}[\mathcal{H}]$ and $h \in \mathcal{H}$. Suppose that α is infinitesimal; then

$$R_L(Q, H + \alpha h) - R_L(Q, H) = -\alpha \int_{\mathcal{Z}} Q(dz) L'(-yH(x)) yh(x) + o(\alpha) \tag{2.2}$$

holds. If α is positive, a preferable hypothesis $h \in \mathcal{H}$ would be a minimizer of equation 2.2. One has $yh(x) = 1 - 2I(y \neq h(x))$, and thus the equality

$$\begin{aligned} &\arg\min_{h \in \mathcal{H}} - \int_{\mathcal{Z}} Q(dz) L'(-yH(x)) yh(x) \\ &\quad = \arg\min_{h \in \mathcal{H}} \int_{\mathcal{Z}} Q(dz) L'(-yH(x)) I(y \neq h(x)) \end{aligned} \tag{2.3}$$

holds. Equation 2.3 is regarded as the gradient direction of risk $R_L(Q, H)$ at H. Although exact minimization with respect to $h \in \mathcal{H}$ is difficult, in practice it is enough to find hypotheses that approximately minimize the differential of risk at H.

The value of equation 2.3 is interpreted as a weighted error of hypothesis h, and the weight on a sample (x, y) is given as $L'(-yH(x))$. When the value of $yH(x)$ is large (small), the weight on the sample (x, y) is small (large). This means that weights on samples will increase if the samples are not well trained. Adjusting weights on samples is an effective technique for boosting algorithms. We can apply common learning algorithms to find hypothesis h, which approximately minimizes weighted error on training samples, because many learning algorithms accept weighted samples as inputs. Even if a learning algorithm does not accept weighted samples as inputs, resampling techniques can be used to generate pseudo-weighted samples. In the context of boosting, the learning algorithm for hypothesis h is referred to as a *weak learner*.

As a consequence of the above argument, a boosting algorithm based on loss function L is given in Figure 1. Note that the boosting algorithm with exponential loss $L(z) = e^z$ provides Adaboost when distribution Q is an empirical distribution and the initial predictor is given as $H^{(0)}(x) = 0$. In the case of Adaboost, each coefficient $\alpha^{(m)}$ is given in an analytical form.

3 Loss Functions and Associated Models

In learning algorithms for classification problems, loss functions are often used to construct predictors that achieve high prediction performance.

Boosting Algorithm under Loss Function L

Inputs: Joint distribution Q on $\mathcal{Z} = \mathcal{X} \times \{1, -1\}$ and initial predictor $H^{(0)} \in \mathcal{S}[\mathcal{H}]$. In practice, Q is the empirical probability of training samples.

Loop For $m = 1, \ldots, M$

1. Find a hypothesis $h^{(m)} \in \mathcal{H}$ such that

$$h^{(m)} = \arg\min_{h \in \mathcal{H}} \int_{\mathcal{Z}} Q(dz) L'(-yH^{(m-1)}(x)) I(y \neq h(x)).$$

   The minimization does not need to be exact.

2. Find a coefficient $\alpha^{(m)} \in \mathbb{R}$ such that

$$\alpha^{(m)} = \arg\min_{\alpha \in \mathbb{R}} R_L(Q, H^{(m-1)} + \alpha h^{(m)}).$$

   Line search methods or Newton methods can be applied to solve the one-dimensional optimization problem.

3. Update the predictor: $H^{(m)} = H^{(m-1)} + \alpha^{(m)} h^{(m)}$.

Output: $H^{(M)}$ as an estimated predictor.

Figure 1: Pseudocode for generic boosting algorithms.

Thus, the design of loss functions is a central issue in learning theory. Typically, convex loss functions are used because of their ease of optimization. For example, exponential loss $L(z) = e^z$ in the Adaboost algorithm (Freund & Schapire, 1997; Friedman et al., 2000) and hinge loss $L(z) = \max\{0, z + 1\}$ in the support vector machine (Cortes & Vapnik, 1995) have been proposed for practical learning algorithms.

For rigorous arguments, let us define loss functions as follows:

Definition 2 (Loss Function). *Loss function $L : \mathbb{R} \to \mathbb{R}$ is a function that is strictly increasing and convex on $\mathbb{R}$ and is strictly convex on the set of negative real numbers.*

Note that nonnegativity or lower-boundedness of loss function is not assumed. In section 5, we introduce an example of unbounded loss function originally proposed by Takenouchi and Eguchi (2004). The condition on convexity is a bit different between the positive and negative numbers. The exponential loss function satisfies the definition, but the hinge loss does not. When a loss function L is differentiable, one has $L'(z) \geq 0$ for all $z \in \mathbb{R}$ due to monotone increasing and $L''(z) > 0$ for all $z < 0$ due to strict convexity. (See proposition B.4 in Bertsekas, 1999, for details.)

In this letter, the analysis is restricted to convex loss functions, since most boosting algorithms use convex loss functions. Although there are some boosting algorithms derived from nonconvex loss functions, usually risk function defined from nonconvex loss function has some local minima, and even a searching approximation of optimal solution is hopeless. The relation between loss function and statistical model is illustrated in this section, and for nonconvex loss functions, the corresponding statistical model is unlikely to have the property of identifiability. This is very undesirable for estimating conditional probability. From a practical viewpoint, using only convex loss functions is not a significant restriction.

To prove the theorems in this letter, we assume the conditions on differentiability of loss functions shown below. We clearly specify which conditions are required to prove theorems:

A1: Loss function L is once continuously differentiable on $\mathbb{R}$.

A2: Loss function L is twice continuously differentiable except at zero, and inequality

$$\sup_{z\neq 0, |z|\leq A} |L''(z)| < \infty$$

holds for any $A > 0$.

A3: Loss function L is three times continuously differentiable except at zero, and inequality

$$\sup_{z\neq 0, |z|\leq A} |L'''(z)| < \infty$$

holds for any $A > 0$.

Here, we explore the relation among predictors, loss functions, and conditional probabilities. Let $P(y|x)$ be a conditional probability of class labels given input x. Suppose that samples are identically and independently distributed. When the number of samples approaches infinity, empirical risk converges in probability to the expectation of the loss function by the law of large numbers,

$$R_L(\widehat{P}, H) \longrightarrow \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P(y|x) L(-yH(x)) = R_L(\mu P, H) \quad (n \to \infty), \tag{3.1}$$

where μ is a probability measure on $\mathcal{X}$ and μP is the probability measure on $\mathcal{X} \times \{1, -1\}$ defined as

$$\mu P(B, y) = \int_B \mu(dx) P(y|x), \tag{3.2}$$

for any measurable set $B \subset \mathcal{X}$.

We derive predictor H that minimizes risk $R_L(\mu P, H)$. The integrand of equation 3.1 is minimized at $H(x)$, which satisfies

$$\frac{\partial\left[\sum_{y=\pm 1} P(y|x)L(-yH(x))\right]}{\partial H(x)} = 0, \tag{3.3}$$

because $\sum_{y=\pm 1} P(y|x)L(-yH(x))$ is convex in $H(x)$ under A1 and extreme values are identical to the minimum value in convex functions. Equation 3.3 is equivalent to

$$\frac{P(1|x)}{P(-1|x)} = \frac{L'(H(x))}{L'(-H(x))}. \tag{3.4}$$

Note that if $P(-1|x)$ is equal to zero, equation 3.3 is reduced to

$$L'(-H(x)) = 0,$$

and it has no solution in $H(x)$ because L' is always positive under A1, as shown in the proof of lemma 1. To solve equation 3.4, let us define an odd function $\rho_L(z)$ as

$$\rho_L(z) = \frac{1}{2}\log\frac{L'(z)}{L'(-z)}.$$

When $L(z) = e^z$, the function ρ_L is equal to the identity function: $\rho_L(z) = z$. For this reason, ρ_L is multiplied by a factor of $\frac{1}{2}$.

To define ρ_L based on L, one needs not only nonnegativity of L' but also positivity of L'. Indeed, the following lemma holds. We leave the proof to appendix A.

Lemma 1. *Let L be a loss function. Under condition A1, function ρ_L is well defined on $\mathbb{R}$ and is strictly increasing.*

Equation 3.4 is equivalent to

$$\frac{1}{2}\log\frac{P(1|x)}{P(-1|x)} = \rho_L(H(x)),$$

and if a solution exists, it is unique by lemma 1. The solution is given by

$$H(x) = \rho_L^{-1}\left(\frac{1}{2}\log\frac{P(1|x)}{P(-1|x)}\right). \tag{3.5}$$

Note that the sign of the predictor, sign($H(x)$), is identical to the Bayes rule of $P(y|x)$ (McLachlan, 1992).

When predictor $H(x)$ satisfies equation 3.5 for arbitrary $x \in \mathcal{X}$, the conditional probability has the form of

$$P(y|x) = \frac{1}{1 + \exp\{-2\rho_L(yH(x))\}}. \tag{3.6}$$

and vice versa. Based on equality 3.6, let us define a statistical model $M[\rho_L]$ as

$$\mathcal{M}[\rho_L] = \{P_{\rho_L}(y|x, H) \mid H \in \mathcal{S}[\mathcal{H}]\},$$

where

$$P_{\rho_L}(y|x, H) = \frac{1}{1 + \exp\{-2\rho_L(yH(x))\}}.$$

The statistical model $\mathcal{M}[\rho_L]$ is specified by L, so $\mathcal{M}[\rho_L]$ is called the associated model of loss function L. The relation between loss functions and associated models is summarized in theorem 1.

Theorem 1. *Suppose that loss function L satisfies condition A1. If $P(y|x) = P_{\rho_L}(y|x, H_\alpha) \in \mathcal{M}[\rho_L]$ holds, then the optimal predictor that minimizes risk $R_L(\mu P, H)$ is equal to $H_\alpha \in \mathcal{S}[\mathcal{H}]$.*

According to theorem 1, the estimator of conditional probability is constructed as follows. Suppose that predictor H is given by boosting algorithm under loss function L. The estimator for the underlying conditional probability is given by $P_{\rho_L}(y|x, H)$. Typically, in boosting algorithms, the dimension of $\mathcal{S}[\mathcal{H}]$ is high, and statistical model $\mathcal{M}[\rho_L]$ is large enough to find a good approximation of conditional probability $P(y|x)$.

For odd function ρ, which is strictly increasing, we can define statistical model $\mathcal{M}[\rho]$ as well as $\mathcal{M}[\rho_L]$, that is,

$$P(y|x) \in \mathcal{M}[\rho]$$
$$\iff {}^{\exists}H \in \mathcal{S}[\mathcal{H}],\ P(y|x) = P_\rho(y|x, H) = \frac{1}{1 + \exp\{-2\rho(yH(x))\}}.$$

If ρ is not strictly increasing, the identifiability of model $\mathcal{M}[\rho]$ is likely to be violated. For nonconvex function L, usually the monotonicity of ρ_L does not hold. In such a case, different predictors may indicate the same conditional probability. To avoid such complicated situations, we impose strict monotonicity on ρ.

The relation among loss functions is clarified by corollary 1.

Corollary 1. *Suppose that loss functions L_1 and L_2 satisfy condition A1 and that $P(y|x) = P_\rho(y|x, H_\alpha) \in \mathcal{M}[\rho]$ holds. If equality $\rho_{L_1} = \rho_{L_2} = \rho$ holds, then the minimizer of $R_{L_1}(\mu P, H)$ in H is identical to that of $R_{L_2}(\mu P, H)$, and the solution is given as H_α.*

Proof. Under the condition of $\rho_{L_1} = \rho_{L_2} = \rho$, we have $\mathcal{M}[\rho] = \mathcal{M}[\rho_{L_1}] = \mathcal{M}[\rho_{L_2}]$ and $P(y|x) = P_\rho(y|x, H_\alpha) = P_{\rho_{L_1}}(y|x, H_\alpha) = P_{\rho_{L_2}}(y|x, H_\alpha)$. Then the statement in the corollary is clear from theorem 1.

Note that generally there are infinitely many loss functions that satisfy $\rho_L = \rho$ for an odd function ρ. The minimizer of $R_L(Q, H)$ is identical for any loss function L with $\rho_L = \rho$, when $Q \in \mathcal{M}[\rho]$.

Example 1. The following three loss functions,

$$L_1(z) = e^z, \quad L_2(z) = \log(1 + e^{2z}), \text{ and } L_3(z) = \begin{cases} z & z \geq 0 \\ \frac{1}{2}e^{2z} - \frac{1}{2} & z < 0 \end{cases}, \quad (3.7)$$

provide the same associated model, because $\rho_{L_1}(z) = \rho_{L_2}(z) = \rho_{L_3}(z) = z$ holds. Note that these loss functions satisfy condition A1. The associated model of these loss functions is given as

$$P_\rho(y|x, H) = \frac{1}{1 + \exp\{-2yH(x)\}}, \quad (3.8)$$

where ρ is the identity function $\rho(z) = z$. Model 3.8 is a logistic model that is widely used for statistical data analysis. Note that loss function L_1 is used for Adaboost, L_2 for Logitboost (Friedman et al., 2000), and L_3 for Madaboost (Domingo & Watanabe, 2000).

Logistic models belong to exponential families in which function ρ is linear. If ρ is a nonlinear function, $\mathcal{M}[\rho]$ is not an exponential family. That is, the associated model has statistical curvature, which is defined in terms of information geometry (Amari & Nagaoka, 2000).

Under a finite number of samples, the chosen predictors under each empirical risk are not necessarily identical. Thus, under the finite samples, the statistical properties of estimated predictors depend on loss functions.

In the context of statistics, loss function L is regarded as an M-estimator under statistical model $\mathcal{M}[\rho_L]$ when the dimension of $\mathcal{M}[\rho_L]$ is finite (van der Vaart, 1998). It is important to design appropriate loss functions according to the situation in which training data are observed. If the underlying probability $P(y|x)$ lies in $\mathcal{M}[\rho_L]$ of finite dimension, the log-likelihood

estimator for $\mathcal{M}[\rho_L]$ has good statistical properties. For a training set with few outliers, the log-likelihood estimator tends to be unstable, and some robust estimators would be useful to deal with contaminated data.

4 Robust Loss Functions

In data analysis, observed samples are often contaminated by outliers or unspecified noise. In the machine learning community, it has been pointed out that Adaboost is sensitive to outliers and that predictors given by Adaboost do not have high prediction performance for noisy data. Some boosting algorithms are proposed to overcome this drawback (Servedio, 2003; Rätsch et al., 2001).

In this section we derive robust loss functions for boosting algorithms. Concepts and techniques in the field of robust statistics (Hampel et al., 1986) are applied to make boosting algorithms robust against outliers. First, we study robust loss functions for one-dimensional statistical models. Next, we apply theoretical results of one-dimensional models to infinite-dimensional models $\mathcal{M}[\rho]$ to derive robust loss functions for boosting algorithms.

4.1 Most B-Robust Loss Functions. First, we consider estimators in one-dimensional models. Let us define one-dimensional model $\mathcal{M}_1[\rho, h, H]$ as

$$\mathcal{M}_1[\rho, h, H] = \left\{ P_\rho^1(y|x, \alpha) \,\middle|\, \alpha \in \mathbb{R} \right\},$$

where $h \in \mathcal{H}$, $H \in \mathcal{S}[\mathcal{H}]$, and

$$P_\rho^1(y|x, \alpha) = \frac{1}{1 + \exp\{-2\rho(yH(x) + \alpha y h(x))\}}.$$

Statistical model $\mathcal{M}_1[\rho, h, H]$ is regarded as a submodel of $M[\rho]$ because $H + \alpha h$ is an element in $\mathcal{S}[\mathcal{H}]$ for arbitrary $\alpha \in \mathbb{R}$. As explained in theorem 1, when $P(y|x) = P_\rho^1(y|x, \alpha_0) \in \mathcal{M}_1[\rho, h, H]$, parameter α, which minimizes the risk $R_L(\mu P, H + \alpha h)$, is given by $\alpha = \alpha_0$ under the condition of $\rho_L = \rho$.

When some outliers may be included in observed samples, loss functions that are tolerant of outliers should be used. Otherwise a few outliers would undermine the reliability of estimated predictors. To construct robust loss functions, we need to quantify the influence of outliers on estimators.

We use influence functions, which are commonly studied in the field of robust statistics (Hampel et al., 1986), to measure the influence of outliers. If joint probability $\mu(dx)P_\rho^1(y|x, \alpha_0)$ is affected by an infinitesimal contamination from outlier $(\tilde{x}, \tilde{y})$, contaminated probability $\widetilde{P}_\varepsilon^1$ is defined by

$$\widetilde{P}_\varepsilon^1(B, y) = (1 - \varepsilon) \int_B \mu(dx) P_\rho^1(y|x, \alpha_0) + \varepsilon I(\tilde{x} \in B, y = \tilde{y}), \tag{4.1}$$

where B is an arbitrary measurable subset in $\mathcal{X}$ and ε is the occurrence probability of outliers. When the risk is measured by loss function L, optimal parameter $\alpha_\varepsilon(\tilde{x}, \tilde{y})$, under the contaminated probability $\widetilde{P}^1_\varepsilon$, is given as

$$\alpha_\varepsilon(\tilde{x}, \tilde{y}) = \arg\min_{\alpha \in \mathbb{R}} \; R_L(\widetilde{P}^1_\varepsilon, H + \alpha h).$$

Note that when $\varepsilon = 0$, $\alpha_\varepsilon(\tilde{x}, \tilde{y})$ is equal to α_0.

Influence function $\phi((\tilde{x}, \tilde{y}), L, \alpha_0)$ is defined by the infinitesimal shift of the estimated parameter as:

$$\phi((\tilde{x}, \tilde{y}), L, \alpha_0) = \lim_{\varepsilon \to +0} \frac{\alpha_\varepsilon(\tilde{x}, \tilde{y}) - \alpha_0}{\varepsilon}. \tag{4.2}$$

From the definition of the influence function, parameter $\alpha_\varepsilon(\tilde{x}, \tilde{y})$ is approximately given as

$$\alpha_\varepsilon(\tilde{x}, \tilde{y}) = \alpha_0 + \varepsilon\, \phi((\tilde{x}, \tilde{y}), L, \alpha_0) + o(\varepsilon),$$

and this expression suggests that the absolute value of $\phi((\tilde{x}, \tilde{y}), L, \alpha_0)$ can be used as a measure of the influence of outliers.

To assess the worst-case influence of contamination, let us define *gross error sensitivity* $\gamma(L, \alpha_0)$ as

$$\gamma(L, \alpha_0) = \operatorname*{ess\,sup}_{\tilde{x} \in \mathcal{X}, \tilde{y} = \pm 1} |\phi((\tilde{x}, \tilde{y}), L, \alpha_0)|, \tag{4.3}$$

where the essential supremum is taken instead of the supremum for a technical reason. For a measurable function $f : \mathcal{Z} \to \mathbb{R}$, the essential supremum is defined as

$$\operatorname*{ess\,sup}_{z \in \mathcal{Z}} \; f(z) = \inf\{c \mid \mu P(\{z | f(z) > c\}) = 0\},$$

where μP is the probability measure on $\mathcal{Z} = \mathcal{X} \times \{1, -1\}$, defined in equation 3.2. Note that $\mu P(B, y) = 0$ if and only if $\mu(B) = 0$, under the condition that $P(y|x) > 0$ holds for any $(x, y) \in \mathcal{X} \times \{1, -1\}$.

