Electronic Journal of Statistics
Vol. 12 (2018) 599–650
ISSN: 1935-7524
https://doi.org/10.1214/18-EJS1404

Robust boosting with truncated loss functions

Zhu Wang

Department of Research
Connecticut Children's Medical Center
Hartford, Connecticut 06106 USA
e-mail: zhuwang@gmail.com

Abstract: Boosting is a powerful machine learning tool with attractive theoretical properties. In recent years, boosting algorithms have been extended to many statistical estimation problems. For data contaminated with outliers, however, development of boosting algorithms is very limited. In this paper, innovative robust boosting algorithms utilizing the majorization-minimization (MM) principle are developed for binary and multi-category classification problems. Based on truncated loss functions, the robust boosting algorithms share a unified framework for linear and nonlinear effects models. The proposed methods can reduce the heavy influence from a small number of outliers which could otherwise distort the results. In addition, adaptive boosting for the truncated loss functions are developed to construct more sparse predictive models. We present convergence guarantees for smooth surrogate loss functions with both iteration-varying and constant step-sizes. We conducted empirical studies using data from simulations, a pediatric database developed for the US Healthcare Cost and Utilization Project, and breast cancer gene expression data. Compared with non-robust boosting, robust boosting improves classification accuracy and variable selection.

MSC 2010 subject classifications: Primary 62H30, 62G35; secondary 68Q32, 90C26.
Keywords and phrases: Robust method, machine learning, boosting, difference of convex, MM algorithm.

Received August 2016.

Contents

1. Introduction

Boosting algorithms are one of the most influential methodological approaches for data analysis developed in the last two decades. Initially boosting was a powerful machine learning algorithm to predict binary outcomes [9]. The basic idea is to iteratively construct simple or weak classifiers and to combine their solutions to obtain a more accurate prediction. For instance, decision trees have been widely used in many classification problems, partly due to their interpretability. However, tree based classification has a mixed track-record of predictive performance. In boosting, single decision-tree models are combined to an arbitrary depth and shape, adjusted to optimize the resulting model's classification performance. The ensemble decision-trees perform substantially better than single-tree models. Boosting can be interpreted as a method for fitting regression models in a stagewise fashion when optimizing well defined loss functions [12]. This gradient descent view of boosting has led to considerable development in different settings for both linear and non-linear effects models [3, 19].

Variable selection is an important issue in data analysis. Among a large number of candidate predictors, predictive models are expected to select a subset of risk factors for improved accuracy and parsimonious interpretation. In recent years, statistical methods to conduct variable selection have been actively developed. Among them, boosting is one of the most attractive methods by simply choosing appropriate base learners [12, 3, 19].

Boosting algorithms for classification include AdaBoost, LogitBoost and HingeBoost [12, 3, 31]. These algorithms were developed to minimize exponential, logistic and hinge loss, respectively. Note that the three loss functions are convex. The convexity property can make optimization in some sense "easier" than the general case - for example, any local minimum must be a global minimum. A convex loss function, however, suffers from the negative impact of outliers. The above boosting algorithms therefore tend to be sensitive to noisy training data. When there exist points far away from their own classes, the classifiers are strongly influenced by such points because of the underlying loss functions used in these boosting algorithms.

To this end, robust boosting algorithms based on nonconvex loss functions have been proposed. For instance, BrownBoost and RobustBoost have shown resistant to outliers [6, 7]. A nonconvex loss can be conveniently obtained by truncating a popular loss function, leading to more robust prediction accuracy [35, 24, 14]. However, there is a lack of gradient decent boosting algorithms for truncated loss functions. In this paper, we aim to fill the gap. Because the truncated loss functions are nonconvex, a key computational trick is the difference of convex (DC) algorithm [29, 35, 24, 14]. The idea is to work with a series of simpler but well defined surrogate functions instead of the original loss function. As a result, the DC algorithm substitutes a series of simple optimization problems for a difficult optimization problem. Most notably, the DC algorithm has been linked with the majorize-minimize (MM) algorithm [15]. Another popular MM algorithm, the so-called EM (expectation-maximization) algorithm has been extensively studied for missing data among many other applications [5, 21]. The major contribution of this paper is to combine the gradient descent boosting and the DC algorithm for a suite of truncated loss functions. These DC-boosting algorithms are more robust to data contaminated with outliers than their counterparts based on standard loss functions. We conduct convergence analysis of the proposed algorithms as well as the standard functional gradient boosting algorithms. We provide implementations of these algorithms through the publicly available `R` package, `bst` (available at `http://cran.r-project.org`).