Gross error sensitivity depends on loss function L. A loss function that minimizes $\gamma(L, \alpha_0)$ is regarded as the most robust loss function against outlier contamination. Let us define the most robust loss function based on gross error sensitivity:

Definition 3 (Most B-Robust Loss Function). *A loss function that minimizes gross error sensitivity, equation 4.3, subject to $\rho_L = \rho$ is called the most B-robust loss function at α_0 for model $\mathcal{M}_1[\rho, h, H]$.*

The condition $\rho_L = \rho$ ensures that the minimizer of $R_L(\mu P, H + \alpha h)$ in α is identical to α_0. Besides $\rho_L = \rho$, we may impose some constraints on loss functions such as A1, A2, or A3. The term *most B-robust loss function* is also used under such conditions. The gross error sensitivity and the most B-robustness are important concepts in robust statistics.

Influence function and gross error sensitivity are defined in the limit of infinitely large training sets. For finite training sets, empirical influence function is defined according to Hampel et al. (1986). The theoretical analysis of empirical influence function is, however, not simple. In this letter, the theoretical analysis for finite training sets is not treated, and some numerical simulations are presented in section 6 to assess the validity of the theoretical results.

Generally the most B-robust loss function is specified for each parameter α_0 in $\mathcal{M}_1[\rho, h, H]$. We show that in our context, the most B-robust loss function does not depend on α_0. Thus, the minimum value of the gross error sensitivity is uniformly attained by loss function L_ρ, which is defined in theorem 3.

The form of the influence function, equation 4.2, is given by theorem 2. The proof is in appendix B.

Theorem 2. *Let L be a loss function that satisfies A1 and A2. We suppose that there exists some constant $\delta > 0$ such that*

$$\forall \alpha \in (\alpha_0 - \delta, \alpha_0 + \delta), \quad \mu(\{x | H_\beta(x) + \alpha h(x) = 0\}) = 0 \tag{4.4}$$

holds, where $H_\beta \in \mathcal{S}[\mathcal{H}]$ and $h \in \mathcal{H}$. For outlier $(\tilde{x}, \tilde{y})$, we assume

$$H_\beta(\tilde{x}) + \alpha_0 h(\tilde{x}) \neq 0. \tag{4.5}$$

Then the influence function of loss function L at α_0 for model $\mathcal{M}_1[\rho_L, h, H_\beta]$ is given as

$$\phi((\tilde{x}, \tilde{y}), L, \alpha_0) = \frac{L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x}))}{C(L, \rho_L)} \tilde{y}h(\tilde{x}), \tag{4.6}$$

where $C(L, \rho)$ is defined as

$$C(L, \rho) = \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho^1(y|x, \alpha_0) L''(-yH_\beta(x) - \alpha_0 yh(x)). \tag{4.7}$$

Note that integral 4.7 is well defined from assumption 4.4, though $L''(z)$ is not necessarily defined at $z = 0$ under conditions A1 and A2.

The form of the most B-robust loss functions for $\mathcal{M}_1[\rho, h, H]$ is specified by the following theorem. The proof is in appendix C.

Theorem 3. *Let $\rho : \mathbb{R} \to \mathbb{R}$ be an odd function, which is once continuously differentiable, and $\rho'(z) > 0$ holds for any $z \in \mathbb{R}$. For parameter α_0 of $\mathcal{M}_1[\rho, h, H_\beta]$ and probability measure μ on $\mathcal{X}$, we suppose that equation 4.4 holds. Then the most B-robust loss function at α_0 for $\mathcal{M}_1[\rho, h, H_\beta]$ subject to A1 and A2 is given as*

$$L_\rho(z) = \begin{cases} z & z \geq 0 \\ \displaystyle\int_0^z e^{2\rho(w)} dw & z < 0 \end{cases}. \tag{4.8}$$

Note that loss function L_ρ does not depend on parameter α_0. This is significant in practice because L_ρ always minimizes the gross error sensitivity regardless of the parameter in $\mathcal{M}_1[\rho, h, H_\beta]$.

The derivative of L_ρ is equal to

$$L'_\rho(z) = \begin{cases} 1 & z \geq 0 \\ e^{2\rho(z)} & z < 0 \end{cases},$$

which is proportional to weights on samples in boosting algorithms. Derivative $L'_\rho(z)$ is constant for $z \geq 0$, and thus significant weights are not assigned to outliers. This is an intuitive reason that why L_ρ is robust against outliers.

The loss function for Madaboost, L_3, is the most B-robust loss function for one-dimensional logistic models in which the function ρ is given as $\rho(z) = z$. Loss functions for Adaboost, Logitboost, and Madaboost, which are defined by L_1, L_2, and L_3 in equation 3.7, are plotted in Figure 2. The derivatives of these loss functions are also plotted.

Next, we study how loss functions control the effect of each sample on estimators. When loss function L is used to estimate parameter α of $\mathcal{M}_1[\rho, h, H]$, the estimator is given as the solution of equations

$$\sum_{i=1}^{n} L'(-y_i H(x_i) - y_i \alpha h(x_i)) y_i h(x_i) = 0.$$

This equation denotes that the derivative of empirical risk vanishes at the estimated parameter. We find that the previous equation is equivalent to

$$\sum_{i=1}^{n} v(H(x_i) + \alpha h(x_i)) \left\{ \frac{y_i + 1}{2} - P_\rho^1(+1|x_i, \alpha) \right\} h(x_i) = 0,$$

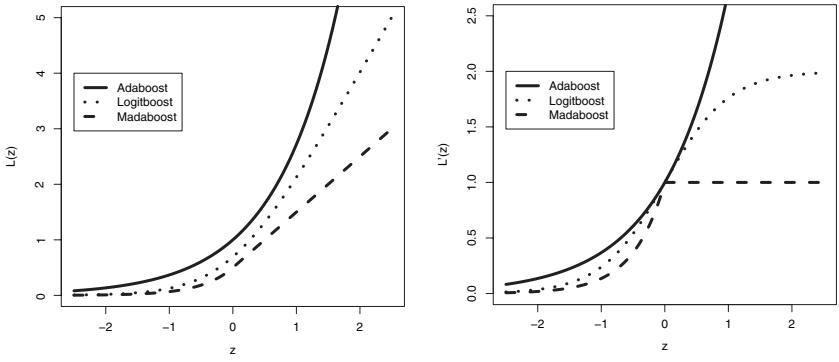

Figure 2: Loss functions for logistic models $\mathcal{M}_1[\rho, h, H]$, where $\rho(z) = z$. (Left) Loss functions for Adaboost, Logitboost, and Madaboost. Constant values are added to loss function such that $\lim_{z\to-\infty} L(z) = 0$. (Right) Derivatives of functions are plotted.

where weight function v is defined by

$$v(z) = L'(z) + L'(-z),$$

because

$$\begin{aligned} & L'(-y_i H(x_i) - y_i \alpha h(x_i)) y_i h(x_i) \\ &= \frac{L'(-y_i H(x_i) - y_i \alpha h(x_i))}{v(-y_i H(x_i) - y_i \alpha h(x_i))} v(-y_i H(x_i) - y_i \alpha h(x_i)) y_i h(x_i) \\ &= y_i \left(1 - P_\rho^1(y_i | x, \alpha)\right) v(H(x_i) + \alpha h(x_i)) h(x_i) \\ &= \left\{ \frac{y_i + 1}{2} - P_\rho^1(1 | x, \alpha) \right\} v(H(x_i) + \alpha h(x_i)) h(x_i) \end{aligned}$$

holds. Note that v is an even function.

Therefore, loss function L determines weights of samples via function $v(z)$. Under logistic models with $\rho(z) = z$, function $v(z)$ for the maximum likelihood estimator is a constant function. The weight functions for Adaboost, Logitboost, and Madaboost are plotted in Figure 3.

Note that the absolute value of $H(x) + \alpha h(x)$ becomes large when conditional probability $P_\rho^1(+1|x, \alpha)$ is close to 0 or 1. In the case of Adaboost with $L(z) = e^z$, if $P_\rho^1(+1|x, \alpha)$ is close to 1, the change in sign of the class labels from 1 to -1 at x significantly affects the estimation result because weight function $v(H(x) + \alpha h(x))$ takes a large value, while the weight function for Madaboost depresses the influence of such outliers. Likewise, a loss function such as L_ρ in equation 4.8 depresses the influence of outliers.

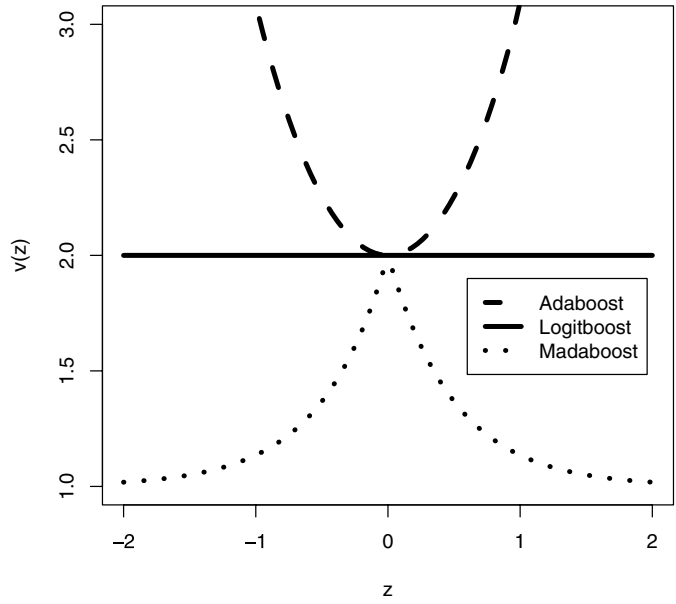

Figure 3: Weight functions for Adaboost, Logitboost, and Madaboost are plotted. Loss functions for these boosting algorithms are given in equation 3.7.

Victoria-Feser (2002) proposed some other weight function candidates to depress the influence of outliers.

4.2 Robustness of Boosting Algorithms. The most B-robust loss functions for one-dimensional model $\mathcal{M}_1[\rho, h, H]$ were studied in the previous section. Next, we study the robustness of boosting algorithms under model $\mathcal{M}[\rho]$. Here, contaminated probability $\widetilde{P}_\varepsilon$ of $P_\rho(y|x, H) \in \mathcal{M}[\rho]$ is defined by

$$\widetilde{P}_\varepsilon(B, y) = (1-\varepsilon)\int_B \mu(dx)P_\rho(y|x, H) + \varepsilon I(\tilde{x} \in B, y = \tilde{y}),$$

where B is any measurable subset in $\mathcal{X}$.

We study the influence of contaminations for $P_\rho(y|x, H)$. This means that we consider the situation such that

$$Q = \widetilde{P}_\varepsilon, \ H^{(0)} = H, \text{ and } M = 1 \tag{4.9}$$

in the boosting algorithm in Figure 1. Let an output of the boosting algorithm under loss function L be $\widetilde{H}_{L,\varepsilon}$.

The setup of equation 4.9 may seem unrealistic. This situation is considered a final phase of the boosting algorithm. That is, when enough accurate predictor is already obtained, we would like to evaluate how much one-step, that is, $M = 1$, boosting algorithm affects the estimator under contamination by outliers.

To measure the influence of outlier $(\tilde{x}, \tilde{y})$ on $\widetilde{H}_{L,\varepsilon}$, we define the *change-of-risk function* $\psi_{L_0}((\tilde{x}, \tilde{y}), L, H)$ as

$$\psi_{L_0}((\tilde{x}, \tilde{y}), L, H) = \lim_{\varepsilon \to +0} \frac{R_{L_0}(\mu P, \widetilde{H}_{L,\varepsilon}) - R_{L_0}(\mu P, H)}{\varepsilon^2}, \tag{4.10}$$

where $P(y|x) = P_\rho(y|x, H) \in \mathcal{M}[\rho]$ and L_0 is a loss function to measure the difference between H and $\widetilde{H}_{L,\varepsilon}$. We suppose that L and L_0 satisfy equality

$$\rho_L = \rho_{L_0} = \rho.$$

A typical example of L_0 is

$$L_0(z) = \log(1 + e^{2\rho(z)}),$$

because $L_0(-yH(x))$ corresponds to the negative log likelihood of $P_\rho(y|x, H) \in \mathcal{M}[\rho]$. The change-of-risk sensitivity $\gamma_{L_0}(L, H)$ for $\mathcal{M}[\rho]$ is also defined as

$$\gamma_{L_0}(L, H) = \operatorname*{ess\,sup}_{\tilde{x} \in \mathcal{X}, \tilde{y} = \pm 1} \psi_{L_0}((\tilde{x}, \tilde{y}), L, H). \tag{4.11}$$

In Hampel et al. (1986), the change-of-variance function and the change-of-variance sensitivity are defined to investigate the influence of outliers on the asymptotic variance of estimators. In our context, the number of parameters in model $\mathcal{M}[\rho]$ is usually infinite, and the standard definition in Hampel et al. (1986) does not work well. Therefore, we define the change-of-risk function and the change-of-risk sensitivity based on risk measure. The minimizer of $\gamma_{L_0}(L, H)$ in L subject to $\rho_L = \rho$ is called the most B-robust loss function for model $\mathcal{M}[\rho]$ according to the definition for one-dimensional models.

When $P(y|x) = P^1_\rho(y|x, \alpha_0) \in \mathcal{M}_1[\rho, h, H]$ holds and the parameter α_0 of $\mathcal{M}_1[\rho, h, H]$ is estimated by using loss function L, proportionality

$$\lim_{\varepsilon \to +0} \frac{R_{L_0}(\mu P, H + \alpha_\varepsilon(\tilde{x}, \tilde{y})h) - R_{L_0}(\mu P, H + \alpha_0 h)}{\varepsilon^2} \propto \phi((\tilde{x}, \tilde{y}), L, H)^2$$

holds. This is shown in the proof of theorem 4. Thus, for the one-dimensional model $\mathcal{M}_1[\rho, h, H]$, we have

$$\psi_{L_0}((\tilde{x}, \tilde{y}), L, H) \propto \phi((\tilde{x}, \tilde{y}), L, H)^2$$

and

$$\gamma_{L_0}(L, H) \propto \gamma(L, H)^2. \tag{4.12}$$

The proportional constant depends on L_0. Hence, change-of-risk function 4.10 and change-of-risk sensitivity 4.11 are regarded as a generalization of influence function 4.2 and gross error sensitivity 4.3, respectively.

According to equation 4.12, we expect that loss function L_ρ defined in theorem 3 also minimizes change-of-risk sensitivity 4.11 for model $\mathcal{M}[\rho]$. The rigorous statement is shown in theorem 4. The proof is in appendix D.

Theorem 4. *Let $\rho : \mathbb{R} \to \mathbb{R}$ be an odd function that is twice continuously differentiable and $\rho'(z) > 0$ holds for any $z \in \mathbb{R}$. Suppose that loss function L_0 satisfies A1 and A2 and that equality $\rho_{L_0} = \rho$ holds. For a probability measure μ on $\mathcal{X}$ and a predictor $H_\beta \in \mathcal{S}[\mathcal{H}]$, we assume that there exists some constant $\delta_0 > 0$ such that*

$$\forall c \in (-\delta_0, \delta_0), \quad \mu(\{x | H_\beta(x) = c\}) = 0 \tag{4.13}$$

holds. Then the loss function L_ρ is a minimizer of the change-of-risk sensitivity $\gamma_{L_0}(L, H_\beta)$ in L subject to conditions, A1, A2, A3, and $\rho_L = \rho$.

Theorem 4 explains that the boosting algorithm based on L_ρ is robust against outliers at probability $P_\rho(y|x, H_\beta)$. If the initial predictor, $H^{(0)}$, is far from $H_\beta(x)$, the influence of outliers on boosting algorithms is still not clearly understood. However, we think that theorem 4 suggests a practical way of constructing robust-boosting algorithms.

4.3 Transformation of Loss Functions. Theorems 3 and 4 give us a way of transforming loss functions for robust boosting algorithms. For loss function L, the most B-robust loss function L_{ρ_L} for model $\mathcal{M}[\rho_L]$ is given as

$$L_{\rho_L}(z) = \begin{cases} z & z \geq 0 \\ \int_0^z \frac{L'(w)}{L'(-w)} dw & z < 0 \end{cases}, \tag{4.14}$$

because

$$e^{2\rho_L(w)} = \frac{L'(w)}{L'(-w)}$$

holds. Note that we can apply Newton methods to determine the value of $\alpha^{(m)}$ in boosting algorithms even though L_{ρ_L} is not given in the analytical form.

A few examples of L_{ρ_L} are shown below.

Example 2. Adaboost uses $L_1(z)$ in equation 3.7 as the loss function. We obtain $L_{\rho_{L_1}}(z) = L_3(z)$ because the function ρ_{L_1} is given as $\rho_{L_1}(z) = z$. That is, the most B-robust boosting algorithm for logistic model $\mathcal{M}[\rho_{L_1}]$ is Madaboost. The target of estimators under model $\mathcal{M}[\rho_{L_1}]$ is log-odds of conditional probabilities,

$$\frac{1}{2}\log\frac{P(1|x)}{P(-1|x)} \in \mathcal{S}[\mathcal{H}],$$

which is a basic quantity in statistical analysis (MacCullagh & Nelder, 1989).

Example 3. Truncated quadratic loss function

$$L_4(z) = (\max\{0, 1+z\})^2$$

is also used for classification problems. The support vector machine with 2-norm soft margin (Schölkopf & Smola, 2001) uses this loss function. Since the domain on which L_4 is strictly convex is restricted, the set of predictors should also be restricted as follows:

$$\bar{\mathcal{S}}[\mathcal{H}] = \{H_\alpha \mid \|\alpha\|_1 < 1\}.$$

On the interval $(-1, 1)$, L_4 satisfies conditions A1, A2 and A3. Thus, theorems 3 and 4 hold for $\bar{\mathcal{S}}[\mathcal{H}]$ instead of $\mathcal{S}[\mathcal{H}]$. Consequently, the most B-robust loss function for L_4 is

$$L_{\rho_{L_4}}(z) = \begin{cases} z & 0 \le z < 1 \\ -z - 2\log(1-z) & -1 < z < 0 \end{cases},$$

where

$$\rho_{L_4}(z) = \frac{1}{2}\log\frac{1+z}{1-z}, \quad -1 < z < 1.$$

The set of conditional probabilities in associated model $\mathcal{M}[\rho_{L_4}]$ is given as

$$P_{\rho_{L_4}}(y|x, H) = \frac{1}{2}\left(1 + yH(x)\right), \quad H \in \bar{\mathcal{S}}[\mathcal{H}].$$

Then the relation between predictors and conditional probabilities is specified by affine transformation.

Rosset (2005) has also proposed a transformation of loss functions to robustify boosting algorithms against outliers. The loss function is referred to as Huberized loss. The definition of Huberized loss is similar to equation 4.14. Indeed, the Huberized loss also has linear part beyond some point. Rosset (2005) investigated the relation between Huberized loss and bagging (Breiman, 1994) and gave the rationale of Huberized loss as hybrids of standard boosting loss functions and the bagging linear loss function.

5 Contamination Models

In this section, we introduce contamination models to deal with a change in sign of class labels or mislabels. Contamination models describe the change in sign of class labels near decision boundaries. Transformation formula 4.14 can be applied to loss functions for contamination models. Then we obtain loss functions that are resistant to an infinitesimally small number of outliers and substantially positive percentage of mislabels.

5.1 Statistical Models for Mislabels. Contamination models were proposed for describing the occurrence of mislabels (Copas, 1988). Typically, conditional probability $P(y|x)$ is contaminated, such as

$$(1-\kappa)P(y|x)+\kappa P(-y|x), \tag{5.1}$$

where κ denotes the ratio of mislabels. Note that the decision boundary of equation 5.1 is identical to that of $P(y|x)$ for values of κ less than 0.5. In an ideal situation, the probability we try to estimate from the observed samples is $P(y|x)$, while in a practical situation, the samples are generated according to the contaminated model, equation 5.1.

A typical example of a mislabel is a false diagnosis. For example, suppose that x denotes a diagnostic test results and y denotes whether a tumor is benign or malignant. Contamination model 5.1 represents the conditional probability of a diagnostic result based on x in the presence of the false diagnosis, while $P(y|x)$ denotes the probability whether a tumor is actually benign or malignant based on diagnostic test result x. Therefore, the value of κ denotes the precision of the diagnosis. It is valid to assume that the samples we can use are generated according to the contamination models.

In the context of boosting, Takenouchi and Eguchi (2004) discussed an extension of model 5.1 with ratio κ mislabeled depending on x. It would be more realistic if κ becomes larger as x is near the decision boundary. The loss function is defined by

$$L(z;\eta)=(1-\eta)e^z+\eta z, \tag{5.2}$$

where η is a nonnegative constant less than one. The boosting algorithm based on equation 5.2 is called *eta-boost*. Note that loss function 5.2 consists of the loss function for Adaboost and a linear term. Derivative $L'(z; \eta)$ with respect to z satisfies

$$z \geq 0 \Longrightarrow 1 \leq L'(z; \eta) \leq e^z$$

and

$$z < 0 \Longrightarrow 1 \geq L'(z; \eta) \geq e^z$$

because of the linear term. This means that the weight distribution of samples is shifted to the uniform distribution in comparison to Adaboost. Thus, the linear term is expected to moderate the overweighting to mislabels.

A model associated with equation 5.2 is given as

$$P(y|x, H) = (1 - \kappa_\eta(x, H))P_0(y|x, H) + \kappa_\eta(x, H)P_0(-y|x, H), \tag{5.3}$$

where $P_0(y|x, H)$ is logistic model

$$P_0(y|x, H) = \frac{1}{1 + e^{-2yH(x)}},$$

and κ is defined by

$$\kappa_\eta(x, H) = \frac{\eta}{(1 - \eta)\sum_{y=\pm 1} e^{-yH(x)} + 2\eta}.$$

Function ρ_L for the associated model of equation 5.2 is also derived as

$$\rho_L(z) = \frac{1}{2} \log \frac{(1 - \eta)e^z + \eta}{(1 - \eta)e^{-z} + \eta}.$$

Note that $\kappa_\eta(x, H)$ is maximized at x such that $H(x) = 0$ and the maximum value is $\eta/2$. Therefore, the mislabel occurs primarily near the decision boundary, and the probability of contamination, $\kappa_\eta(x, H)$, decreases exponentially as input x moves away from the decision boundary.

5.2 Robust Eta-Boost Algorithm. In Takenouchi and Eguchi (2004), the eta-boost algorithm is shown to work well on noisy samples. Possible extreme outliers, however, would degrade the generalization performance of eta-boost because the loss function increases exponentially as well as that of Adaboost.

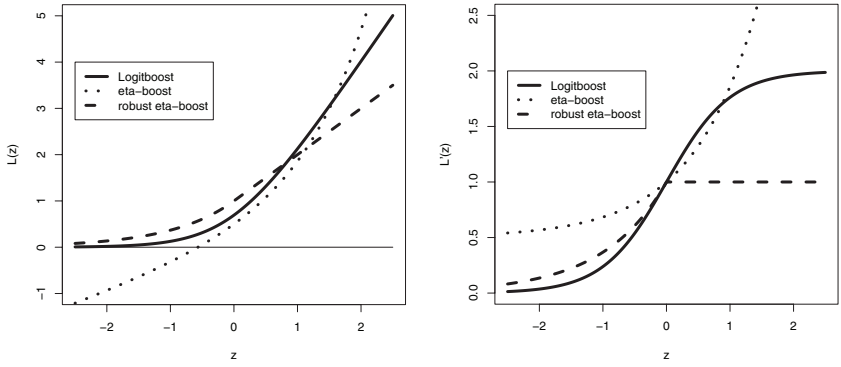

Figure 4: Loss functions for the contamination models. (Left) The loss functions for Logitboost, eta-boost, and robust eta-boost. The value of η is set to 0.5. Constant values are added to the loss function for robust eta-boost such that $\lim_{z\to-\infty} L(z;\eta) = 0$. (Right) Derivatives of functions are plotted.

As studied in the last section, transformation 4.14 of loss function 5.2 provides a robust modification of eta-boost. The transformed loss function is analytically given as

$$L_{\rho_L}(z;\eta) = \begin{cases} z & z \geq 0 \\ \frac{1}{\eta^2}[(1-\eta)(e^z-1)\eta + (2\eta-1)\log(1+(e^z-1)\eta)] & z < 0 \end{cases}, \tag{5.4}$$

and the derivative is

$$L'_{\rho_L}(z;\eta) = \begin{cases} 1 & z \geq 0 \\ \frac{(1-\eta)e^z+\eta}{(1-\eta)e^{-z}+\eta} & z < 0 \end{cases}.$$

We refer to the boosting algorithm under $L_{\rho_L}(z;\eta)$ as *robust eta-boost*. Note that

$$\lim_{\eta\to 0} L_{\rho_L}(z,\eta) = L_3(z),$$

which means that L_{ρ_L} is an extension of L_3. By using loss function 5.4, the boosting algorithm is expected to be robust against both mislabels near the decision boundary and outliers far from the decision boundary. Loss functions for eta-boost and robust eta-boost are plotted with the loss function of Logitboost in Figure 4. The derivatives are also plotted.

6 Illustrative Examples

In classification problems, the concept of margin is important in achieving high prediction accuracy (Vapnik, 1998; Schapire et al., 1998). A common definition of margin on a training sample (x, y) is given as $yH_\alpha(x)/\|\alpha\|_1$. Most binary classification methods are designed to maximize the smallest margin over training samples, that is, the optimization problem

$$\max_{\alpha} \min_{i} \frac{y_i H_\alpha(x_i)}{\|\alpha\|_1} \tag{6.1}$$

is solved approximately. In Adaboost algorithm, the predictor minimizing the exponential loss,

$$\frac{1}{n}\sum_{i=1}^{n} \exp\left\{-\|\alpha\|_1 \frac{y_i H_\alpha(x_i)}{\|\alpha\|_1}\right\},$$

can be regarded as an approximate solution of equation 6.1. In the same way, boosting under loss function L is also regarded as a learning algorithm that approximately provides predictors maximizing the smallest margin.

Under noisy data, the predictor, which is an exact solution of equation 6.1, typically overfits the training data. As a result, estimated classifiers do not achieve high prediction accuracy. Many researchers point out that maximization of the smallest soft margin often provides promising results for noisy data (Vapnik, 1998; Meir & Rätsch, 2003). A soft margin relaxes the penalty of misclassification in comparison to a margin. This relaxation depends on each learning algorithm, and the design of the soft margin is significant for achieving stable estimation of predictors under noisy data.

Robust boosting algorithms (Demiriz, Bennett, & Shawe-Taylor, 2002; Rätsch, 2001; Rätsch et al., 2000; Rätsch et al., 2001; Rätsch, Demiriz, & Bennett, 2002) also maximize the smallest soft margin. For example, in RoBoost (Rätsch et al., 2000) and Adaboost$_{Reg}$ (Rätsch et al., 2001), a soft margin is substituted into exponential loss of Adaboost instead of a margin, where the definition of soft margin is different between RoBoost and Adaboost$_{Reg}$.

In this letter, we propose the robust boosting algorithm under loss function L_ρ. Our methods do not use soft margin directly. We examine boosting with the most B-robust loss function L_ρ under noisy data and compare the results with existing boosting algorithms. In this section, six boosting algorithms are compared: Adaboost, Logitboost, Madaboost, eta-boost, robust eta-boost, and Adaboost$_{Reg}$. Loss functions for Adaboost, Logitboost, and Madaboost are introduced in equation 3.7. Loss functions for eta-boost and

robust eta-boost are shown in section 5. Adaboost$_{Reg}$ differs slightly from other boosting algorithms. In the learning process of Adaboost$_{Reg}$, changes of the weights are monitored, and the history of weights in the process is used to detect outliers. The algorithm of Adaboost$_{Reg}$ is more complicated, and its calculation is more time-consuming than other boosting algorithms based on coordinate descent methods derived from loss functions. In numerical experiments, the computational cost of Adaboost$_{Reg}$ is more than twice that of robust eta-boost, including hyperparameter estimation.

First, we study two-dimensional binary classification problems. The robustness of the proposed method is examined for toy problems. Next, we apply boosting algorithms to the data in the IDA benchmark repository (Rätsch et al., 2001; Rätsch et al., 2000).

6.1 Synthetic Example. Labeled samples are generated from a fixed probability, and class labels of a few samples are flipped as outliers. The detailed setup is as follows. Input vector x is uniformly distributed on the two-dimensional region $[-\pi, \pi] \times [-\pi, \pi]$, and the conditional probability of the class label is defined by

$$P(y|x_1, x_2; \eta, H) = \frac{(1-\eta)e^{yH(x_1,x_2)} + \eta}{(1-\eta)\sum_{y'=\pm 1} e^{y'H(x_1,x_2)} + 2\eta}, \tag{6.2}$$

where

$$H(x_1, x_2) = x_2 - 3\sin x_1.$$

Note that $P(y|x_1, x_2; \eta, H)$ corresponds to model 5.3 with $H(x_1, x_2) = x_2 - 3\sin x_1$. Moreover, a% samples are flipped as outliers, where these are randomly chosen from the top $10a$% samples sorted in descending order of $|H(x_1, x_2)|$. Hence, outliers are scattered far from the decision boundary. Typical training samples are shown in Figure 5. In these numerical experiments, the learning algorithm called "decision stumps" (Friedman et al., 2000) is used as a weak learner.

First, test errors of robust eta-boost and Adaboost$_{Reg}$ (Rätsch et al., 2001) are compared. The value of η in $P(y|x_1, x_2; \eta, H)$ is set to 0.5; that is, there are about 25% mislabels around the decision boundary. The averaged test errors in 100 different runs for robust eta-boost and Adaboost$_{Reg}$ are shown in Figure 6. The axis of the abscissa denotes the number of m in the boosting algorithm. In each run, the number of training data is set to 200, where 2% outliers are mixed with the training data. The test error is calculated using 5000 test samples that include neither mislabels nor outliers, because our main objective of the numerical experiments is to observe the influence of contamination on the test error of label prediction. Both Adaboost$_{Reg}$ and robust eta-boost include hyperparameters such as the regularization

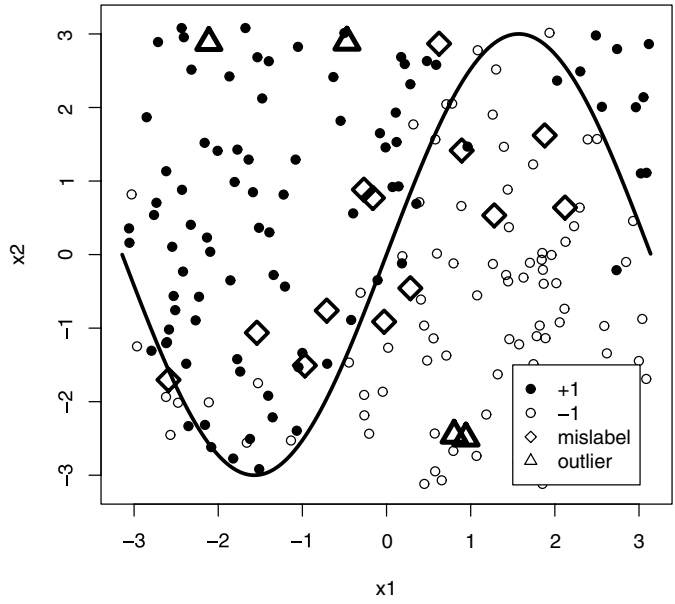

Figure 5: Typical training samples used in numerical experiments are plotted. The value of η is set to 0.5, and 2% outliers are mixed in these training samples. Mislabels are observed around the decision boundary, and some outliers are scattered far from the decision boundary.

parameter C and the mislabel ratio η, respectively. Although appropriate selection of the value of C in Adaboost$_{Reg}$ results in a fairly good generalization performance, the estimation of C is often difficult because of its wide range of values. On the other hand, the value of η for robust eta-boost is restricted to the interval $[0, 1]$, where the value has a clear meaning.

Next, the training results by Adaboost, Madaboost, and Adaboost$_{Reg}$ are compared from the viewpoint of the robustness. The mislabel ratio η in equation 6.2 is set to 0.0. Thus, there are no mislabeled data. The regularization parameter, C, for Adaboost$_{Reg}$ is set to $C = 500$. The averaged test errors in 50 different runs for Adaboost, Madaboost, and Adaboost$_{Reg}$ are shown in Figures 7a and 7b, with respect to the number of boosting iterations. As shown in the figure, Adaboost is sensitive to outliers. To demonstrate this sensitivity to the outliers, we plot the test error differences against the number of boosting iterations. In Figure 7c, the difference of test errors between 2% outliers and nonoutliers is plotted. In fact, for the training data with outliers, Figure 7b shows that Madaboost and Adaboost$_{Reg}$ stably provide an optimal performance around 50 boosting iterations, while Adaboost fails to capture such an optimal performance, which was obtained for the training

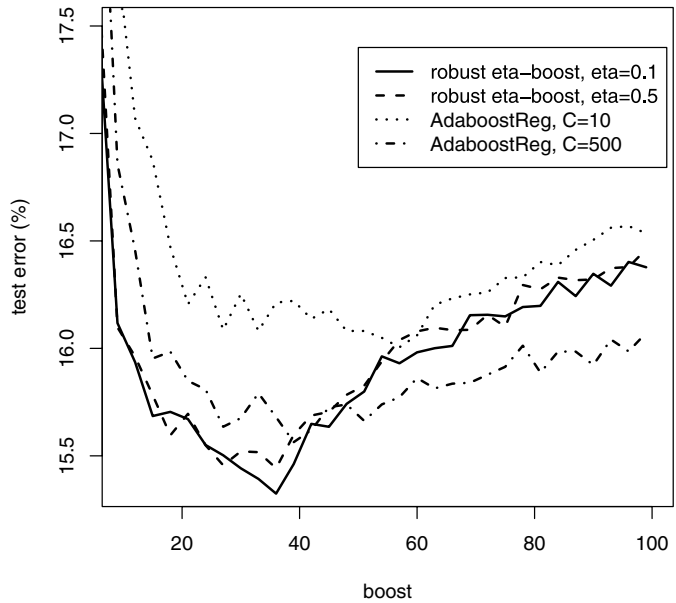

Figure 6: Averaged test errors for Adaboost$_{Reg}$ and robust eta-boost. Samples are contaminated by 2% outliers. Regularization parameter C for Adaboost$_{Reg}$ is set to 10 or 500; η in robust eta-boost is set to 0.1 or 0.5.

data without outliers as observed in Figure 7a. Thus, we conclude that Adaboost is more sensitive to outliers than Madaboost and Adaboost$_{Reg}$ under this numerical experiment.

Next, we compare six boosting algorithms. The results are shown in Table 1. In this case, the mislabel ratio η in equation 6.2 is set to 0.5. We apply 10-fold cross validation to determine hyperparameters such as the number of boosting iterations, the regularization parameter of Adaboost$_{Reg}$, and the mislabel ratio in eta-boost and robust eta-boost. The test error is evaluated for 5000 test samples that include neither mislabels nor outliers. In Table 1, asterisks denote that the test error of a particular learning algorithm is 5%($*$) or 1%($**$), significantly less than that of Adaboost under the paired t-test. The generalization performance is shown for each level of outliers, and some of the results in Table 1 are depicted in Figure 8 by box-and-whisker plots.

In case of nonoutliers, Adaboost$_{Reg}$ and eta-boost are better than the other algorithms, although the significance of their superiority is not clear. This result suggests that Adaboost$_{Reg}$ provides good performance even under contamination by mislabels near the decision boundary. As the ratio

of outliers increases, boosting algorithms other than robust eta-boost and Madaboost are affected by outliers.

As indicated by the analysis of loss functions based on gross error sensitivity, loss functions with bounded derivatives are tolerant of outliers. When there are a few outliers in the training data, the averaged results for robust eta-boost are slightly better than those for Madaboost. This suggests that the parameter η in robust eta-boost effectively depresses the influence of mislabels on estimators.

Finally, we study another kind of contamination: adding nuisance dimensions of pure noise (Blanchard, Schäfer, Rozenholc, & Müller, 2005). More precisely, for a two-dimensional input (x_1, x_2) of training data, nuisance dimensions are added as $(x_1, x_2, \varepsilon_3, \ldots, \varepsilon_\ell)$, where ε_i's are independently generated from the normal distribution with mean 0 and standard deviation 100. That is, the noise level of nuisance dimension is quite large. The dimension of input in test data is also extended, $(x_1, x_2, 0, \ldots, 0) \in \mathbb{R}^\ell$. In this experiment, the parameters of the conditional probability 6.2 are given as $\eta = 0$ and $H(x_1, x_2) = 10(x_2 - 3\sin x_1)$. Blanchard et al. (2005) have reported that the support vector machine is not robust in this setting. The results of the experiments are shown in Table 2, where "dimension" denotes

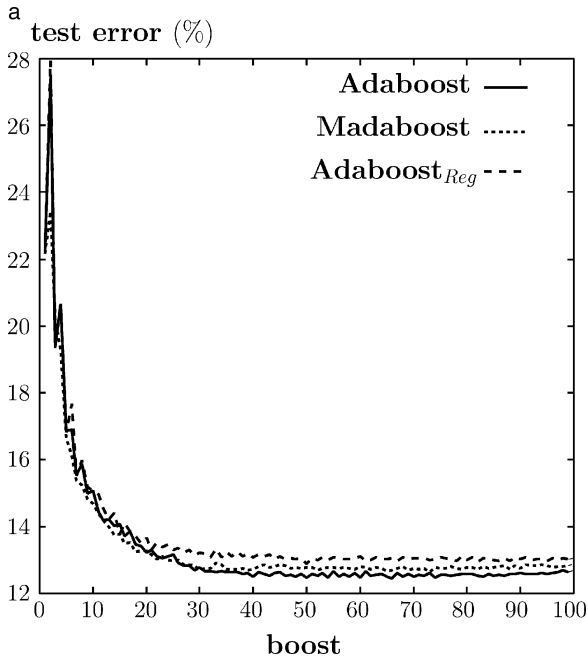

Figure 7: Test error of boosting algorithms. (a) Outliers: 0%. (b) Outliers: 2%. (c) Difference between test errors of 2%—outliers and nonoutliers.