Methods for robust boosting have applications in many scientific fields, including healthcare research and gene expression analysis. In this article we apply the robust boosting methods to classify healthcare costs using the KID healthcare database (available at `www.hcup-us.ahrq.gov`). Healthcare cost is a tremendous burden of public expenditure, and high-cost patients are typically related to severe disease diagnosis. Early identification of high-cost patients can help design targeted interventions that can defer or even avoid adverse outcomes [22]. Administrative healthcare databases have different sources of variability including outliers. Research examining inpatient expenditures can be distorted by a small number of extremely high or low charges that have undue influence on the results. This is not desirable because these charges could reflect measurement errors that distort the goodness of fit of statistical models. Suspicious charges are potential data entry errors and represent discharges that are outliers in terms of the average charge per day of stay [10]. Another challenge is that administrative data have voluminous amount of patient records including patient demographics, payment information, disease severity, comorbidities, diagnostic and procedure information and hospital characteristics. For instance, there are hundreds of diagnosis categories as potential predictors. Therefore it is crucial to identify a small number of risk factors for healthcare costs stratification. Likewise, gene expression data may be mislabelled and also contain many predictors.

In Section 2 we describe robust truncated loss functions for binary classification problems. We show how to implement the DC technique to a truncated loss. In Section 3 we present DC boosting algorithm (DCBA) to minimize truncated loss functions. To further reduce risk factors, adaptive robust boosting is presented. We generalize robust boosting to multi-class problems in Section 4. Theoretical analyses of the boosting algorithms are presented in Section 5. The performance of the proposed algorithms is investigated in Section 6 via simulated data. In Section 7, the proposed algorithms are applied to classify healthcare costs and clinical status from breast cancer patients. Proofs of the theoretical results are in the Appendix.

2. Robust loss functions

Assume that we have observations (x_i, y_i), where x_i is a p-dimensional predictor variable for $i = 1, 2, ..., n$. For a binary outcome y taking values $+1$ and -1, with prediction f and margin $u = yf$, consider a margin based loss function $\ell(u)$. Table 1 lists three widely used loss functions: logistic, exponential and hinge loss [12, 3, 30]. In particular, support vector machines utilize the hinge loss [30]. These loss functions are sensitive to outliers because the loss values are unbounded and can go to infinity with outliers, see Figure 1. Therefore classification rules based on these loss functions can suffer from outliers. A simple remedy is to truncate the unbounded loss functions [1, 36, 24]. A truncated loss at a constant location s is

$$L(u, s) = \min(s, \ell(u)). \tag{1}$$

Because truncation reduces the impact of misclassified outliers, the resulting classifiers are more robust and accurate than the standard classifiers. Table 1 and Figure 1 compare standard loss functions $\ell(u)$ and truncated counterparts [35, 24, 14, 1, 36]. The truncated logistic loss shows that $L(u, s)$ increases as u decreases, but once u is less than s, the truncated loss becomes a constant. This implies that the truncated loss becomes larger up to an upperbound as an observation deviates further away from the classification boundary. For outliers located further away from the boundary satisfying $u \leq s$, the truncated loss maintains a constant value $\ell(s)$ so that the outliers cannot further influence the classification boundary. In contrast, as can be seen in Figure 1, the standard logistic loss has no boundary and the impact of outliers grows to infinity. The interpretations of truncated exponential and hinge loss functions are similar. One exception is the difference logistic loss. As u decreases, the difference logistic loss increases at a smaller rate compared to logistic loss, and converges to a constant limit value.

Theoretical properties of truncated logistic, exponential and hinge loss have been established [35, 24]. In particular, Fisher consistency provides justifications for these truncated loss functions when used in classification. In the statistical literature, Fisher consistency originally means that the estimation procedure using the entire population will produce the true value of the estimation. For instance, the maximum likelihood estimation is typically Fisher consistent. In decision theory, a loss function is Fisher consistent if the population minimizer of

TABLE 1
Loss functions. $z_+ = \max(0, z)$, $s > 0$ *for difference logistic, and* $s \le 0$ *otherwise.*

Truncated loss	$L(u, s)$	Standard loss	$\ell(u)$
Truncated logistic	$\min\big(\log(1+\exp(-u)),$ $\log(1+\exp(-s))\big)$	logistic	$\log(1+\exp(-u))$
Difference logistic	$\log(1+\exp(-u))-$ $\log(1+\exp(-u-s))$	logistic	$\log(1+\exp(-u))$
Truncated exponential	$\min\big(\exp(-u), \exp(-s)\big)$	exponential	$\exp(-u)$
Truncated hinge	$(1-u)_+ - (s-u)_+$	hinge	$(1-u)_+$

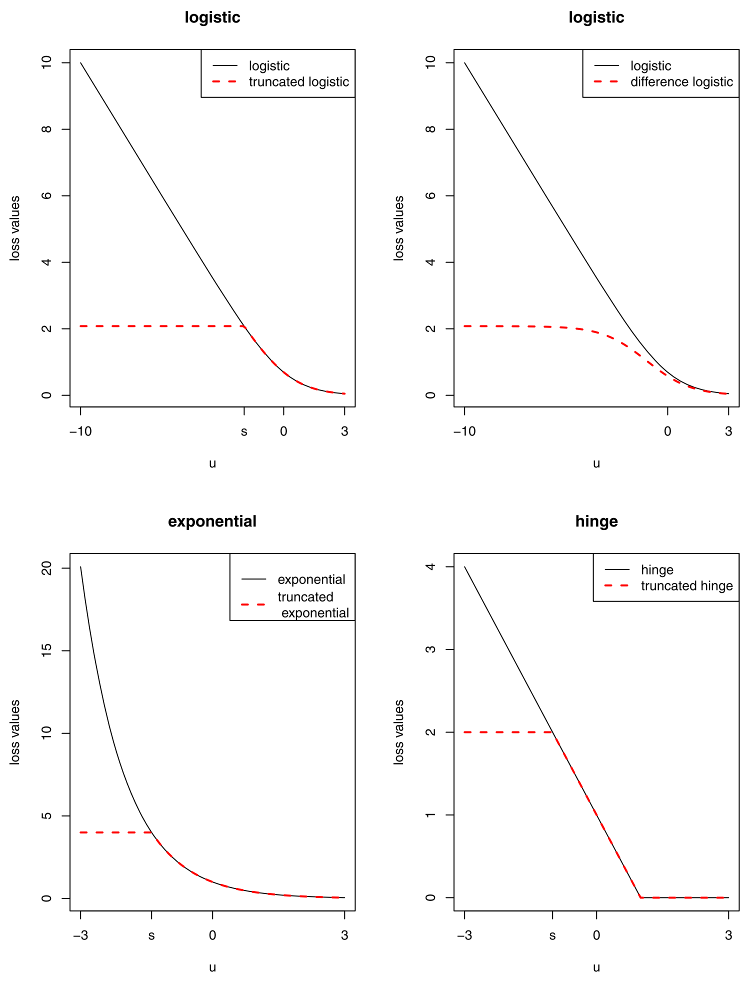


FIG 1. *Loss functions*

the risk leads to the Bayes optimal decision rule given by $f(x) = \text{sign}(p(x) - \frac{1}{2})$, where $p(x) = P(Y = +1|x)$ for any x in the predictor space [17]. Minimizing an empirical risk $\frac{1}{n}\sum_{i=1}^{n} L(y_i f(x_i), s)$, a classification procedure effectively approximates the optimal decision rule if the risk is Fisher consistent. For the difference logistic loss [14], we have the following properties.

Proposition 1. *The difference logistic loss $L_{DL}(u, s)$ and the truncated logistic loss $L_{TL}(u, s)$ are asymptotically equivalent. Specifically,*

$$\lim_{u \to -\infty \text{ or } u \to \infty} \{L_{DL}(u, \log(1 + \exp(-s))) - L_{TL}(u, s)\} = 0.$$

Proposition 2. *The minimizer f^* of $E(L_{DL}(Yf(X), s))$ has the same sign as $p(x) - 1/2$.*

Proposition 1 recognizes the relationship between the difference logistic and truncated logistic loss, and Proposition 2 establishes Fisher consistency of the former loss.

In general, we may write a nonconvex loss function $L(u, s)$ as a difference of convex loss between $\ell(u)$ and $-\ell_s(u)$:

$$L(u, s) = \ell(u) + \ell_s(u). \tag{2}$$

For instance, Equation (1) may be converted to (2) where

$$\ell_s(u) = -\max(0, \ell(u) - s).$$

Many robust truncated loss functions in the literature can be written in this format, as demonstrated in Table 2 [14, 35, 24]. Note, the nonconvex loss functions may be nonsmooth, such as the truncated hinge loss.