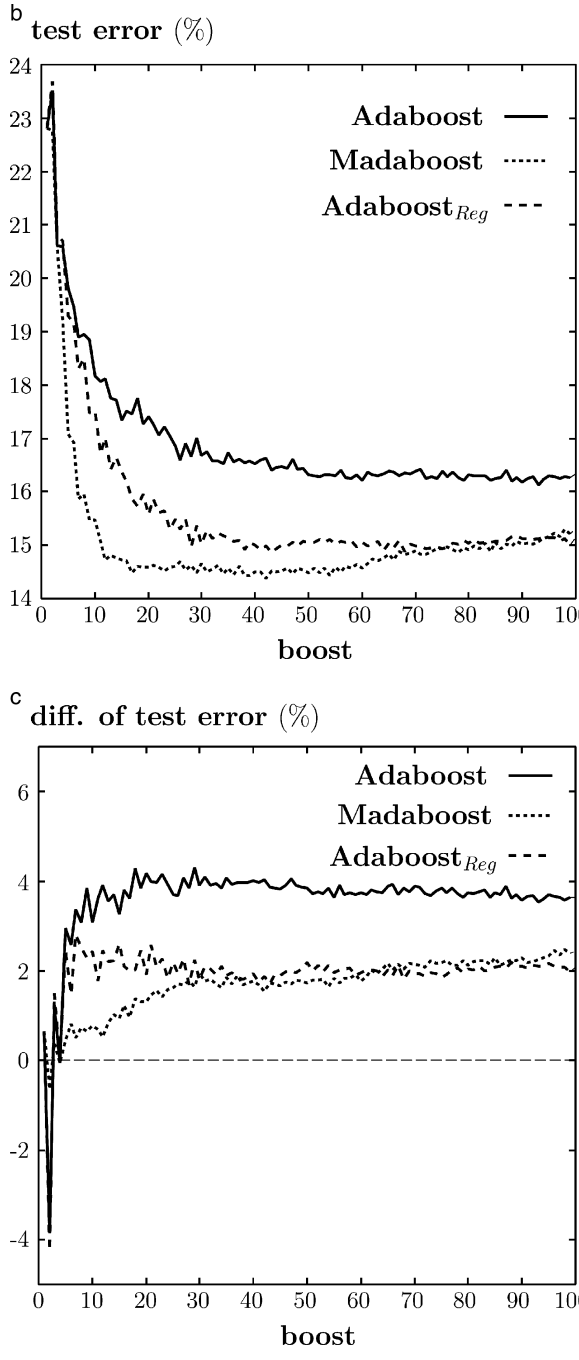

Figure 7: Continued

Table 1: Averaged Test Errors for Six Boosting Algorithms.

Outliers	0%	2%	5%
Adaboost	14.86 ± 0.23	16.50 ± 0.24	17.45 ± 0.30
Logitboost	14.95 ± 0.22	$^{*}16.01 \pm 0.22$	$^{*}16.93 \pm 0.24$
Madaboost	15.04 ± 0.22	$^{*}15.98 \pm 0.24$	$^{*}16.89 \pm 0.29$
Eta-boost	14.70 ± 0.19	16.90 ± 0.26	17.43 ± 0.26
Robust eta-boost	14.87 ± 0.20	$^{**}\mathbf{15.70 \pm 0.22}$	$^{**}\mathbf{16.62 \pm 0.21}$
Adaboost$_{Reg}$	$\mathbf{14.57 \pm 0.18}$	16.09 ± 0.30	16.96 ± 0.30

Notes: Asterisks denote that the test errors of the learning algorithm are 5%($*$) or 1%($**$), significantly less than that of Adaboost under the paired t-test. Boldface denotes the best method for each level of outliers.

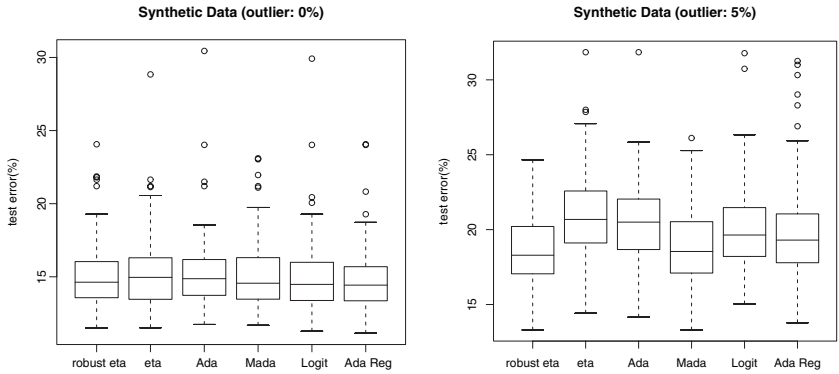

Figure 8: Test results of synthetic data. Test errors of robust eta-boost, eta-boost, Adaboost, Madaboost, Logitboost, and Adaboost$_{Reg}$ over 100 trials are shown by box-and-whisker plots. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to the most extreme data value within $1.5 \cdot$ IQR (interquartile range) of the box. Outliers are data with values beyond the end of the whiskers, which are displayed as open circles. (Left) Results without outliers. (Right) Results with 5% outliers.

the dimension of the input vector. For each setup, we consider 50 repetitions of a training set of size 200 and a test set of size 5000. In the numerical experiments, all hyperparameters are estimated from contaminated training data by applying five-fold cross validation. All of the methods except the support vector machine are stable against additional noisy dimensions. Note that the applied weak learner is the decision stumps. In general, decision

Table 2: Results of Boosting Methods, with Contamination of the Data by Extra Noisy Dimensions.

Dimension	2	4	6	8
Adaboost	7.6 ± 1.8	7.7 ± 1.6	8.9 ± 2.0	9.1 ± 1.7
Logitboost	7.4 ± 1.3	7.8 ± 1.4	8.4 ± 1.2	9.3 ± 1.9
Madaboost	7.6 ± 1.8	7.7 ± 1.2	8.6 ± 1.7	9.3 ± 1.5
Eta-boost	10.2 ± 2.2	11.2 ± 2.8	11.0 ± 2.2	11.0 ± 2.5
Robust eta-boost	7.2 ± 1.3	7.9 ± 1.5	8.7 ± 1.6	9.2 ± 1.7
Adaboost$_{Reg}$	7.6 ± 1.5	7.9 ± 1.3	8.8 ± 1.8	9.2 ± 1.5
SVM	8.1 ± 1.6	14.0 ± 1.1	16.5 ± 1.0	18.4 ± 1.3

Notes: The "dimension" in the table includes both the informative part and the purely noisy part. For each setup, we consider 50 repetitions of a training set of size 200 and a test set of size 5000. The averaged test error and the standard deviation are reported in percentage.

stumps are quite robust to various kinds of noises. We think that the stability of the boosting algorithms to additional nuisance dimensions comes from the robustness of decision stumps.

6.2 Experiments on Benchmark Data. We use 13 artificial and real-world data sets from the UCI, DELVE, and STATLOG benchmark repositories: Banana, Breast-Cancer, Diabetes, German, Heart, Image, Ringnorm, Flare-Solar, Splice, Thyroid, Titanic, Twonorm, and Waveform. All data sets are provided as IDA benchmark repository. (See Rätsch et al., 2001, and Rätsch et al., 2000, for details of data sets.) Each data set includes training sets and test sets. We divide each training set into two parts: 70% for parameter estimation and 30% for hyperparameter estimation. The properties of each data set are shown in Table 3, where dim, #train, #val, #test, and #real denote the input dimension, the size of training set for parameter estimation, the size of validation set for hyperparameter estimation, the size of test set, and the number of realizations of the data, respectively. For real-world data, realization of data denotes different random partitions of data.

We apply "decision stumps" (Friedman et al., 2000) as a weak learner. In the boosting algorithm, the parameter α of the predictor H_α is estimated on the training set for parameter estimation. The validation set is used to estimate hyperparameters such as the number of boosting iterations m, the value of η in eta-boost and robust eta-boost, or the regularization parameter C in Adaboost$_{Reg}$. We can apply the cross-validation technique to estimate the hyperparameters. In the experiments, however, we do not use cross validation because it is time-consuming.

Accuracy of estimators is evaluated as follows. The prediction accuracy of H is measured by the test error of sign(H) on the test set of each data set.

Table 3: Properties of Each Data Set.

	Data Property				
Name	dim	# train	# val	# test	# real
Banana	2	280	120	4900	100
Breast-Cancer	9	140	60	77	100
Diabetes	8	327	141	300	100
German	20	489	211	300	100
Heart	13	118	52	100	100
Image	18	909	391	1010	20
Ringnorm	20	280	120	7000	100
Flare-solar	9	466	200	400	100
Splice	60	700	300	2175	20
Thyroid	5	98	42	75	100
Titanic	3	105	45	2051	100
Twonorm	20	280	120	7000	100
Waveform	21	280	120	4600	100

Note: Dim, # train, # val, # test, and # real denote the input dimension, the size of the training set for parameter estimation, the size of the validation set, the size of the test set, and the number of realizations of the data, respectively.

The log-loss, defined as

$$\text{log-loss} = -\frac{1}{N}\sum_{i=1}^{N} \log P_{\rho_L}(\bar{y}_i|\bar{x}_i, H),$$

is used to measure the estimation accuracy of the conditional probability $P_{\rho_L}(y|x, H)$ given by the loss function L, where $\{(\bar{x}_i, \bar{y}_i),\ i = 1, \ldots, N\}$ denotes the test set of each data set. Log-loss is commonly used to measure discrepancy between the underlying probability and the estimated one (Grünwald & Dawid, 2004). To estimate the accurate classifier, we adopt the hyperparameter minimizing test error on the validation set. On the other hand, for the estimation of conditional probabilities, hyperparameter minimizing log-loss on validation set will be preferable.

For Adaboost$_{Reg}$, however, there is no clear correspondence between predictors and conditional probabilities. In Adaboost$_{Reg}$ algorithm, the soft margin is substituted into the loss function $L(-z) = e^{-z/2}$, and thus the statistical model associated with the loss function is expected to provide a natural correspondence between predictors and conditional probabilities. Therefore, we define the conditional probability estimated by Adaboost$_{Reg}$ as

$$P(y|x, H) = \frac{1}{1 + \exp\{-yH(x)\}},$$

where H is a predictor given by Adaboost$_{Reg}$.

Our main concern is to observe the influence of outliers or mislabels on estimators. Thus, outliers and mislabels are added to the training set, where training set includes all the samples for parameter estimation and validation. First, labels of training set are flipped randomly as mislabels. Note that mislabels are mixed uniformly. To add outliers to the training set, we require rough estimate of the conditional probability of labels, since the probability of outliers will be small under ideal situation. We input the training data without mislabels and outliers into rpart algorithm (Breiman, Friedman, Olshen, & Stone, 1984), and obtain rough estimates of conditional probability. New training samples with low conditional probability are generated and added to the training set as outliers. In the same way, mislabels are mixed to test sets, but outliers are not added. Since we are interested in estimation accuracy for underlying conditional probability, test sets are also contaminated by mislabels.

A predictor, H, is estimated by using training and validation sets in each realized data set, and the accuracy of H is evaluated on the test set. If the number of realizations is 100, one has 100 estimates. The averaged values of test error and log-loss $\times$ 10 for these estimates are shown with standard deviation in Tables 5 to 17 in appendix E, and the summary of experiments over 13 data sets are shown in Table 4. Table 4 indicates the winning percentage of a particular method over 13 data sets, where averaged value of test error or log-loss is used to compare with each other methods. For example, under the criterion of test error, robust eta-boost wins seven times over 13 data sets with 0% mislabels and 3% outliers. In Tables 5 to 17, detailed results are shown, where the p-values are calculated by one-side paired t-test against Adaboost.

Rätsch et al. (2001) used the same data sets (IDA repository) to assess the prediction accuracy of some boosting algorithms including Adaboost$_{Reg}$. In their experiments, for example, the test error on Banana is about 10%. In our experiments, the test error on Banana is roughly 27% under 0% mislabels and 0% outliers as shown in Table 5. The difference mainly comes from the choice of weak learner. Rätsch et al. (2001) applied radial basis function nets as the weak learner, and we use decision stumps. When we apply rpart as the weak learner, the test error on Banana improves to about 14%. On Splice and Flare-Solar, our results are better than those in Rätsch et al. (2001). In practice, the choice of weak learner is significant. In this letter, however, we focus only on investigating the properties of loss functions.

Our experiments show that:

- From the summary of the test error in Table 4, robust eta-boost and Adaboost$_{Reg}$ perform better than the other methods when there are no mislabels. In particular, Adaboost$_{Reg}$ can depress the influence of outliers even if their ratio is high.
- When the ratio of mislabel is high (20%), Madaboost seems to provide a good classifier. However, as a whole, it is not a dominating method.

Table 4: Frequency That a Particular Method Wins over 13 Data Sets, Where Averaged Values of the Test Error or Log-Loss Are Used for Comparison.

Mislabel	0%			20%		
Outlier	0%	3%	5%	0%	3%	5%
Summary: Test error						
Adaboost	0/13	2/13	0/13	1/13	0/13	2/13
Logitboost	1/13	0/13	1/13	2/13	0/13	1/13
Madaboost	1/13	1/13	0/13	1/13	**7/13**	0/13
Eta-boost	1/13	0/13	1/13	0/13	0/13	1/13
Robust eta-boost	**7/13**	**7/13**	4/13	4/13	3/13	3/13
Adaboost$_{Reg}$	3/13	3/13	**7/13**	5/13	3/13	6/13
Summary: Log-loss						
Adaboost	2/13	0/13	1/13	0/13	0/13	1/13
Logitboost	2/13	1/13	1/13	2/13	1/13	1/13
Madaboost	2/13	2/13	1/13	3/13	6/13	**7/13**
Eta-boost	0/13	1/13	0/13	0/13	0/13	0/13
Robust eta-boost	6/13	**8/13**	**9/13**	**8/13**	6/13	4/13
Adaboost$_{Reg}$	1/13	1/13	1/13	0/13	0/13	0/13

Note: Boosting algorithms in boldface represent wins over more than half over 13 data sets.

- From the lower part of Table 4, robust eta-boost outperforms the other methods in estimation of conditional probabilities. When the ratio of mislabel is high (20%), Madaboost also provides accurate estimates of the conditional probability as well as robust eta-boost.

In summary, even if mislabels or outliers contaminate training data, robust eta-boost can depress the influence of contamination on estimators in comparison to the other methods. When the target of learning is an accurate classifier, Adaboost$_{Reg}$ and robust eta-boost are better than the other methods. Regarding the estimation of conditional probabilities, Adaboost$_{Reg}$ does not perform well. Robust eta-boost and Madaboost provide a stable and accurate estimator of conditional probabilities. For Adaboost$_{Reg}$, rigorous correspondence between predictors and conditional probability is unknown. This may be a main reason that the probability estimator given by Adaboost$_{Reg}$ is not accurate in comparison to the other boosting algorithms.

7 Conclusion

We formulated a way of constructing robust boosting algorithms. We proposed a transformation formula that derives the most B-robust loss function. Applying the transformation formula to loss functions whose associated

models are contamination models, we obtain the robust eta-boost algorithm, which is robust against both mislabels and outliers, especially for the estimation of conditional probability.

In numerical experiments, we verified that robust eta-boost is robust against outliers and mislabels in comparison to the other boosting algorithms. For the estimation of conditional probability, Adaboost$_{Reg}$ does not perform well, because the relation between predictors and conditional probabilities is unclear. In order to establish the theoretical validity of Adaboost$_{Reg}$, it will be helpful to investigate the relation between predictors and conditional probabilities in Adaboost$_{Reg}$ algorithm. Alternatively, as an estimator of conditional probability, robust eta-boost and Madaboost perform better than the others, especially in the case of contamination by outliers. These results are compatible with theoretical conclusions developed in this letter.

We focused on only the most B-robust loss function, which is expected to be the least affected by outliers in the sense of gross error sensitivity. There are, however, many candidates that improve robustness from different points of view. For example, the most B-robust loss function is derived without considering the efficiency, which is measured by the variance of the estimator caused by fluctuations of given samples. Under certain assumptions, efficiency can be compatible with robustness. A loss function that copes with efficiency and robustness is called the optimal B-robust loss function. The optimal B-robust loss function has one parameter to adjust the balance between efficiency and robustness. Roughly, the optimal B-robust loss function can be composed as an intermediate of the most B-robust loss function and the efficient loss function. For logistic models, the efficient loss function is given as L_2 in equation 3.7, which is used for Logitboost. The optimal B-robust loss function for logistic models is shown in Figure 9.

Regularization terms are other important factors for good generalization performance, because the dimension of statistical models for boosting algorithms is high in general. We think that the next step of our research is theoretical analysis to derive appropriate regularization terms for more accurate prediction under contaminated data.

Appendix A: Proof of Lemma 1

Because L is convex on $\mathbb{R}$, inequality

$$L(z_1) + L'(z_1)(z_2 - z_1) \le L(z_2) \iff L'(z_1) \le \frac{L(z_2) - L(z_1)}{z_2 - z_1}$$

is valid for any $z_1 < z_2$. Likewise, we have

$$L(z_2) + L'(z_2)(z_1 - z_2) \le L(z_1) \iff \frac{L(z_2) - L(z_1)}{z_2 - z_1} \le L'(z_2)$$

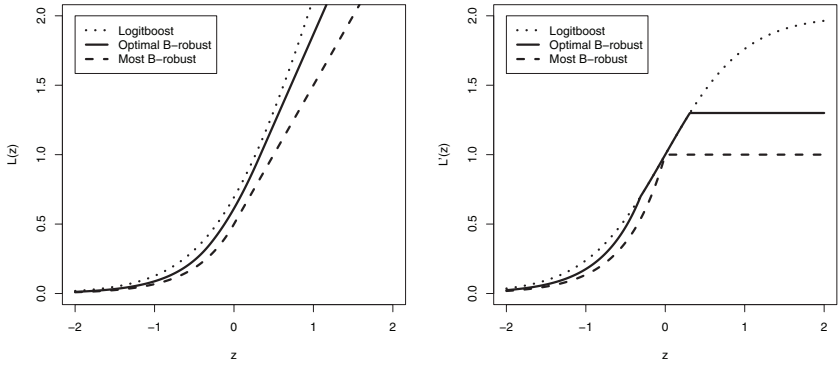

Figure 9: Loss functions for logistic models, where function ρ is defined as $\rho(z) = z$. (Left) Loss functions. Logitboost loss, optimal B-robust loss, and most B-robust loss (Madaboost) are plotted. Constant values are added such that $\lim_{z\to-\infty} L(z) = 0$. (Right) Derivatives of functions are plotted.

for $z_1 < z_2$. Hence for $z_1 < z_2$,

$$L'(z_1) \leq L'(z_2)$$

holds. Moreover, since L is strictly convex for negative real values, inequality

$$L(z_1) + L'(z_1)(z_2 - z_1) < L(z_2) \Longleftrightarrow\ L'(z_1) < \frac{L(z_2) - L(z_1)}{z_2 - z_1}$$

is valid for $z_1 < z_2 < 0$, and then, for $z_1 < z_2 < 0$, we have

$$L'(z_1) < L'(z_2).$$

When $z_1 < 0$ and $z_1 < z_2$, there exists a negative value $z_3 < 0$ such that $z_1 < z_3 < z_2$. Then the following inequality holds:

$$\begin{aligned} L(z_1) + L'(z_1)(z_2 - z_1) &= L(z_1) + L'(z_1)(z_3 - z_1) + L'(z_1)(z_2 - z_3) \\ &< L(z_3) + L'(z_1)(z_2 - z_3) \\ &\leq L(z_3) + L'(z_3)(z_2 - z_3) \\ &\leq L(z_2). \end{aligned} \tag{A.1}$$

From equation A.1, monotonicity

$$z_1 < 0, z_1 < z_2 \Longrightarrow L'(z_1) < L'(z_2) \tag{A.2}$$

is derived.

Next, we show that inequality $0 < L'(z)$ also holds for any $z \in \mathbb{R}$. For any $z \in \mathbb{R}$, there exists $z_1 < 0$ such that $z_1 < z$, and then $L'(z_1) < L'(z)$ holds because of inequality A.2. Since loss function L is strictly increasing, we have $0 \leq L'(z_1)$, and then we find that $0 < L'(z)$ holds for any $z \in \mathbb{R}$. Thus, ρ_L is well defined on $\mathbb{R}$.

If $0 \leq z_1 < z_2$ holds, we have inequalities

$$0 < L'(z_1) \leq L'(z_2),$$

and

$$0 < L'(-z_2) < L'(-z_1).$$

Thus, $\rho_L(z_1) < \rho_L(z_2)$ holds. Similarly, we can prove that $\rho_L(z)$ is strictly increasing in other cases such as $z_1 < 0 \leq z_2$ or $z_1 < z_2 < 0$.