TABLE 2
Truncated loss function $L(u, s) = \ell(u) + \ell_s(u)$

Truncated loss	$\ell(u)$	$\ell_s(u)$
Truncated logistic	$\log(1 + \exp(-u))$	$-\big(\log(1 + \exp(-u)) - \log(1 + \exp(-s))\big)_+$
Difference logistic	$\log(1 + \exp(-u))$	$-\log(1 + \exp(-u - s))$
Truncated exponential	$\exp(-u)$	$-\max(0, \exp(-u) - \exp(-s))$
Truncated hinge	$(1 - u)_+$	$-(s - u)_+$

An important issue is how to minimize the nonconvex truncated loss function $L(u, s)$. Since $\ell_s(\cdot)$ in (2) is a concave function, we can utilize the DC or majorize-minimize (MM) scheme. In the sequel we use notation $L(f, s)$ instead of $L(u, s)$ after replacing $u = yf$. Even if suppressed in the notation, it should be understood that the loss function $L(f, s)$ depends on the classification outcome y. The DC algorithm is an iterative process. Given the current estimate $f^{(k-1)}$ in the $(k-1)$-th iteration, we apply a linear majorization to the concave function $\ell_s(f)$ in (2):

$$\begin{aligned} \ell_s(f) &\le \ell_s(f^{(k-1)}) + \frac{\partial \ell_s(f)}{\partial f}|_{f=f^{(k-1)}}(f - f^{(k-1)}) \\ &\triangleq h_k(f). \end{aligned} \tag{3}$$

Let $H_k(f)$ denote the surrogate loss given by

$$H_k(f) = \ell(f) + h_k(f). \tag{4}$$

We have the following relationship regarding the objective loss function $L(f, s)$:

$$L(f, s) \leq H_k(f) \quad \text{for all } f, \quad L(f^{(k-1)}, s) = H_k(f^{(k-1)}). \tag{5}$$

Since $\ell(f)$ is a convex function and $h_k(f)$ is a linear function, $H_k(f) = \ell(f) + h_k(f)$ is convex. Therefore $H_k(f)$ can be readily minimized or reduced to obtain a new estimate $f^{(k)}$. Together with (5), the following statement holds:

$$L(f^{(k)}, s) \leq H_k(f^{(k)}) \leq H_k(f^{(k-1)}) = L(f^{(k-1)}, s). \tag{6}$$

The descent property (6) makes the DC algorithm numerically stable. Since the algorithm only requires decreasing $H_k(f)$, the gradient descent boosting algorithm is a simple yet powerful tool to accomplish this task. Let $L_{DCF}(f, s)$ denote all terms related to f in $H_k(f)$:

$$L_{DCF}(f, s) = \ell(f) + \frac{\partial \ell_s(f)}{\partial f}|_{f=f^{(k-1)}} f.$$

We then obtain

$$H_k(f) = L_{DCF}(f, s) + \ell_s(f^{(k)}) - \frac{\partial \ell_s(f)}{\partial f}|_{f=f^{(k-1)}} f^{(k-1)}.$$

After eliminating constant terms in $H_k(f)$, minimizing $H_k(f)$ is equivalent to minimizing its simplified version $L_{DCF}(f, s)$. For truncated loss functions in Table 2, the corresponding surrogate loss functions $L_{DCF}(f, s)$ are presented in Table 3. We have two remarks on the subscript DCF. First, the two letters DC highlight the linear majorization trick (3) that is often referred to the DC algorithm. Second, the letter F emphasizes that the DC algorithm in this paper applies to function estimation problems which are different from parameter estimation problems typically found in the literature [1, 14, 35, 36, 24].

TABLE 3
Truncated nonconvex loss and surrogate convex loss at the $(k+1)$-th iteration. $I(\cdot)$ is the indicator function.

Truncated loss	Surrogate loss $L_{DCF}(f, s)$
Truncated logistic	$\log(1 + \exp(-yf)) + yf \frac{\exp(-yf^{(k)})}{1+\exp(-yf^{(k)})} \mathrm{I}\left(yf^{(k)} < s\right)$
Difference logistic	$\log(1 + \exp(-yf)) + yf \frac{\exp(-yf^{(k)}-s)}{1+\exp(-yf^{(k)}-s)}$
Truncated exponential	$\exp(-yf) + yf \exp(-yf^{(k)}) \mathrm{I}\left(yf^{(k)} < s\right)$
Truncated hinge	$(1 - yf)_+ + yf \mathrm{I}(s - yf^{(k)} > 0)$

3. Boosting nonconvex loss functions

The truncated loss (2) is minimized through the empirical loss:

$$\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i)) + \ell_s(y_i, f(x_i)). \tag{7}$$

We propose difference of convex boosting algorithm (DCBA) in Section 3.1 and adaptive DCBA in Section 3.2. To solve the nonconvex minimization problem (7), these algorithms minimize a sequence of surrogate convex problems given the current estimate $f(x_i)^{(k-1)}$:

$$\frac{1}{n}\sum_{i=1}^{n} L_{DCF}(y_i, f(x_i), s), \tag{8}$$

where

$$L_{DCF}(y_i, f(x_i), s) = \ell(y_i, f(x_i)) + \frac{\partial \ell_s(y_i, f)}{\partial f}|_{f=f(x_i)^{(k-1)}} f(x_i).$$

3.1. Difference of convex boosting algorithm

The DCBA (Algorithm 1) has a sequence of nested loops. The outer loop is to update the surrogate convex function (8) using the current estimate $f^{(k-1)}$. At the inner loop, the gradient decent boosting algorithm minimizes the convex loss (8). When implemented in a computer program, we choose a constant step-size $w_{m+1} = \nu$ for $0 < \nu \leq 1$ in this and subsequent boosting algorithms throughout the paper. Algorithm 1 can be utilized to fit a variety of models with different base learners, including componentwise linear least squares, componentwise smoothing splines and regression trees [3]. These base learners are presented in greater detail in Section 5.1 when a convergence analysis is concerned.

As will be clear in Section 5.4, Algorithm 1 monotonically decreases the loss function (7) under certain conditions although there is no guarantee that the algorithm can locate the global minimizer for the nonconvex loss. However, with a suitable starting point, the DC algorithm converges quite often to a global one. See Tao and An [29] and the references therein. In practice, different starting points can be explored for an optimal solution. This would add computing burden to Algorithm 1 which already has two layers of iterations. Note, the initial value in line 3 is updated with the estimate in line 10. This can be named as a warm start. To partially alleviate the starting point issue, a practical strategy is to begin with a cold start. For instance, the developed computer program has an option to change line 3 from $f_0(x) = f^{(k-1)}$ to a constant vector such that every element is $\frac{1}{n}\sum_{i=1}^{n} U_i$ or 0. As a result, the loss values still monotonically decrease from line 3 to 9 under certain conditions. However, the initial loss value in the $(k+1)$-th DC outer loop (line 3 of Algorithm 1) may be larger than the last loss value in the k-th loop (line 10 of Algorithm 1). In practice, a

Algorithm 1 Difference of Convex Boosting Algorithm (DCBA)

```
 1: Input: training samples $\{(x_1, y_1), ..., (x_n, y_n)\}$, iteration count $K, M$, parameter $s$, starting point $f^{(0)}$.
 2: for $k = 1$ to $K$ do
 3:     Initialization: $f_0(x) = f^{(k-1)}$.
 4:     Construct the surrogate convex loss $L_{DCF}(y_i, f, s)$ majorizing the objective function $L$ at $f^{(k-1)}$ via (8).
 5:     for $m = 1$ to $M$ do
 6:         Compute the residuals, defined as negative gradient of loss function (see Table 4),
              $$U_i = -\frac{\partial L_{DCF}(y_i, f, s)}{\partial f}|_{f=f_{m-1}(x_i)} = -\frac{\partial \ell(y_i, f)}{\partial f}|_{f=f_{m-1}(x_i)} - \frac{\partial \ell_s(y_i, f)}{\partial f}|_{f=f^{(k-1)}}.$$
 7:         Fit a base learner $g_m$ to the residuals $U_i$ with predictors $x_i$, for $i = 1, 2, ..., n$.
 8:         Update the estimated function with a selected step-size $w_m$,
              $$f_m = f_{m-1} + w_m g_m.$$
 9:     end for
10:     Update the current estimate $f^{(k)} = f_M(x)$.
11: end for
12: Output: the classifier sign($f^{(K)}$).
```

general observation is that the last loss values decrease for $k = 1, 2, ..., K$, albeit at a slower convergence rate. With high-dimensional data as in this paper, this strategy avoids overfitting, generates better prediction and provides more parsimonious models. The numerical results in Section 6 and 7 support such a choice. Indeed, the aforementioned cold start approach follows the general consensus of early stopping of boosting before convergence of a loss function [3].