Appendix B: Proof of Theorem 2

Inequality

$$|-yH_\beta(x) - y\alpha h(x)| \leq \|\beta\|_1 + |\alpha_0| + \delta$$

holds for any $\alpha \in (\alpha_0 - \delta, \alpha_0 + \delta)$, and then we have

$$\begin{aligned}\left|\frac{\partial}{\partial\alpha} L(-yH_\beta(x) - y\alpha h(x))\right| &= |-L'(-yH_\beta(x) - y\alpha h(x))yh(x)| \\ &\leq L'(\|\beta\|_1 + |\alpha_0| + \delta)\end{aligned} \tag{B.1}$$

because L' is a positive and nondecreasing function as shown in the proof of lemma 1. Hence, by Lebesgue's bounded convergence theorem (Halmos, 1974), the exchange of integration and differentiation,

$$\begin{aligned}&\frac{\partial}{\partial\alpha}\int_{\mathcal{X}} \sum_{y=\pm1} \mu(dx) P^1_{\rho_L}(y|x,\alpha_0) L(-yH_\beta(x) - y\alpha h(x)) \\ &\quad = -\int_{\mathcal{X}} \sum_{y=\pm1} \mu(dx) P^1_{\rho_L}(y|x,\alpha_0) L'(-yH_\beta(x) - y\alpha h(x)) yh(x),\end{aligned} \tag{B.2}$$

is valid for $\alpha \in (\alpha_0 - \delta, \alpha_0 + \delta)$ under condition A1. This means that $R_L(\widetilde{P}_\varepsilon^1, H_\beta + \alpha h)$ is differentiable in α, where $\widetilde{P}_\varepsilon^1$ is given by equation 4.1 with $\rho = \rho_L$.

Let us define $\eta(\alpha, \varepsilon)$ by

$$\eta(\alpha, \varepsilon) = \frac{\partial}{\partial \alpha} R_L\big(\widetilde{P}_\varepsilon^1, H_\beta + \alpha h\big).$$

From equation B.2, we find that $\eta(\alpha, \varepsilon)$ is of the form

$$\begin{aligned}\eta(\alpha, \varepsilon) = &-(1-\varepsilon)\int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P_{\rho_L}^1(y|x, \alpha_0) L'(-yH_\beta(x) - y\alpha h(x)) yh(x) \\ &- \varepsilon L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha h(\tilde{x}))\tilde{y}h(\tilde{x}), \end{aligned} \tag{B.3}$$

where $\eta(\alpha, \varepsilon)$ is defined on $(\alpha, \varepsilon) \in (\alpha_0 - \delta, \alpha_0 + \delta) \times [0, 1]$. The domain of ε can be analytically prolonged to $(-c, c)$ for some $c \in (0, 1)$. Thus, we can assume that $\eta(\alpha, \varepsilon)$ is defined on

$$(\alpha, \varepsilon) \in (\alpha_0 - \delta, \alpha_0 + \delta) \times (-c, c).$$

Since $R_L(\widetilde{P}_\varepsilon^1, H_\beta + \alpha h)$ is convex in α, a solution of equation $\eta(\alpha, \varepsilon) = 0$ in α is a minimizer of $R_L(\widetilde{P}_\varepsilon^1, H_\beta + \alpha h)$. That is,

$$\alpha_\varepsilon(\tilde{x}, \tilde{y}) = \operatorname*{arg\,min}_{\alpha \in \mathbb{R}} \; R_L\big(\widetilde{P}_\varepsilon^1, H_\beta + \alpha h\big) \Longleftrightarrow \eta(\alpha_\varepsilon(\tilde{x}, \tilde{y}), \varepsilon) = 0$$

holds, at least for $0 \le \varepsilon < c$. Even for $\varepsilon < 0$, we can define $\alpha_\varepsilon(\tilde{x}, \tilde{y})$ by a zero of $\eta(\alpha, \varepsilon)$ in α. We show that $\eta(\alpha, \varepsilon)$ is once continuously differentiable in a vicinity of $(\alpha_0, 0)$, and we apply the implicit function theorem to obtain $\alpha_\varepsilon(\tilde{x}, \tilde{y})$. Note that when $\alpha_\varepsilon(\tilde{x}, \tilde{y})$ is differentiable at $\varepsilon = 0$, the differential is equal to influence function $\phi((\tilde{x}, \tilde{y}), L, \alpha_0)$ because

$$\left.\frac{\partial}{\partial \varepsilon}\alpha_\varepsilon(\tilde{x}, \tilde{y})\right|_{\varepsilon=0} = \lim_{\varepsilon \to 0} \frac{\alpha_\varepsilon(\tilde{x}, \tilde{y}) - \alpha_0}{\varepsilon} = \lim_{\varepsilon \to +0} \frac{\alpha_\varepsilon(\tilde{x}, \tilde{y}) - \alpha_0}{\varepsilon} = \phi((\tilde{x}, \tilde{y}), L, \alpha_0)$$

holds.

First, the integration part of equation B.3 is continuous in α because we can apply Lebesgue's bounded convergence theorem to the first term of $\eta(\alpha, \varepsilon)$ with inequality B.1 and the continuity of L'. The second term of equation B.3 is also continuous in α due to condition A1. Thus, the continuity of $\eta(\alpha, \varepsilon)$ in a vicinity of $(\alpha_0, 0)$ is valid, and so is $\frac{\partial}{\partial \varepsilon}\eta(\alpha, \varepsilon)$.

Next, we consider the differentiability of $\eta(\alpha, \varepsilon)$ by α. From A2 and equation 4.4, for almost every $(x, y) \in \mathcal{X} \times \{1, -1\}$, $L'(-yH_\beta(x) - y\alpha h(x))$ is

differentiable by α around α_0. Moreover, for $\alpha \in (\alpha_0 - \delta, \alpha_0 + \delta)$, inequality

$$|L''(-yH_\beta(x) - y\alpha h(x))| \leq \sup_{z\neq 0, |z|\leq \|\beta\|_1 + |\alpha_0| + \delta} |L''(z)| < \infty$$

holds almost everywhere. Thus, the exchange of integration and differential,

$$\frac{\partial}{\partial \alpha} \int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P^1_{\rho_L}(y|x, \alpha_0) L'(-yH_\beta(x) - y\alpha h(x)) y h(x)$$
$$= -\int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P^1_{\rho_L}(y|x, \alpha_0) L''(-yH_\beta(x) - y\alpha h(x)), \tag{B.4}$$

is valid for $\alpha \in (\alpha_0 - \delta, \alpha_0 + \delta)$. We can prove the continuity of equation B.4 around $\alpha = \alpha_0$ in a similar fashion. The second term of $\eta(\alpha, \varepsilon)$ is also differentiable by α because of equation 4.5 and A2. Therefore, $\eta(\alpha, \varepsilon)$ is differentiable by α, and its derivative is equal to

$$\frac{\partial}{\partial \alpha} \eta(\alpha, \varepsilon) = (1 - \varepsilon) \int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P^1_{\rho_L}(y|x, \alpha_0) L''(-yH_\beta(x) - y\alpha h(x))$$
$$+ \varepsilon L''(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha h(\tilde{x})).$$

Note that the second term, $L''(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha h(\tilde{x}))$, is continuous around $\alpha = \alpha_0$ under equation 4.5 and A2.

Therefore, $\eta(\alpha, \varepsilon)$ is once continuously differentiable. Let $\mathcal{X}_+$ and $\mathcal{X}_-$ be

$$\mathcal{X}_+ = \{x \in \mathcal{X} | H_\beta(x) + \alpha_0 h(x) > 0\}$$

and

$$\mathcal{X}_- = \{x \in \mathcal{X} | H_\beta(x) + \alpha_0 h(x) < 0\},$$

respectively. From condition 4.4, $\mu(\mathcal{X}_+) > 0$ or $\mu(\mathcal{X}_-) > 0$ holds. Thus, the derivative of $\eta(\alpha, \varepsilon)$ by α at $(\alpha, \varepsilon) = (\alpha_0, 0)$ satisfies the following inequality,

$$\frac{\partial}{\partial \alpha} \eta(\alpha, \varepsilon)\Big|_{\alpha=\alpha_0, \varepsilon=0} = \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P^1_{\rho_L}(y|x, \alpha_0) L''(-yH_\beta(x) - y\alpha_0 h(x))$$
$$\geq \int_{\mathcal{X}_+} \mu(dx) P^1_{\rho_L}(1|x, \alpha_0) L''(-H_\beta(x) - \alpha_0 h(x))$$
$$+ \int_{\mathcal{X}_-} \mu(dx) P^1_{\rho_L}(-1|x, \alpha_0) L''(H_\beta(x) + \alpha_0 h(x))$$
$$> 0, \tag{B.5}$$

since L is strictly convex for $z < 0$, and $L''(z) > 0$ holds for $z < 0$. Given the continuous differentiability of $\eta(\alpha, \varepsilon)$ around $(\alpha, \varepsilon) = (\alpha_0, 0)$, and the positivity of $\frac{\partial \eta}{\partial \alpha}(\alpha_0, 0)$, we can apply the implicit function theorem to $\eta(\alpha, \varepsilon)$. Consequently, $\alpha_\varepsilon(\tilde{x}, \tilde{y})$ is once continuously differentiable at $\varepsilon = 0$, and the derivative is given by

$$\phi((\tilde{x}, \tilde{y}), L, \alpha_0) = \frac{\partial}{\partial \varepsilon} \alpha_\varepsilon(\tilde{x}, \tilde{y}) \Big|_{\varepsilon=0} = -\left(\frac{\partial}{\partial \alpha} \eta(\alpha, \varepsilon) \right)^{-1} \frac{\partial}{\partial \varepsilon} \eta(\alpha, \varepsilon) \Big|_{\alpha=\alpha_0, \varepsilon=0}, \tag{B.6}$$

from the implicit function theorem. The right-hand side of equation B.6 is equal to equation 4.6 because equalities

$$\frac{\partial}{\partial \alpha} \eta(\alpha_0, 0) = \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho^1(y|x, \alpha_0) L''(-y H_\beta(x) - \alpha_0 y h(x))$$

and

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} \eta(\alpha_0, 0) &= -\frac{\partial}{\partial \varepsilon} \varepsilon \cdot L'(-\tilde{y} H_\beta(\tilde{x}) - \tilde{y} \alpha_0 h(\tilde{x})) \tilde{y} h(\tilde{x}) \Big|_{\varepsilon=0} \\ &= -L'(-\tilde{y} H_\beta(\tilde{x}) - \tilde{y} \alpha_0 h(\tilde{x})) \tilde{y} h(\tilde{x}) \end{aligned}$$

hold. For the last equation, the fact that $\eta(\alpha_0, 0) = 0$ is used.

Appendix C: Proof of Theorem 3

First, we show the existence of loss functions, such as $\rho_L = \rho$, under conditions A1 and A2. Lemma 2 is provided for technical details in the proof of theorem 3.

Lemma 2. *Let $\rho : \mathbb{R} \to \mathbb{R}$ be an odd function, which is once continuously differentiable, and $\rho'(z) > 0$ holds for any $z \in \mathbb{R}$. Then loss function*

$$L(z) = \begin{cases} z & z \geq 0 \\ \int_0^z e^{2\rho(w)} dw & z < 0 \end{cases}$$

satisfies A1, A2, and $\rho_L = \rho$.

Proof. The derivative of L is equal to

$$L'(z) = \begin{cases} 1 & z \geq 0 \\ e^{2\rho(z)} & z < 0 \end{cases}.$$

We see that $\rho_L = \rho$ holds and that $L'(z)$ is continuous. Loss function, L, is a strictly increasing function because $L'(z) > 0$. We can also verify that function L is convex because L' is a nondecreasing function. For $z < 0$, we have $L''(z) = 2\rho'(z)e^{2\rho(z)} > 0$, and this inequality denotes that L is strictly convex for $z < 0$. Thus, condition A1 is verified. Twice continuous differentiability except at zero is also satisfied. The twice differential of L except at $z = 0$ is given as

$$L''(z) = \begin{cases} 0 & z > 0 \\ 2\rho'(z)e^{2\rho(z)} & z < 0 \end{cases},$$

and then, for any $A > 0$, inequality

$$\sup_{z \neq 0, |z| \leq A} |L''(z)| = \sup_{-A \leq z < 0} 2\rho'(z)e^{2\rho(z)} \leq \sup_{-A \leq z \leq 0} 2\rho'(z)e^{2\rho(z)} < \infty$$

holds because of the positivity and the continuity of $2\rho'(z)e^{2\rho(z)}$ on $\mathbb{R}$.

Remark 1. If ρ is twice continuously differentiable, loss function L in lemma 2 satisfies A1, A2, and A3. The proof is almost the same as that in lemma 2. This result is applied to the proof of theorem 4.

Proof of Theorem 3. Suppose that loss function L satisfies A1, A2, and $\rho_L = \rho$. The existence of such a loss function is ensured by lemma 2. The influence function is equal to

$$\phi((\tilde{x}, \tilde{y}), L, \alpha_0) = \frac{L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x}))}{C(L, \rho)} \tilde{y}h(\tilde{x}),$$

that is, given in theorem 2, and then the gross error sensitivity of L is given as

$$\gamma(L, \alpha_0) = \frac{\operatorname{ess\,sup}_{\tilde{x} \in \mathcal{X}, \tilde{y} = \pm 1} L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x}))}{C(L, \rho)}.$$

We assumed $H_\beta(\tilde{x}) + \alpha_0 h(\tilde{x}) \neq 0$ to calculate the influence function in theorem 2, but this constraint does not affect the essential supremum because

of condition 4.4. Note that inequality

$$0 < \operatorname*{ess\,sup}_{\tilde{x}\in\mathcal{X},\tilde{y}=\pm 1} L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x})) < \infty$$

holds because $\sup_{\tilde{x}\in\mathcal{X},\tilde{y}=\pm 1} |-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x})|$ is bounded and L' is positive and continuous. The multiplication of constant to loss functions does not affect the value of the gross error sensitivity. For this reason, we can assume that

$$\operatorname*{ess\,sup}_{\tilde{x}\in\mathcal{X},\tilde{y}=\pm 1} L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x})) = 1 \tag{C.1}$$

holds for arbitrary loss functions.

To derive the most B-robust loss functions, we need to solve the minimization problem of the gross error sensitivity, which is equivalent to the maximization of the functional,

$$\begin{aligned}&\max_L C(L,\rho)\\ &\quad s.t.\quad \rho_L = \rho,\\ &\qquad \operatorname*{ess\,sup}_{\tilde{x}\in\mathcal{X},\tilde{y}=\pm 1} L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha_0 h(\tilde{x})) = 1,\end{aligned}$$

under the condition that L is a loss function satisfying A1 and A2. Since $\rho_L = \rho$ holds, we have

$$L'(z) = L'(-z)e^{2\rho(z)},$$

and the derivative of both sides gives us

$$L''(z) + L''(-z)e^{2\rho(z)} = 2\rho'(z)L'(-z)e^{2\rho(z)}, \tag{C.2}$$

except at $z = 0$. By substituting equation C.2 into $C(L,\rho)$, we obtain

$$C(L,\rho) = 2\int_{\mathcal{X}} \mu(dx)P_\rho^1(1|x,\alpha_0)\rho'(H_\beta(x) + \alpha_0 h(x))L'(-H_\beta(x) - \alpha_0 h(x)),$$

where the integrand takes positive value. For any loss function L with $\rho_L = \rho$, the inequalities

$$L'(-yH_\beta(x) - y\alpha_0 h(x)) \le 1$$

and

$$L'(-yH_\beta(x) - y\alpha_0 h(x)) = L'(yH_\beta(x) + y\alpha_0 h(x))e^{2\rho(-yH_\beta(x)-y\alpha_0 h(x))} \leq e^{2\rho(-yH_\beta(x)-y\alpha_0 h(x))}$$

hold for almost every (x, y) due to equation C.1. Hence, the inequality

$$L'(yH_\beta(x) + y\alpha_0 h(x)) \leq \min\left\{1, e^{2\rho(yH_\beta(x)+y\alpha_0 h(x))}\right\}$$

is valid with probability one. Let us define L_ρ as

$$L_\rho(z) = \int_0^z \min\left\{1, e^{2\rho(w)}\right\} dw. \tag{C.3}$$

We see that $\rho_{L_\rho} = \rho$ and $L'_\rho(z) \leq 1$ holds for any $z \in \mathbb{R}$. Consequently, we obtain the inequality

$$C(L, \alpha_0) \leq C(L_\rho, \alpha_0),$$

because

$$L'(-H_\beta(x) - \alpha_0 h(x)) \leq L'_\rho(-H_\beta(x) - \alpha_0 h(x))$$

holds with probability one. Due to lemma 2, L_ρ satisfies A1 and A2, that is, the influence function for loss function L_ρ exists. Thus, the above argument is valid. Integration C.3 is equal to equation 4.8.

Appendix D: Proof of Theorem 4

The proof consists of four propositions.

Proposition 1. *The optimal weak hypothesis for the boosting algorithm with equation 4.9 does not depend on ε.*

Proof. In the boosting algorithm under loss function L, chosen hypothesis $h^{(1)} \in \mathcal{H}$ is the minimizer of

$$\int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} \widetilde{P}_\varepsilon(y|x, H_\beta) L'(-yH_\beta(x)) I(y \neq h(x))$$
$$= \frac{1}{2} \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} \widetilde{P}_\varepsilon(y|x, H_\beta) L'(-yH_\beta(x)) - \frac{\varepsilon}{2} L'(-\tilde{y}H_\beta(\tilde{x}))\tilde{y}h(\tilde{x}), \tag{D.1}$$

where we used equality

$$\int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L'(-yH_\beta(x)) yh(x) = 0 \tag{D.2}$$

which comes from the optimality of H_β in $\mathcal{M}[\rho]$. Due to equation D.1, $h^{(1)}$ does not depend on the value of $\varepsilon > 0$ but only on outlier $(\tilde{x}, \tilde{y})$. Thus, for given $(\tilde{x}, \tilde{y})$, hypothesis $h^{(1)}$ is fixed.

Proposition 2. *Suppose that $H_\beta(\tilde{x}) \neq 0$. Let $\eta(\alpha, \varepsilon)$ be*

$$\begin{aligned}\eta(\alpha, \varepsilon) = &-(1-\varepsilon) \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L'(-yH_\beta(x) - y\alpha h^{(1)}(x)) yh(x) \\ &- \varepsilon L'(-\tilde{y}H_\beta(\tilde{x}) - \tilde{y}\alpha h^{(1)}(\tilde{x}))\tilde{y}h(\tilde{x}),\end{aligned}$$

as in the proof of theorem 2. Then $\eta(\alpha, \varepsilon)$ is twice continuously differentiable.

Proof. Under condition A1, $\eta(\alpha, \varepsilon)$ is identical to the derivative of $R_L(\widetilde{P}_\varepsilon, H_\beta + \alpha h^{(1)})$ with respect to α for $0 \leq \varepsilon \leq 1$. Applying the same argument in the proof of theorem 2, we can derive the change-of-risk function.