Prediction accuracy of robust boosting is greatly impacted by two tuning parameters. One is the location of truncation, which is often considered as hyper-parameter [35, 24]. For instance, consider the truncated logistic loss. If the truncation parameter s is too small towards $-\infty$, then the loss becomes similar to the standard logistic loss, which is sensitive to outliers. On the other

TABLE 4
Pseudo residuals in robust boosting algorithms at line 6 in Algoirithm 1.

Truncated loss	DCBA	Residual $U_{i,m}$
Truncated logistic	TLogitBoost	$y_i \frac{\exp(-y_i f_{m-1}(x_i))}{1+\exp(-y_i f_{m-1}(x_i))} - y_i \frac{\exp(-y_i f^{(k-1)})}{1+\exp(-y_i f^{(k-1)})} \mathrm{I}\left(y_i f^{(k-1)} < s\right)$
Difference logistic	DLogitBoost	$y_i \frac{\exp(-y_i f_{m-1}(x_i))}{1+\exp(-y_i f_{m-1}(x_i))} - y_i \frac{\exp(-y_i f^{(k-1)}-s)}{1+\exp(-y_i f^{(k-1)}-s)}$
Truncated exponential	TAdaBoost	$y_i \exp(-y_i f_{m-1}(x_i)) - y_i \exp(-y_i f^{(k-1)}) \mathrm{I}\left(y_i f^{(k-1)} < s\right)$
Truncated hinge	THingeBoost	$y_i \mathrm{I}(1 - y_i f_{m-1}(x_i) > 0) - y_i \mathrm{I}(s - y_i f^{(k-1)} > 0)$

hand, if s is too close to 0, the properties of the logistic loss can be difficult to maintain for the data. Therefore a good choice of s should be as effective as the standard logistic loss and also robust to outliers. We select s from several candidate values as in Wu and Liu [35], Park and Liu [24]. Another tuning parameter is the boosting iteration. A larger number implies better fitting for the training data and possibly more predictors in the model, which may suggest overfitting with deteriorated prediction accuracy from the test data. In practice, an optimal boosting iteration can be chosen by a data driven method, such as a cross-validation scheme or a modified Bayesian information criterion (BIC) [25]. Alternatively, we may have disjointed training/tuning data sets and use the training data for model building and tuning data for tuning parameter selection.

3.2. Adaptive difference of convex boosting algorithm

We consider boosting as a sequential algorithm that iteratively fits a base learner with selected predictors, in particular, a single predictor variable. How to choose a particular variable in each iteration is a paramount important issue in variable selection with high-dimensional data. A simple approach to reducing non-effective predictor variables is adaptive boosting. Proposed by Bühlmann and Hothorn [4], twin boosting is a generic adaptive boosting procedure with two steps: the first step is the usual boosting, and the second step is constructed to resemble the first boosting round. As a result, if a variable has not been selected in the first round of boosting, it will not be included in the second. In addition, depending on the estimates of variables selected in the first round of boosting, variables are selected with different weights in the second round of boosting. Twin boosting has much better variable selection results than the corresponding boosting algorithm and can also improve prediction accuracy [4]. Here we implement adaptive boosting (Algorithm 2) to minimize the truncated nonconvex loss function (7).

Algorithm 2 Adaptive DCBA

```
1: Input: training samples $\{(x_1, y_1), ..., (x_n, y_n)\}$, iteration count $K, M$, parameter $s$, starting point $f^{(0)}$.
2: Run first round of the DCBA (Algorithm 1) to obtain the initial function estimates $f_{init}$ and select effective predictors. For simplicity, assume the selected predictors are $x^{(1)}, x^{(2)}, ..., x^{(d)}$ where $d \le p$.
3: Run a modified DCBA among the remaining predictors to obtain the final function estimates $\hat{f}_{final}$. Specifically, denote $g^{(j)} = \left(g(x_1^{(j)}), g(x_2^{(j)}), ..., g(x_n^{(j)})\right)$ and modify line 7 in Algorithm 1 as below:
   Fit a base learner $g_m$ to the residuals $U_i$ with the $\hat{l}$-th predictor, where $\hat{l}$ is given by
       $$\hat{l} = \operatorname*{argmax}_{1 \le j \le d} \hat{C_j}^2 (2\langle U_i, g^{(j)}\rangle - \|g^{(j)}\|^2),$$
   where $\hat{C_j} = \langle \hat{f}_{init} - \overline{\hat{f}_{init}}, g^{(j)}\rangle / \|g^{(j)}\|$, $\overline{\hat{f}_{init}} = 1/n \sum_{i=1}^n \hat{f}_{init}(x_i)$, $\langle\cdot\rangle$ and $\|\cdot\|$ are the inner product and norm, respectively.
4: Output: the classifier sign($\hat{f}_{final}$).
```

4. Robust multi-class boosting

There is limited research on robust multi-class boosting algorithm. See McDonald et al. [20] for an extension of BrownBoost in this setting. We now generalize the DCBA to multi-class problems using truncated loss functions [34, 35]. We focus on robust multi-class hinge loss while the methodology can be extended to other truncated loss functions. Denote $\ell(u) = (u+1)_+$ and $\ell_s(u) = -(u-s)_+$, the truncated hinge loss $\ell(u) + \ell_s(u)$ and non-robust hinge loss $\ell(u) = (u+1)_+$ are displayed in Figure 2. As we can see from the figure, the truncated loss becomes a constant if $u \geq s$. This loss function has been utilized to develop robust multi-class classification rules. For a J-class problem with a response belonging to $\{1, ..., J\}$, denote $f = (f_1, ..., f_J)$ a J-tuple of functions. We first consider to minimize a standard hinge loss [34]:

$$L(f(x)) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J} \mathrm{I}(y_i \neq j)(f_j(x_i)+1)_+, \tag{9}$$

subject to

$$f_1(x_i) + ... + f_J(x_i) = 0. \tag{10}$$

The constraint (10) is used for uniqueness of function estimators. The classification rule is $\operatorname{argmax}_j f_j$, which shares the Bayes decision rule. For a slightly different form from (9), Wang [32] developed a boosting algorithm as well as adaptive boosting algorithm.

The robust boosting aims to minimize the objective function

$$L(f(x), s) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J} \mathrm{I}(y_i \neq j)\Big\{(f_j(x_i)+1)_+ - (f_j(x_i)-s)_+\Big\} \tag{11}$$

with constraint (10). This truncated multi-class loss is a generalization of the truncated binary hinge loss in Table 1 and holds Fisher consistency for $s \geq 0$. Different from a binary classification problem in Section 2, there are total of J differences of convex functions in (11). Given the current estimates $f_j^{(k-1)}$, the concave functions $-(f_j(x_i)-s)_+$ in (11) can be linearly majorized, thus the objective is majorized at $f_j^{(k-1)}$:

$$\begin{aligned} L(f(x), s) \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J} \mathrm{I}(y_i \neq j)\Big\{ &(f_j(x_i)+1)_+ - \Big(f_j(x_i)^{(k-1)} - s\Big)_+ \\ &- (f_j(x_i) - f_j(x_i)^{(k-1)})\mathrm{I}(f_j(x_i)^{(k-1)} \geq s)\Big\}. \end{aligned} \tag{12}$$

In an iterative process with given $f_j^{(k-1)}$, estimation of $f_j, j = 1, ..., J$ can be achieved by minimizing the right hand side of (12), or equivalently

$$\frac{1}{n}\sum_{i=1}^{n} L_{DCF}(y_i, f(x_i), s), \tag{13}$$

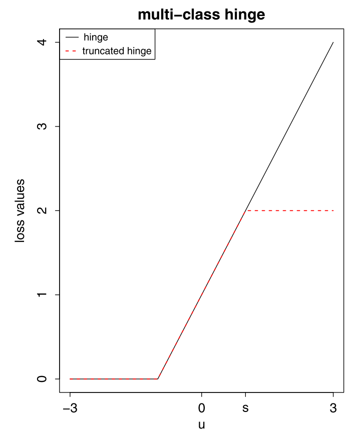


Fig 2. *Multi-class hinge loss*

where

$$L_{DCF}(y_i, f(x_i), s) = \sum_{j=1}^{J} \mathrm{I}(y_i \neq j) \left\{ (f_j(x_i) + 1)_+ - f_j(x_i)\mathrm{I}(f_j(x_i)^{(k-1)} \geq s) \right\}.$$