We will show that there exist $\delta_{\tilde{x}} > 0$ and $\varepsilon_0 > 0$ such that $\eta(\alpha, \varepsilon)$ is twice continuously differentiable on $(-\delta_{\tilde{x}}, \delta_{\tilde{x}}) \times (-\varepsilon_0, \varepsilon_0)$. The differentiability of $\eta(\alpha, \varepsilon)$ with respect to ε is clear, and any small positive constant ε_0 is valid. Next, we study the differentiability with respect to α. If we set $\delta_{\tilde{x}} = \min\{|H_\beta(\tilde{x})|, \delta_0\}$, where δ_0 is given in equation 4.13, then $H_\beta(\tilde{x}) + \alpha h^{(1)}(\tilde{x}) \neq 0$ for $\alpha \in (-\delta_{\tilde{x}}, \delta_{\tilde{x}})$, and it is clear that the second term of $\eta(\alpha, \varepsilon)$ is twice continuously differentiable under A3. Note that for any $\alpha \in (-\delta_{\tilde{x}}, \delta_{\tilde{x}})$, $H_\beta(x) + \alpha h^{(1)}(x)$ is not equal to zero almost everywhere because we have

$$\begin{aligned}\mu(\{x|H_\beta(x) + \alpha h^{(1)}(x) = 0\}) &= \mu(\{x|H_\beta(x) + \alpha = 0, h^{(1)}(x) = 1\}) \\ &\quad + \mu(\{x|H_\beta(x) - \alpha = 0, h^{(1)}(x) = -1\}) \\ &\leq \mu(\{x|H_\beta(x) + \alpha = 0\}) + \mu(\{x|H_\beta(x) - \alpha = 0\}) \\ &= 0\end{aligned}$$

from equation 4.13. Then integrations

$$\int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L''(-yH_\beta(x) - y\alpha h^{(1)}(x)) \tag{D.3}$$

and

$$\int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L'''(-yH_\beta(x) - y\alpha h^{(1)}(x)) y h^{(1)}(x) \tag{D.4}$$

are well defined under the condition that L satisfies A2 and A3. Under A2, boundedness

$$\begin{aligned} |L''(-yH_\beta(x) - y\alpha h^{(1)}(x))| &\le \sup_{x\in\mathcal{X}, y=\pm 1} |L''(-yH_\beta(x) - y\alpha h^{(1)}(x))| \\ &\le \sup_{z\neq 0, |z|\le \|\beta\|_1 + \delta_{\tilde{x}}} |L''(z)| \\ &< \infty \end{aligned}$$

holds almost everywhere for $\alpha \in (-\delta_{\tilde{x}}, \delta_{\tilde{x}})$. Thus, by Lebesgue's bounded convergence theorem, the exchange of integration and differentiation

$$\begin{aligned} &-\frac{\partial}{\partial \alpha} \int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P_\rho(y|x, H_\beta) L'(-yH_\beta(x) - y\alpha h^{(1)}(x)) y h^{(1)}(x) \\ &\quad = \int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P_\rho(y|x, H_\beta) L''(-yH_\beta(x) - y\alpha h^{(1)}(x)) \end{aligned}$$

is valid and the continuity of equation D.3 in α also holds in a similar fashion. Likewise, from A3, boundedness

$$\begin{aligned} |L'''(-yH_\beta(x) - y\alpha h^{(1)}(x)) y h^{(1)}(x)| &\le \sup_{x\in\mathcal{X}, y=\pm 1} |L'''(-yH_\beta(x) - y\alpha h^{(1)}(x))| \\ &\le \sup_{z\neq 0, |z|\le \|\beta\|_1 + \delta_{\tilde{x}}} |L'''(z)| \\ &< \infty \end{aligned}$$

holds almost everywhere for $\alpha \in (-\delta_{\tilde{x}}, \delta_{\tilde{x}})$. Then the exchange of integration and differentiation,

$$\begin{aligned} &-\frac{\partial}{\partial \alpha} \int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P_\rho(y|x, H_\beta) L''(-yH_\beta(x) - y\alpha h^{(1)}(x)) \\ &\quad = \int_{\mathcal{X}} \sum_{y=\pm 1} \mu(dx) P_\rho(y|x, H_\beta) L'''(-yH_\beta(x) - y\alpha h^{(1)}(x)) y h^{(1)}(x), \end{aligned}$$

is valid, and we see that the continuity of equation D.4 in α also holds in a similar fashion. Consequently, we find that $\eta(\alpha, \varepsilon)$ is twice continuously differentiable on $(-\delta_{\tilde{x}}, \delta_{\tilde{x}}) \times (-\varepsilon_0, \varepsilon_0)$.

Proposition 3. *The change-of-risk function, $\psi_{L_0}((\tilde{x}, \tilde{y}), L, H)$, is given as*

$$\psi_{L_0}((\tilde{x}, \tilde{y}), L, H_\beta) = \operatorname*{ess\,sup}_{\tilde{x} \in \mathcal{X}} L'(|H_\beta(\tilde{x})|)^2 \cdot \frac{1}{2C(L, \rho)^2} \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L_0''(-yH_\beta(x)),$$

where the definition of $C(L, \rho)$ is given by equation 4.7.

Proof. The positivity of $\frac{\partial \eta}{\partial \alpha}(0, 0)$ can be proved in a similar way to equation B.5. Thus, applying the implicit function theorem, we find that there exists a parameter α_ε satisfying equation

$$\eta(\alpha_\varepsilon, \varepsilon) = 0$$

in a vicinity of $\varepsilon = 0$. Note that the function α_ε is also twice continuously differentiable with respect to ε on $(-2\varepsilon_1, 2\varepsilon_1)$, where ε_1 satisfies $0 < 2\varepsilon_1 < \varepsilon_0$. The inequality $|\alpha_\varepsilon| < \delta_{\tilde{x}}$ holds for $\varepsilon \in (-2\varepsilon_1, 2\varepsilon_1)$. Since $\frac{\partial \alpha_\varepsilon}{\partial \varepsilon}$ and $\frac{\partial^2 \alpha_\varepsilon}{\partial \varepsilon^2}$ are continuous, there are positive constants, C_1 and C_2, such that $|\frac{\partial \alpha_\varepsilon}{\partial \varepsilon}| \le C_1$ and $|\frac{\partial^2 \alpha_\varepsilon}{\partial \varepsilon^2}| \le C_2$ hold for any $\varepsilon \in (-\varepsilon_1, \varepsilon_1)$. Predictor $\widetilde{H}_{L,\varepsilon}$ derived from loss function L under the contaminated probability $\widetilde{P}_\varepsilon$ is given as

$$\widetilde{H}_{L,\varepsilon} = H_\beta + \alpha_\varepsilon h^{(1)}.$$

Note that α_ε depends on ε and $(\tilde{x}, \tilde{y})$ while $h^{(1)}$ depends only on $(\tilde{x}, \tilde{y})$.

By using Lebesgue's bounded convergence theorem, we find that the twice-continuous differentiability of $R_{L_0}(P, \widetilde{H}_{L,\varepsilon})$ with respect to ε on $(-\varepsilon_1, \varepsilon_1)$ also holds under A1 and A2 for L_0, because the boundedness of the differentials is satisfied almost everywhere. Indeed, the following inequalities hold with probability one:

$$\begin{aligned} \left| \frac{\partial}{\partial \varepsilon} L_0(-yH_\beta(x) - y\alpha_\varepsilon h^{(1)}(x)) \right| &= \left| L_0'(-yH_\beta(x) - y\alpha_\varepsilon h^{(1)}(x)) \frac{\partial \alpha_\varepsilon}{\partial \varepsilon} \right| \\ &\le C_1 \sup_{|z| \le \|\beta\|_1 + \delta_{\tilde{x}}} |L_0'(z)| \\ &\le C_1 L_0'(\|\beta\|_1 + \delta_{\tilde{x}}), \\ &< \infty, \end{aligned} \tag{D.5}$$

and

$$\left|\frac{\partial^2}{\partial\varepsilon^2}L_0(-yH_\beta(x)-y\alpha_\varepsilon h^{(1)}(x))\right| = \left|L_0''(-yH_\beta(x)-y\alpha_\varepsilon h^{(1)}(x))\left(\frac{\partial\alpha_\varepsilon}{\partial\varepsilon}\right)^2 - L_0'(-yH_\beta(x)-y\alpha_\varepsilon h^{(1)}(x))yh^{(1)}(x)\frac{\partial^2\alpha_\varepsilon}{\partial\varepsilon^2}\right|$$
$$\leq C_1^2 \sup_{z\neq 0,|z|\leq\|\beta\|_1+\delta_{\tilde{x}}}|L_0''(z)| + C_2 \sup_{z\neq 0,|z|\leq\|\beta\|_1+\delta_{\tilde{x}}}|L_0'(z)| < \infty. \tag{D.6}$$

Consequently, the Taylor expansion of risk $R_{L_0}(P, \widetilde{H}_{L,\varepsilon})$ exists up to the order of two,

$$\begin{aligned} R_{L_0}(P, \widetilde{H}_{L,\varepsilon}) &= R_{L_0}(P, H_\beta) - \varepsilon\int_{\mathcal{X}}\mu(dx)\sum_{y=\pm1}P_\rho(y|x, H_\beta)L_0'(-yH_\beta(x))yh^{(1)}(x)\frac{\partial\alpha_\varepsilon}{\partial\varepsilon} \\ &+ \frac{\varepsilon^2}{2}\int_{\mathcal{X}}\mu(dx)\sum_{y=\pm1}P_\rho(y|x, H_\beta)L_0''\big(-yH_\beta(x)-y\alpha_{\theta\varepsilon}h^{(1)}\big)\left(\frac{\partial\alpha_\varepsilon}{\partial\varepsilon}(\theta\varepsilon)\right)^2 \\ &- \frac{\varepsilon^2}{2}\int_{\mathcal{X}}\mu(dx)\sum_{y=\pm1}P_\rho(y|x, H_\beta)L_0'\big(-yH_\beta(x)-y\alpha_{\theta\varepsilon}h^{(1)}\big)yh^{(1)}\frac{\partial^2\alpha_\varepsilon}{\partial\varepsilon^2}(\theta\varepsilon), \end{aligned}$$

where $\theta \in [0, 1]$. The exchange of limitation $\varepsilon \to 0$ and the integration is valid because of equations D.5 and D.6. Since equation D.2 holds even for $L = L_0$ and equality

$$\left.\frac{\partial\alpha_\varepsilon}{\partial\varepsilon}\right|_{\varepsilon=0} = \phi((\tilde{x}, \tilde{y}), L, 0)$$

holds, we have

$$\psi_{L_0}((\tilde{x}, \tilde{y}), L, H_\beta) = \frac{1}{2}\phi((\tilde{x}, \tilde{y}), L, 0)^2\int_{\mathcal{X}}\mu(dx)\sum_{y=\pm1}P_\rho(y|x, H_\beta)L_0''(-yH_\beta(x)),$$

where $\phi((\tilde{x}, \tilde{y}), L, 0)$ is the influence function at $\alpha = 0$ for model $\mathcal{M}_1[\rho, h^{(1)}, H_\beta]$. Note that influence function $\phi((\tilde{x}, \tilde{y}), L, 0)$ does not depend on the choice of hypothesis $h^{(1)}$.

Therefore, change-of-risk sensitivity is equal to

$$\gamma_{L_0}(L, H_\beta) = \operatorname*{ess\,sup}_{\tilde{x}\in\mathcal{X},\tilde{y}=\pm 1} \frac{1}{2}\phi((\tilde{x},\tilde{y}), L, 0)^2 \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L_0''(-yH_\beta(x))$$

$$= \frac{\operatorname{ess\,sup}_{\tilde{x}\in\mathcal{X}} L'(|H_\beta(\tilde{x})|)^2}{2C(L,\rho)^2} \int_{\mathcal{X}} \mu(dx) \sum_{y=\pm 1} P_\rho(y|x, H_\beta) L_0''(-yH_\beta(x)).$$

We have assumed that $H_\beta(\tilde{x}) \neq 0$ to calculate change-of-risk function, but this constraint does not affect the essential supremum of $L'(|H_\beta(\tilde{x})|)$ because

$$\mu(\{x|H_\beta(x) = 0\}) = 0$$

holds from equation 4.13.

Proposition 4. *The loss function L_ρ minimizes the change-of-risk sensitivity $\gamma_{L_0}(L, H_\beta)$.*

Proof. We solve the minimization problem analogous to the proof of theorem 3,

$$\begin{aligned} \max_L \quad & C(L,\rho) \\ s.t. \quad & \rho_L = \rho, \\ & \operatorname*{ess\,sup}_{\tilde{x}\in\mathcal{X}} L'(|H_\beta(\tilde{x})|) = 1, \end{aligned}$$

under the condition that L is a loss function with conditions A1, A2, and A3. Due to the constraint of the minimization problem, loss function L satisfies the following inequality with probability one,

$$\begin{aligned} & L'(H(x)) \leq 1, \\ & L'(H(x)) = L'(-H(x))e^{2\rho(H(x))} \leq e^{2\rho(H(x))}, \end{aligned}$$

and then,

$$L'(H(x)) \leq \min\left\{1, e^{2\rho(H(x))}\right\} = L'_\rho(H(x)) \tag{D.7}$$

holds almost everywhere with respect to measure μ on $\mathcal{X}$. Since ρ is twice continuously differentiable, we find that L_ρ is three times continuously differentiable except at zero. We can verify that loss function L_ρ satisfies conditions A1, A2, and A3 as pointed out in remark 1 in appendix C. Under the imposed constraints, inequality $C(L,\rho) \leq C(L_\rho,\rho)$ follows from equation D.7 in a similar way of the proof in theorem 3. Consequently, we find that L_ρ minimizes the change-of-risk sensitivity, $\gamma_{L_0}(L, H_\beta)$.

Appendix E: Results of Experiments

Table 5: Error and Log-Loss×10 on the Test Set of Banana Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Banana: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	28.4 ± 1.9	–	29.1 ± 2.2	–	28.9 ± 2.2	–
LB	27.7 ± 1.7	0.00	28.5 ± 2.3	0.02	28.7 ± 2.0	0.15
MB	26.0 ± 2.2	0.00	27.0 ± 2.5	0.00	27.4 ± 2.6	0.00
Eta	28.4 ± 2.1	0.54	28.8 ± 2.3	0.13	29.2 ± 2.2	0.87
r-eta	$\bullet 25.5 \pm 2.1$	0.00	$\bullet 26.6 \pm 2.5$	0.00	$\bullet 26.5 \pm 2.1$	0.00
AB$_R$	27.7 ± 1.8	0.00	28.4 ± 2.0	0.01	28.4 ± 2.0	0.04
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	38.8 ± 2.8	–	38.5 ± 2.1	–	38.7 ± 2.7	–
LB	38.7 ± 2.6	0.26	38.6 ± 2.6	0.67	38.6 ± 2.5	0.39
MB	38.2 ± 2.9	0.04	$\bullet 37.8 \pm 2.4$	0.01	38.0 ± 2.5	0.01
Eta	38.6 ± 2.4	0.20	38.2 ± 2.0	0.13	38.6 ± 2.6	0.42
r-eta	$\bullet 37.5 \pm 1.8$	0.00	37.8 ± 1.9	0.00	$\bullet 37.5 \pm 2.4$	0.00
AB$_R$	38.9 ± 2.6	0.64	38.6 ± 2.3	0.66	38.6 ± 2.7	0.38
Banana: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.01 ± 0.18	–	6.03 ± 0.15	–	6.03 ± 0.12	–
LB	$\bullet 5.99 \pm 0.15$	0.12	6.01 ± 0.14	0.05	$\bullet 6.00 \pm 0.13$	0.03
MB	6.03 ± 0.17	0.82	6.02 ± 0.13	0.30	6.01 ± 0.14	0.14
Eta	6.04 ± 0.25	0.82	6.09 ± 0.34	0.95	6.14 ± 0.41	0.99
r-eta	5.99 ± 0.17	0.17	$\bullet 5.99 \pm 0.12$	0.00	6.02 ± 0.17	0.18
AB$_R$	6.03 ± 0.17	0.76	6.04 ± 0.17	0.59	6.08 ± 0.22	0.98
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.76 ± 0.11	–	6.77 ± 0.09	–	6.78 ± 0.11	–
LB	6.76 ± 0.11	0.41	6.77 ± 0.11	0.51	6.77 ± 0.10	0.28
MB	$\bullet 6.74 \pm 0.10$	0.01	$\bullet 6.75 \pm 0.10$	0.04	$\bullet 6.75 \pm 0.09$	0.01
Eta	6.82 ± 0.33	0.95	6.80 ± 0.26	0.91	6.91 ± 0.43	0.99
r-eta	6.74 ± 0.10	0.01	6.76 ± 0.11	0.26	6.75 ± 0.11	0.03
AB$_R$	6.79 ± 0.15	0.93	6.82 ± 0.14	0.99	6.82 ± 0.16	0.99

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 6: Error and Log-Loss×10 on the Test Set of Breast-Cancer Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Breast-Cancer: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	28.1 ± 4.5	–	27.9 ± 4.5	–	27.5 ± 4.4	–
LB	27.6 ± 4.4	0.14	27.3 ± 4.4	0.10	27.6 ± 4.1	0.58
MB	27.4 ± 4.3	0.06	27.3 ± 4.1	0.07	27.7 ± 4.6	0.74
Eta	27.6 ± 4.2	0.12	28.4 ± 4.3	0.85	28.0 ± 4.7	0.89
r-eta	$\bullet 27.0 \pm 4.8$	0.01	$\bullet 27.3 \pm 4.4$	0.11	27.3 ± 4.9	0.28
AB$_R$	27.6 ± 4.7	0.17	27.7 ± 4.3	0.35	$\bullet 27.2 \pm 4.9$	0.24
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	37.6 ± 5.0	–	38.2 ± 5.2	–	38.3 ± 5.8	–
LB	37.4 ± 5.6	0.30	38.2 ± 5.6	0.47	38.5 ± 5.6	0.75
MB	36.8 ± 5.8	0.03	$\bullet 37.7 \pm 5.3$	0.13	37.8 ± 6.2	0.15
Eta	37.3 ± 5.9	0.23	38.6 ± 5.2	0.82	$\bullet 37.7 \pm 5.3$	0.12
r-eta	$\bullet 36.5 \pm 5.9$	0.01	38.6 ± 5.3	0.87	37.9 ± 5.2	0.17
AB$_R$	37.7 ± 5.6	0.64	38.6 ± 5.4	0.79	37.8 ± 5.9	0.16
Breast-Cancer: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	5.87 ± 0.69	–	5.81 ± 0.63	–	5.77 ± 0.66	–
LB	5.81 ± 0.70	0.12	5.82 ± 0.64	0.57	5.78 ± 0.61	0.59
MB	5.81 ± 0.67	0.15	5.77 ± 0.60	0.22	5.86 ± 0.65	0.96
Eta	6.18 ± 2.15	0.94	5.85 ± 0.89	0.71	5.86 ± 1.04	0.82
r-eta	$\bullet 5.74 \pm 0.65$	0.00	$\bullet 5.76 \pm 0.68$	0.17	$\bullet 5.74 \pm 0.64$	0.22
AB$_R$	5.98 ± 0.66	0.98	5.82 ± 0.69	0.60	5.92 ± 0.88	0.99
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.71 ± 0.46	–	6.78 ± 0.53	–	6.74 ± 0.50	–
LB	6.70 ± 0.49	0.35	6.81 ± 0.47	0.76	6.81 ± 0.53	0.98
MB	6.66 ± 0.46	0.07	6.78 ± 0.46	0.48	$\bullet 6.72 \pm 0.48$	0.30
Eta	6.68 ± 0.50	0.24	6.79 ± 0.57	0.59	6.84 ± 1.01	0.86
r-eta	$\bullet 6.63 \pm 0.41$	0.01	$\bullet 6.75 \pm 0.53$	0.20	6.79 ± 0.56	0.91
AB$_R$	6.76 ± 0.51	0.93	6.89 ± 0.63	0.99	6.86 ± 0.60	0.99

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 7: Error and Log-Loss×10 on the Test Set of Diabetes Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Diabetes: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	25.2 ± 1.8	–	25.5 ± 1.8	–	25.6 ± 1.9	–
LB	25.3 ± 2.1	0.70	25.4 ± 1.8	0.39	25.7 ± 2.1	0.57
MB	25.5 ± 1.8	0.95	25.8 ± 1.8	0.95	25.9 ± 1.9	0.86
Eta	25.7 ± 2.1	0.97	25.6 ± 1.8	0.80	25.6 ± 2.1	0.40
r-eta	25.7 ± 1.9	0.99	25.8 ± 1.9	0.94	25.6 ± 1.9	0.37
AB$_R$	$\bullet 25.0 \pm 1.7$	0.22	$\bullet 25.1 \pm 1.8$	0.02	$\bullet 25.1 \pm 1.9$	0.02
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	36.6 ± 2.8	–	36.9 ± 2.9	–	37.1 ± 2.9	–
LB	37.0 ± 2.8	0.93	37.2 ± 2.7	0.86	37.0 ± 2.9	0.26
MB	36.6 ± 2.6	0.58	36.9 ± 3.0	0.42	37.4 ± 3.0	0.75
Eta	36.8 ± 2.8	0.75	36.9 ± 3.3	0.43	37.1 ± 2.8	0.40
r-eta	36.8 ± 2.9	0.76	$\bullet 36.9 \pm 2.8$	0.40	$\bullet 37.0 \pm 2.4$	0.27
AB$_R$	$\bullet 36.4 \pm 3.0$	0.31	37.2 ± 3.1	0.77	37.1 ± 2.5	0.39
Diabetes: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	5.29 ± 0.30	–	5.34 ± 0.31	–	5.30 ± 0.31	–
LB	5.26 ± 0.33	0.12	5.26 ± 0.33	0.01	5.29 ± 0.32	0.30
MB	5.26 ± 0.29	0.13	5.28 ± 0.29	0.03	5.28 ± 0.27	0.20
Eta	5.40 ± 0.50	0.99	5.40 ± 0.35	0.98	5.40 ± 0.75	0.90
r-eta	$\bullet 5.25 \pm 0.29$	0.04	$\bullet 5.25 \pm 0.28$	0.00	$\bullet 5.27 \pm 0.29$	0.10
AB$_R$	5.38 ± 0.33	0.99	5.39 ± 0.34	0.93	5.39 ± 0.38	0.99
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.61 ± 0.30	–	6.64 ± 0.26	–	6.62 ± 0.25	–
LB	6.63 ± 0.28	0.75	6.65 ± 0.29	0.68	6.62 ± 0.27	0.59
MB	6.58 ± 0.27	0.14	$\bullet 6.60 \pm 0.28$	0.04	$\bullet 6.60 \pm 0.23$	0.24
Eta	6.68 ± 0.62	0.90	6.65 ± 0.38	0.65	6.68 ± 0.38	0.96
r-eta	$\bullet 6.57 \pm 0.33$	0.06	6.62 ± 0.25	0.14	6.62 ± 0.24	0.50
AB$_R$	6.61 ± 0.30	0.52	6.66 ± 0.28	0.90	6.67 ± 0.27	0.99

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 8: Error and Log-Loss×10 on the Test Set of German Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

German: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	25.7 ± 2.6	–	25.9 ± 2.5	–	26.2 ± 2.7	–
LB	25.4 ± 2.5	0.13	25.7 ± 2.4	0.26	26.0 ± 2.6	0.20
MB	25.5 ± 2.5	0.25	25.6 ± 2.4	0.18	25.6 ± 2.6	0.01
Eta	26.0 ± 2.9	0.87	25.9 ± 2.5	0.56	26.8 ± 2.8	0.99
r-eta	$\bullet 25.3 \pm 2.6$	0.06	$\bullet 25.3 \pm 2.4$	0.01	25.9 ± 2.7	0.09
AB$_R$	25.5 ± 2.5	0.21	25.6 ± 2.7	0.06	$\bullet 25.5 \pm 2.7$	0.00
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	36.8 ± 3.0	–	37.5 ± 2.8	–	$\bullet 37.2 \pm 2.8$	–
LB	$\bullet 36.6 \pm 2.7$	0.28	37.3 ± 2.7	0.16	38.0 ± 2.9	0.99
MB	36.8 ± 2.9	0.51	$\bullet 37.1 \pm 2.7$	0.05	37.5 ± 2.8	0.89
Eta	36.8 ± 3.1	0.53	37.7 ± 2.8	0.66	37.9 ± 2.9	0.99
r-eta	36.8 ± 3.0	0.47	37.1 ± 2.8	0.04	37.5 ± 2.9	0.85
AB$_R$	36.9 ± 2.9	0.63	37.6 ± 2.6	0.61	37.6 ± 2.6	0.95
German: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	5.29 ± 0.32	–	5.24 ± 0.33	–	5.28 ± 0.32	–
LB	5.22 ± 0.32	0.00	5.22 ± 0.31	0.20	5.19 ± 0.30	0.00
MB	$\bullet 5.20 \pm 0.31$	0.00	5.20 ± 0.28	0.02	$\bullet 5.16 \pm 0.27$	0.00
Eta	5.34 ± 0.39	0.98	5.31 ± 0.38	0.99	5.33 ± 0.35	0.99
r-eta	5.20 ± 0.34	0.00	$\bullet 5.19 \pm 0.30$	0.01	5.21 ± 0.30	0.00
AB$_R$	5.31 ± 0.34	0.83	5.29 ± 0.34	0.98	5.32 ± 0.35	0.95
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.51 ± 0.23	–	6.55 ± 0.20	–	6.54 ± 0.22	–
LB	$\bullet 6.48 \pm 0.22$	0.05	6.55 ± 0.21	0.48	6.54 ± 0.22	0.60
MB	6.50 ± 0.22	0.27	6.54 ± 0.20	0.24	6.51 ± 0.21	0.03
Eta	6.50 ± 0.25	0.32	6.56 ± 0.21	0.69	6.54 ± 0.21	0.56
r-eta	6.49 ± 0.23	0.15	$\bullet 6.53 \pm 0.21$	0.08	$\bullet 6.51 \pm 0.21$	0.01
AB$_R$	6.52 ± 0.23	0.60	6.58 ± 0.23	0.96	6.58 ± 0.22	0.98

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 9: Error and Log-Loss×10 on the Test Set of Heart Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Heart: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	19.6 ± 4.0	–	19.0 ± 4.2	–	19.9 ± 4.4	–
LB	19.4 ± 4.2	0.34	19.4 ± 4.0	0.8	19.4 ± 4.1	0.07
MB	19.5 ± 4.6	0.47	19.4 ± 3.8	0.77	18.9 ± 4.6	0.03
Eta	19.4 ± 4.2	0.3	19.2 ± 3.9	0.63	19.7 ± 4.8	0.33
r-eta	$\bullet 18.7 \pm 4.0$	0.02	18.7 ± 4.0	0.26	19.5 ± 4.1	0.19
AB$_R$	19.1 ± 4.1	0.12	$\bullet 18.5 \pm 4.0$	0.10	$\bullet 18.9 \pm 3.8$	0.01
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	$\bullet 35.2 \pm 5.2$	–	36.0 ± 5.2	–	$\bullet 35.6 \pm 5.4$	–
LB	35.3 ± 5.1	0.59	36.1 ± 5.0	0.64	35.7 ± 5.5	0.55
MB	36.0 ± 5.5	0.94	$\bullet 35.2 \pm 5.3$	0.04	35.7 ± 5.9	0.54
Eta	35.2 ± 5.6	0.50	35.4 ± 5.7	0.11	36.2 ± 5.2	0.90
r-eta	35.6 ± 5.1	0.79	35.4 ± 5.0	0.11	35.9 ± 5.2	0.68
AB$_R$	35.6 ± 5.1	0.80	35.6 ± 5.6	0.23	35.9 ± 4.8	0.72
Heart: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	5.02 ± 1.01	–	4.79 ± 0.92	–	4.89 ± 0.92	–
LB	4.87 ± 0.98	0.03	4.68 ± 0.87	0.09	4.74 ± 0.88	0.02
MB	$\bullet 4.80 \pm 0.94$	0.01	$\bullet 4.63 \pm 0.88$	0.03	4.63 ± 0.93	0.00
Eta	5.14 ± 1.27	0.87	5.02 ± 1.37	0.97	5.04 ± 1.15	0.91
r-eta	4.84 ± 1.04	0.03	4.70 ± 0.95	0.16	$\bullet 4.58 \pm 0.82$	0.00
AB$_R$	5.09 ± 1.06	0.77	5.06 ± 1.22	0.99	4.97 ± 1.04	0.80
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.74 ± 0.71	–	6.74 ± 0.65	–	$\bullet 6.76 \pm 0.70$	–
LB	$\bullet 6.72 \pm 0.63$	0.35	6.78 ± 0.70	0.73	6.78 ± 0.66	0.56
MB	6.73 ± 0.67	0.43	$\bullet 6.72 \pm 0.66$	0.32	6.80 ± 0.64	0.69
Eta	6.86 ± 1.35	0.83	6.83 ± 0.91	0.91	6.96 ± 1.36	0.92
r-eta	6.83 ± 0.80	0.93	6.74 ± 0.73	0.47	6.79 ± 0.66	0.63
AB$_R$	6.85 ± 0.84	0.95	6.94 ± 0.94	0.99	6.86 ± 0.70	0.90

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 10: Error and Log-Loss×10 on the Test Set of Image Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Image: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	4.4 ± 0.6	–	4.4 ± 0.7	–	5.4 ± 0.9	–
LB	$\bullet 4.1 \pm 0.6$	0.06	4.6 ± 0.6	0.89	4.4 ± 0.6	0.00
MB	4.2 ± 0.6	0.03	4.3 ± 0.6	0.31	4.8 ± 0.8	0.01
Eta	6.2 ± 0.9	0.99	6.2 ± 1.0	0.99	6.4 ± 1.4	0.99
r-eta	4.2 ± 0.6	0.06	$\bullet 4.2 \pm 0.6$	0.21	$\bullet 4.2 \pm 0.8$	0.00
AB$_R$	4.4 ± 0.6	0.60	4.4 ± 0.8	0.60	4.7 ± 0.8	0.00
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	25.7 ± 2.4	–	25.6 ± 2.0	–	25.5 ± 1.8	–
LB	25.1 ± 1.9	0.05	25.6 ± 1.8	0.60	24.9 ± 1.7	0.08
MB	24.7 ± 1.9	0.01	$\bullet 24.4 \pm 1.5$	0.00	24.7 ± 2.0	0.03
Eta	25.1 ± 1.9	0.12	25.7 ± 1.7	0.62	25.9 ± 2.0	0.88
r-eta	$\bullet 24.6 \pm 1.5$	0.01	24.6 ± 1.7	0.00	24.7 ± 1.7	0.01
AB$_R$	24.7 ± 2.0	0.02	24.6 ± 1.5	0.01	$\bullet 24.5 \pm 1.2$	0.01
Image: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	1.33 ± 0.19	–	1.32 ± 0.22	–	1.46 ± 0.12	–
LB	1.43 ± 0.12	0.99	1.40 ± 0.15	0.96	1.55 ± 0.14	0.99
MB	1.62 ± 0.10	0.99	1.59 ± 0.13	0.99	1.67 ± 0.11	0.99
Eta	1.97 ± 0.19	0.99	1.86 ± 0.35	0.99	2.07 ± 0.41	0.99
r-eta	1.60 ± 0.12	0.99	1.57 ± 0.14	0.99	1.72 ± 0.14	0.99
AB$_R$	$\bullet 1.32 \pm 0.22$	0.34	$\bullet 1.28 \pm 0.19$	0.16	$\bullet 1.38 \pm 0.20$	0.03
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	5.71 ± 0.18	–	5.71 ± 0.14	–	5.69 ± 0.15	–
LB	5.68 ± 0.16	0.12	5.68 ± 0.14	0.08	5.66 ± 0.16	0.09
MB	5.65 ± 0.16	0.02	$\bullet 5.65 \pm 0.12$	0.01	$\bullet 5.63 \pm 0.14$	0.01
Eta	5.72 ± 0.17	0.66	5.77 ± 0.15	0.96	5.77 ± 0.19	0.99
r-eta	$\bullet 5.64 \pm 0.17$	0.01	5.67 ± 0.14	0.08	5.67 ± 0.16	0.18
AB$_R$	5.71 ± 0.17	0.56	5.70 ± 0.16	0.34	5.69 ± 0.17	0.41

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 11: Error and Log-Loss×10 on the Test Set of Ringnorm Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Ringnorm: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	8.7 ± 1.2	–	11.7 ± 1.7	–	11.8 ± 1.9	–
LB	8.6 ± 1.3	0.34	11.2 ± 1.6	0.00	11.5 ± 1.8	0.05
MB	8.7 ± 1.1	0.43	11.5 ± 1.7	0.08	11.6 ± 2.2	0.27
Eta	10.8 ± 1.9	0.99	13.4 ± 2.3	0.99	14.2 ± 2.7	0.99
r-eta	$\bullet 8.2 \pm 1.1$	0.00	$\bullet 11.0 \pm 1.6$	0.00	$\bullet 11.4 \pm 1.8$	0.02
AB$_R$	8.5 ± 1.0	0.14	11.4 ± 1.7	0.05	11.7 ± 1.6	0.33
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	31.0 ± 2.2	–	33.2 ± 2.3	–	33.5 ± 2.0	–
LB	31.0 ± 2.0	0.53	32.7 ± 2.0	0.03	33.4 ± 2.2	0.38
MB	31.2 ± 2.2	0.76	32.9 ± 2.2	0.18	33.5 ± 2.2	0.53
Eta	31.4 ± 2.1	0.92	33.4 ± 2.1	0.84	33.9 ± 2.2	0.95
r-eta	30.8 ± 1.7	0.19	32.8 ± 2.1	0.06	33.2 ± 2.1	0.11
AB$_R$	$\bullet 30.5 \pm 2.0$	0.01	$\bullet 32.5 \pm 2.3$	0.01	$\bullet 33.0 \pm 2.4$	0.02
Ringnorm: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	1.98 ± 0.21	–	2.91 ± 0.42	–	3.04 ± 0.40	–
LB	$\bullet 1.97 \pm 0.12$	0.33	2.70 ± 0.39	0.00	2.73 ± 0.35	0.00
MB	2.10 ± 0.13	0.99	2.71 ± 0.32	0.00	2.76 ± 0.30	0.00
Eta	2.75 ± 0.80	0.99	3.54 ± 1.10	0.99	3.67 ± 0.65	0.99
r-eta	2.07 ± 0.11	0.99	$\bullet 2.64 \pm 0.25$	0.00	$\bullet 2.70 \pm 0.22$	0.00
AB$_R$	1.99 ± 0.20	0.68	2.84 ± 0.46	0.08	3.03 ± 0.43	0.42
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.30 ± 0.21	–	6.52 ± 0.22	–	6.56 ± 0.24	–
LB	6.26 ± 0.22	0.06	6.44 ± 0.22	0.00	6.51 ± 0.24	0.02
MB	$\bullet 6.22 \pm 0.24$	0.00	6.43 ± 0.23	0.00	6.48 ± 0.22	0.00
Eta	6.45 ± 0.58	0.99	6.49 ± 0.23	0.09	6.60 ± 0.32	0.86
r-eta	6.23 ± 0.24	0.00	$\bullet 6.42 \pm 0.27$	0.00	$\bullet 6.42 \pm 0.24$	0.00
AB$_R$	6.35 ± 0.25	0.93	6.58 ± 0.29	0.97	6.59 ± 0.24	0.84

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 12: Error and Log-Loss×10 on the Test Set of Flare-Solar Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Flare-Solar: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	32.5 ± 1.9	–	33.0 ± 2.5	–	32.9 ± 1.8	–
LB	32.6 ± 2.3	0.79	32.7 ± 1.6	0.07	32.9 ± 1.7	0.58
MB	32.6 ± 1.9	0.92	$\bullet 32.6 \pm 1.8$	0.03	32.9 ± 1.8	0.38
Eta	$\bullet 32.4 \pm 1.8$	0.09	32.8 ± 1.9	0.17	$\bullet 32.7 \pm 1.8$	0.00
r-eta	32.4 ± 1.7	0.41	32.6 ± 1.7	0.03	32.8 ± 1.9	0.04
AB$_R$	32.8 ± 2.6	0.93	32.8 ± 1.8	0.16	32.8 ± 1.9	0.03
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	40.9 ± 3.0	–	41.0 ± 3.0	–	40.8 ± 3.2	–
LB	40.8 ± 3.1	0.44	41.1 ± 3.0	0.65	$\bullet 40.8 \pm 3.2$	0.44
MB	$\bullet 40.8 \pm 2.9$	0.36	$\bullet 40.6 \pm 2.8$	0.11	40.9 ± 3.4	0.61
Eta	41.2 ± 3.3	0.87	40.8 ± 2.9	0.28	41.4 ± 3.5	0.92
r-eta	40.9 ± 3.2	0.53	40.9 ± 2.8	0.32	40.8 ± 3.1	0.49
AB$_R$	41.4 ± 3.3	0.97	41.1 ± 2.6	0.65	41.2 ± 3.0	0.88
Flare-Solar: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	$\bullet 5.73 \pm 0.11$	–	5.85 ± 0.14	–	$\bullet 5.82 \pm 0.12$	–
LB	5.75 ± 0.11	0.99	5.87 ± 0.13	0.99	5.86 ± 0.11	0.99
MB	5.76 ± 0.11	0.99	5.90 ± 0.11	0.99	5.88 ± 0.11	0.99
Eta	5.80 ± 0.14	0.99	$\bullet 5.85 \pm 0.13$	0.50	5.83 ± 0.12	0.92
r-eta	5.76 ± 0.10	0.99	5.88 ± 0.10	0.99	5.88 ± 0.10	0.99
AB$_R$	5.75 ± 0.16	0.98	5.86 ± 0.18	0.81	5.83 ± 0.15	0.62
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.66 ± 0.15	–	6.67 ± 0.12	–	6.70 ± 0.13	–
LB	6.66 ± 0.14	0.11	6.68 ± 0.12	0.74	6.67 ± 0.12	0.01
MB	6.66 ± 0.13	0.18	$\bullet 6.67 \pm 0.12$	0.31	$\bullet 6.67 \pm 0.11$	0.00
Eta	6.66 ± 0.13	0.14	6.70 ± 0.13	0.99	6.72 ± 0.15	0.99
r-eta	$\bullet 6.65 \pm 0.14$	0.03	6.67 ± 0.10	0.35	6.68 ± 0.11	0.02
AB$_R$	6.69 ± 0.16	0.98	6.73 ± 0.17	0.99	6.70 ± 0.17	0.73

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 13: Error and Log-Loss×10 on the Test Set of Splice Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Splice: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	8.6 ± 1.0	–	11.8 ± 2.1	–	11.9 ± 1.9	–
LB	8.4 ± 1.4	0.28	10.9 ± 1.7	0.03	11.6 ± 1.7	0.26
MB	8.7 ± 1.0	0.54	11.6 ± 1.8	0.21	11.5 ± 1.4	0.14
Eta	10.3 ± 2.0	0.99	14.8 ± 2.6	0.99	14.0 ± 2.0	0.99
r-eta	$\bullet 8.0 \pm 0.9$	0.04	$\bullet 10.8 \pm 1.5$	0.01	11.8 ± 2.0	0.38
AB$_R$	8.2 ± 1.0	0.07	11.4 ± 2.0	0.25	$\bullet 11.2 \pm 1.5$	0.05
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	31.0 ± 1.9	–	33.0 ± 2.7	–	33.1 ± 2.0	–
LB	31.4 ± 1.7	0.88	33.0 ± 1.9	0.51	33.5 ± 1.8	0.75
MB	31.1 ± 1.8	0.58	33.5 ± 2.5	0.82	32.8 ± 1.8	0.29
Eta	31.5 ± 2.1	0.87	32.9 ± 2.0	0.45	32.7 ± 1.4	0.22
r-eta	30.6 ± 2.0	0.21	$\bullet 32.9 \pm 2.2$	0.43	32.7 ± 1.6	0.20
AB$_R$	$\bullet 30.5 \pm 1.6$	0.14	32.9 ± 2.1	0.44	$\bullet 32.0 \pm 1.6$	0.03
Splice: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	$\bullet 1.94 \pm 0.16$	–	2.96 ± 0.57	–	2.91 ± 0.40	–
LB	1.98 ± 0.16	0.85	2.63 ± 0.31	0.00	2.80 ± 0.36	0.19
MB	2.09 ± 0.11	0.99	2.83 ± 0.43	0.14	2.75 ± 0.30	0.02
Eta	2.47 ± 0.44	0.99	3.49 ± 0.50	0.99	3.62 ± 0.56	0.99
r-eta	2.08 ± 0.13	0.99	$\bullet 2.61 \pm 0.28$	0.00	$\bullet 2.70 \pm 0.31$	0.01
AB$_R$	1.95 ± 0.18	0.53	2.92 ± 0.43	0.37	2.96 ± 0.29	0.71
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.34 ± 0.17	–	6.53 ± 0.26	–	6.55 ± 0.15	–
LB	6.32 ± 0.26	0.40	6.56 ± 0.23	0.66	6.54 ± 0.25	0.36
MB	6.24 ± 0.22	0.02	6.49 ± 0.26	0.31	6.46 ± 0.16	0.01
Eta	6.34 ± 0.27	0.53	6.65 ± 0.97	0.70	6.63 ± 0.29	0.84
r-eta	$\bullet 6.16 \pm 0.21$	0.00	$\bullet 6.43 \pm 0.24$	0.10	$\bullet 6.46 \pm 0.19$	0.04
AB$_R$	6.32 ± 0.24	0.40	6.57 ± 0.19	0.76	6.57 ± 0.27	0.65

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 14: Error and Log-Loss×10 on the Test Set of Thyroid Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Thyroid: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	7.7 ± 3.3	–	8.8 ± 3.9	–	8.5 ± 3.9	–
LB	7.7 ± 3.2	0.50	8.9 ± 4.5	0.56	8.1 ± 3.6	0.13
MB	7.2 ± 3.5	0.09	8.6 ± 3.8	0.29	8.7 ± 3.8	0.66
Eta	9.4 ± 4.5	0.99	10.0 ± 5.0	0.99	10.0 ± 5.8	0.99
r-eta	$\bullet 7.2 \pm 2.9$	0.07	$\bullet 8.1 \pm 3.3$	0.02	$\bullet 7.8 \pm 3.2$	0.02
AB$_R$	7.9 ± 3.3	0.74	8.5 ± 4.0	0.28	8.4 ± 3.4	0.36
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	28.4 ± 6.0	–	29.1 ± 6.2	–	29.9 ± 6.5	–
LB	$\bullet 28.1 \pm 5.2$	0.29	28.5 ± 6.3	0.14	30.1 ± 6.1	0.64
MB	28.5 ± 5.2	0.56	$\bullet 28.3 \pm 6.5$	0.08	29.7 ± 6.2	0.30
Eta	28.6 ± 5.3	0.60	29.5 ± 6.6	0.76	30.7 ± 6.2	0.93
r-eta	28.6 ± 5.3	0.66	28.7 ± 6.6	0.27	29.8 ± 6.5	0.39
AB$_R$	28.5 ± 5.5	0.57	29.0 ± 5.7	0.42	$\bullet 29.4 \pm 6.2$	0.16
Thyroid: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	2.29 ± 1.55	–	2.58 ± 1.26	–	2.53 ± 1.15	–
LB	1.98 ± 1.14	0.01	2.33 ± 0.91	0.01	2.08 ± 0.83	0.00
MB	1.87 ± 0.93	0.00	2.42 ± 1.07	0.09	2.15 ± 0.88	0.00
Eta	8.46 ± 16.0	0.99	8.00 ± 12.7	0.99	5.78 ± 9.22	0.99
r-eta	$\bullet 1.64 \pm 0.96$	0.00	$\bullet 2.10 \pm 0.94$	0.00	$\bullet 1.95 \pm 0.82$	0.00
AB$_R$	3.25 ± 4.49	0.99	2.83 ± 1.47	0.97	2.91 ± 1.77	0.99
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.18 ± 0.69	–	6.16 ± 0.73	–	6.47 ± 0.99	–
LB	6.07 ± 0.71	0.04	$\bullet 6.12 \pm 0.71$	0.23	6.24 ± 0.87	0.00
MB	$\bullet 5.98 \pm 0.68$	0.00	6.12 ± 0.75	0.28	$\bullet 6.22 \pm 0.82$	0.00
Eta	7.03 ± 3.54	0.99	6.46 ± 2.10	0.94	7.19 ± 3.58	0.98
r-eta	6.11 ± 0.77	0.17	6.17 ± 0.79	0.56	6.35 ± 1.03	0.05
AB$_R$	6.30 ± 1.07	0.93	6.34 ± 0.98	0.99	6.42 ± 0.93	0.28

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 15: Error and Log-Loss×10 on the Test Set of Titanic Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Titanic: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	22.8 ± 1.5	–	$\bullet 22.5 \pm 0.9$	–	23.1 ± 1.9	–
LB	22.8 ± 1.3	0.33	22.7 ± 1.5	0.86	$\bullet 22.7 \pm 1.2$	0.03
MB	$\bullet 22.7 \pm 1.1$	0.22	22.7 ± 1.2	0.94	22.9 ± 1.4	0.11
Eta	23.0 ± 1.6	0.81	22.9 ± 1.6	0.99	23.3 ± 2.2	0.84
r-eta	22.8 ± 1.5	0.41	22.6 ± 1.1	0.85	22.9 ± 1.5	0.06
AB$_R$	23.0 ± 1.8	0.78	22.9 ± 1.6	0.99	23.1 ± 2.1	0.54
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	34.6 ± 2.6	–	35.1 ± 2.2	–	34.8 ± 2.3	–
LB	34.6 ± 2.0	0.43	35.0 ± 2.2	0.26	34.4 ± 2.0	0.01
MB	34.7 ± 2.6	0.65	35.2 ± 2.1	0.61	34.2 ± 1.6	0.00
Eta	34.5 ± 2.0	0.32	35.2 ± 2.2	0.57	34.4 ± 1.9	0.02
r-eta	$\bullet 34.5 \pm 2.3$	0.19	$\bullet 35.0 \pm 2.0$	0.18	$\bullet 34.1 \pm 1.9$	0.00
AB$_R$	34.7 ± 3.8	0.53	35.4 ± 3.5	0.77	34.4 ± 2.0	0.02
Titanic: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	5.32 ± 0.20	–	5.34 ± 0.24	–	5.37 ± 0.22	–
LB	5.32 ± 0.20	0.55	$\bullet 5.30 \pm 0.18$	0.02	5.35 ± 0.20	0.09
MB	5.32 ± 0.16	0.39	5.33 ± 0.17	0.34	5.35 ± 0.22	0.09
Eta	5.34 ± 0.24	0.80	5.37 ± 0.25	0.89	5.39 ± 0.33	0.69
r-eta	$\bullet 5.30 \pm 0.20$	0.09	5.31 ± 0.16	0.06	$\bullet 5.33 \pm 0.20$	0.02
AB$_R$	5.43 ± 0.29	0.99	5.46 ± 0.47	0.99	5.45 ± 0.30	0.99
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.49 ± 0.16	–	6.52 ± 0.15	–	6.52 ± 0.17	–
LB	6.49 ± 0.17	0.61	6.54 ± 0.18	0.92	$\bullet 6.49 \pm 0.14$	0.03
MB	6.48 ± 0.16	0.33	6.54 ± 0.19	0.88	6.50 ± 0.15	0.08
Eta	6.46 ± 0.14	0.02	6.53 ± 0.21	0.74	6.60 ± 0.97	0.81
r-eta	$\bullet 6.46 \pm 0.13$	0.02	$\bullet 6.52 \pm 0.15$	0.46	6.49 ± 0.16	0.03
AB$_R$	6.57 ± 0.26	0.99	6.64 ± 0.38	0.99	6.59 ± 0.29	0.99

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Table 16: Error and Log-Loss×10 on the Test Set of Twonorm Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Twonorm: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	5.9 ± 1.2	–	$\bullet 6.5 \pm 1.0$	–	6.8 ± 1.0	–
LB	5.9 ± 1.2	0.45	6.8 ± 1.0	0.99	7.1 ± 1.0	0.99
MB	5.9 ± 1.1	0.40	6.8 ± 1.0	0.99	7.1 ± 1.0	0.99
Eta	9.7 ± 1.4	0.99	9.5 ± 1.3	0.99	9.6 ± 1.2	0.99
r-eta	5.7 ± 0.8	0.09	6.9 ± 1.2	0.99	7.1 ± 1.2	0.99
AB$_R$	$\bullet 5.7 \pm 0.9$	0.09	6.5 ± 1.1	0.60	$\bullet 6.4 \pm 1.0$	0.01
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	30.6 ± 1.4	–	30.4 ± 1.5	–	30.5 ± 1.6	–
LB	31.4 ± 1.6	0.99	31.4 ± 1.4	0.99	30.9 ± 1.6	0.99
MB	31.4 ± 1.8	0.99	31.5 ± 1.9	0.99	31.1 ± 1.5	0.99
Eta	30.4 ± 1.7	0.19	30.3 ± 1.6	0.43	30.4 ± 1.5	0.26
r-eta	31.3 ± 1.5	0.99	31.0 ± 1.5	0.99	31.1 ± 1.5	0.99
AB$_R$	$\bullet 29.6 \pm 1.6$	0.00	$\bullet 29.3 \pm 2.0$	0.00	$\bullet 29.2 \pm 1.7$	0.00
Twonorm: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	2.13 ± 0.35	–	2.06 ± 0.29	–	2.10 ± 0.41	–
LB	1.73 ± 0.20	0.00	1.83 ± 0.26	0.00	1.84 ± 0.25	0.00
MB	1.66 ± 0.21	0.00	$\bullet 1.74 \pm 0.20$	0.00	1.79 ± 0.24	0.00
Eta	4.08 ± 1.47	0.99	3.40 ± 0.91	0.99	3.27 ± 0.87	0.99
r-eta	$\bullet 1.64 \pm 0.22$	0.00	1.78 ± 0.27	0.00	$\bullet 1.75 \pm 0.24$	0.00
AB$_R$	2.27 ± 0.59	0.98	2.15 ± 0.31	0.99	2.03 ± 0.33	0.06
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.69 ± 0.21	–	6.66 ± 0.22	–	6.61 ± 0.23	–
LB	6.64 ± 0.22	0.03	6.62 ± 0.22	0.07	6.59 ± 0.26	0.18
MB	6.64 ± 0.23	0.03	6.63 ± 0.23	0.14	6.57 ± 0.24	0.06
Eta	6.66 ± 0.27	0.19	6.69 ± 0.35	0.80	6.62 ± 0.30	0.56
r-eta	$\bullet 6.61 \pm 0.24$	0.00	$\bullet 6.59 \pm 0.27$	0.01	$\bullet 6.48 \pm 0.26$	0.00
AB$_R$	6.75 ± 0.32	0.96	6.70 ± 0.26	0.93	6.71 ± 0.24	0.99

Notes: The *p*-values are calculated by one-side paired *t*-test against Adaboost. The dot indicates the best method under each noise condition.

Table 17: Error and Log-Loss×10 on the Test Set of Waveform Among Six Boosting Algorithms: Adaboost (AB), Logitboost (LB), Madaboost (MB), Eta-Boost (Eta), Robust Eta-Boost (r-eta), and Adaboost$_{Reg}$ (AB$_R$).

Waveform: Test Error						
Mislabel	0%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	13.7 ± 1.2	–	14.3 ± 1.3	–	14.2 ± 1.3	–
LB	14.1 ± 1.3	0.99	14.6 ± 1.4	0.97	14.7 ± 1.3	0.99
MB	14.2 ± 1.1	0.99	14.9 ± 1.4	0.99	15.1 ± 1.3	0.99
Eta	14.7 ± 1.3	0.99	15.0 ± 1.4	0.99	15.0 ± 1.3	0.99
r-eta	14.3 ± 1.3	0.99	14.7 ± 1.3	0.99	14.8 ± 1.3	0.99
AB$_R$	$\bullet 13.2 \pm 1.0$	0.00	$\bullet 13.8 \pm 1.3$	0.00	$\bullet 13.8 \pm 1.1$	0.01
Mislabel	20%					
Outlier	0%		3%		5%	
	Error	p-val	Error	p-val	Error	p-val
AB	32.9 ± 1.5	–	32.8 ± 1.4	–	33.0 ± 1.2	–
LB	33.1 ± 1.5	0.86	32.9 ± 1.2	0.82	32.9 ± 1.1	0.17
MB	33.2 ± 1.5	0.96	33.0 ± 1.3	0.97	33.1 ± 1.2	0.85
Eta	32.7 ± 1.4	0.15	32.6 ± 1.3	0.05	32.8 ± 1.3	0.11
r-eta	33.1 ± 1.5	0.81	32.6 ± 1.5	0.15	33.0 ± 1.3	0.64
AB$_R$	$\bullet 32.3 \pm 1.7$	0.00	$\bullet 32.4 \pm 1.5$	0.01	$\bullet 32.4 \pm 1.3$	0.00
Waveform: Log-Loss×10						
Mislabel	0%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	3.68 ± 0.36	–	3.76 ± 0.42	–	3.69 ± 0.34	–
LB	3.60 ± 0.34	0.02	3.61 ± 0.38	0.00	3.62 ± 0.33	0.02
MB	3.52 ± 0.32	0.00	3.60 ± 0.35	0.00	3.54 ± 0.33	0.00
Eta	4.19 ± 0.69	0.99	4.21 ± 0.94	0.99	4.13 ± 0.94	0.99
r-eta	$\bullet 3.44 \pm 0.35$	0.00	$\bullet 3.47 \pm 0.29$	0.00	$\bullet 3.47 \pm 0.31$	0.00
AB$_R$	3.81 ± 0.54	0.99	3.82 ± 0.43	0.88	3.75 ± 0.37	0.92
Mislabel	20%					
Outlier	0%		3%		5%	
	Log-Loss	p-val	Log-Loss	p-val	Log-Loss	p-val
AB	6.48 ± 0.21	–	6.45 ± 0.22	–	6.50 ± 0.24	–
LB	6.49 ± 0.23	0.64	6.47 ± 0.21	0.82	6.47 ± 0.19	0.14
MB	6.49 ± 0.23	0.71	$\bullet 6.43 \pm 0.18$	0.18	$\bullet 6.46 \pm 0.17$	0.08
Eta	6.50 ± 0.34	0.69	6.48 ± 0.32	0.84	6.54 ± 0.44	0.84
r-eta	$\bullet 6.45 \pm 0.20$	0.12	6.44 ± 0.24	0.34	6.48 ± 0.22	0.27
AB$_R$	6.53 ± 0.26	0.97	6.53 ± 0.26	0.99	6.50 ± 0.25	0.61

Notes: The p-values are calculated by one-side paired t-test against Adaboost. The dot indicates the best method under each noise condition.

Acknowledgments

We thank the anonymous referees for helpful comments. This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B), 17700277, 2005.

References

Amari, S., & Nagaoka, H. (2000). *Methods of information geometry*. New York: Oxford University Press.

Bartlett, P. L., Jordan, M. I., & McAuliffe, J. D. (2003). *Convexity, classification, and risk bounds* (Tech. rep. 638). Berkeley: Statistics Department, University of California, Berkeley.

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Belmont, MA: Athena Scientific.

Blanchard, G., Schäfer, C., Rozenholc, Y., & Müller, K.-R. (2005). *Optimal dyadic decision trees* (Tech. rep.) Berlin: Fraunhofer FIRST, 2005. Available online at http://ida.first.fraunhofer.de/blanchard/publi/index.html.

Breiman, L. (1994). *Bagging predictors* (Tech. rep. 421). Berkeley: Statistics Department, University of California, Berkeley.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth and Brooks/Cole.

Copas, J. (1988). Binary regression models for contaminated data. *J. Royal Statist. Soc. B., 50*, 225–265.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Demiriz, A., Bennett, K. P., & Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Machine Learning, 46*(1–3), 225–254.

Domingo, C., & Watanabe, O. (2000). MadaBoost: A modification of AdaBoost. In *Proc. of the 13th Conference on Computational Learning Theory, COLT'00*. San Francisco: Morgan Kaufmann.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics, 28*, 337–407.

Grünwald, P. D., & Dawid, A. P. (2004). Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *Annals of Statistics, 32*(4), 1367–1433.

Halmos, P. R. (1974). *Measure theory*. New York: Springer-Verlag.

Hampel, F. R., Rousseeuw, P. J., Ronchetti, E. M., & Stahel, W. A. (1986). *Robust statistics: The approach based on influence functions*. New York: Wiley.

Kalai, A., & Servedio, R. A. (2003). Boosting in the presence of noise. In *STOC '03: Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*. New York: ACM.

Kanamori, T., Takenouchi, T., Eguchi, S., & Murata, N. (2004). The most robust loss function for boosting. In *Neural Information Processing: 11th International Conference, ICONIP* (pp. 496–501). Berlin: Springer.

Lebanon, G., & Lafferty, J. (2002). Boosting and maximum likelihood for exponential models. In T. G. Dieterrich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14*. Cambridge, MA: MIT Press.

MacCullagh, P. A., & Nelder, J. (1989). *Generalized linear models*. London: Chapman and Hall.

Mason, L., Baxter, J., Bartlett, P. L., & Frean, M. (1999). Boosting algorithms as gradient descent. In M. S. Stearns, S. Solla, & D. Cohen (Eds.), *Advances in neural information processing systems, 11*. Cambridge, MA: MIT Press.

McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition*. New York: Wiley.

Meir, R., & Rätsch, G. (2003). *An introduction to boosting and leveraging*. New York: Springer.

Murata, N., Takenouchi, T., Kanamori, T., & Eguchi, S. (2004). Information geometry of *u*-boost and Bregman divergence. *Neural Computation, 16*(7), 1437–1481.

Rätsch, G. (2001). *Robust boosting via convex optimization*. Unpublished doctoral dissertation, University of Potsdam.

Rätsch, G., Demiriz, A., & Bennett, K. (2002). Sparse regression ensembles in infinite and finite hypothesis spaces. *Machine Learning, 48*, 193–221.

Rätsch, G., Onoda, T., & Müller, K.-R. (2001). Soft margins for Adaboost. *Machine Learning, 42*(3), 287–320.

Rätsch, G., Schölkopf, B., Smola, A. J., Mika, S., Onoda, T., & Müller, K.-R. (2000). *Robust ensemble learning*. Cambridge, MA: MIT Press.

Rosset, S. (2005). Robust boosting and its relation to bagging. In *Proceedings of the 11th ACM SICKDD International Conference on Knowledge Discovery in Data Mining* (pp. 249–255). New York: ACM Press.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics, 26*(5), 1651–1686.

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT Press.

Servedio, R. (2003). Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research, 4*, 633–648.

Takenouchi, T., & Eguchi, S. (2004). Robustifying Adaboost by adding the naive error rate. *Neural Computation, 16*(4), 767–787.

van der Vaart, A. W. (1998). *Asymptotic statistics*. Cambridge: Cambridge University Press.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Victoria-Feser, M.-P. (2002). Robust inference with binary data. *Psychometrika, 67*(1), 21–32.

Received January 29, 2006; accepted August 2, 2006.