Since $(f_j(x_i) + 1)_+$ is a convex function and $f_j(x_i)\mathrm{I}(f_j(x_i)^{(k-1)} \geq s)$ is a linear function, the difference between these two functions is thus convex. As a result, $L_{DCF}(y_i, f(x_i), s)$ is convex, hence we can use the boosting technology. For the truncated hinge loss, we present a multi-class DCBA, Algorithm 3 (mTHingeBoost). To circumvent overfitting as in Algorithm 1, we may change line 3 and instead begin with some constant, for instance, 0. In conjunction with Algorithm 2, we can develop an adaptive robust multi-class DCBA, AmTHingeBoost. Without the outer loop in Algorithm 3, i.e., without lines 2-4, 15 and 16, and changing line 7 such that $U_{ij} = -\mathrm{I}(y_i \neq j)\mathrm{I}(f_j(x_i) + 1 > 0)$, then the modified algorithm amounts to minimizing the standard loss (9). We call this reduced algorithm mHingeBoost. This algorithm can be conveniently extended for adaptive boosting, which is referred to AmHingeBoost.

5. Convergence analysis

Convergence of functional gradient boosting has been investigated by Mason et al. [18], Grubb and Bagnell [13] among others. In this section, we provide new convergence results. We first review three commonly utilized base learners in the literature including this paper. The properties of these base learners are building blocks of the analysis. We then develop new convergence results of the functional gradient boosting algorithm. Next, we extend the results to a universal MM boosting algorithm, and evaluate Algorithm 1 as a special case. It is worth noting that we omit convergence analysis of the adaptive DCBA,

Algorithm 3 Multi-class DCBA for Truncated Hinge Loss

```
1: Input: training samples $\{(x_1, y_1), ..., (x_n, y_n)\}$, iteration count $K, M$, parameter $s$, starting points $f_j^{(0)}$ for $j = 1, ..., J$, learning rate $0 < \nu \leq 1$.
2: for $k = 1$ to $K$ do
3:   Initialization $f_j(x_i) = f_j^{(k-1)}$ for $j = 1, ..., J$.
4:   Construct the surrogate convex loss $L_{DCF}(y_i, f, s)$ majorizing the objective loss (11) at $f_j(x_i)$ via (13).
5:   for $m = 1$ to $M$ do
6:     for class $j = 1$ to $J$ do
7:       Compute $U_{ij}$ the negative gradient with respect to $f_j$:
           $$\begin{aligned} U_{ij} &= -\frac{\partial L_{DCF}(y_i, f, s)}{\partial f_j}|_{f=f_j(x_i)} \\ &= -\mathrm{I}(y_i \neq j)\left\{\mathrm{I}(f_j(x_i) + 1 > 0) - \mathrm{I}(f_j(x_i)^{(k-1)} \geq s)\right\}. \end{aligned}$$
8:       Fit a predict model $g_{mj}$ for pseudo response variable $U_{ij}$ with predictor variable $x_i$.
9:     end for
10:    for class $j = 1$ to $J$ do
11:      Set $g_{mj} \leftarrow \frac{J-1}{J}(g_{mj} - \frac{1}{J}\sum_{q=1}^{J} g_{mq})$.
12:      Set $f_j(x_i) \leftarrow f_j(x_i) + \nu g_{mj}$.
13:    end for
14:  end for
15:  Update the current estimate $f_j^{(k)} = f_j(x_i), j = 1, ..., J$.
16: end for
17: Output: the classifier $\arg\max_{1 \leq j \leq J} f_j^{(K)}$.
```

Algorithm 2. This algorithm has two rounds of Algorithm 1 while the second round has a different variable selection. As will be clear in the sequel, variable selection doesn't contribute to the convergence analysis. Hence the convergence results for the DCBA also hold for the adaptive DCBA.

5.1. Base learners in boosting

Suppose we have data $(x_1, U_1), ..., (x_n, U_n)$ where $U_i \in \mathbb{R}$ and $x_i = (x_{i1}, ..., x_{ip})^\intercal \in \mathbb{R}^p$. The design matrix X is a $n \times p$ matrix given by $X = (x_1^\intercal, x_2^\intercal, ..., x_n^\intercal)^\intercal$, and the response vector is given by $U = (U_1, U_2, ..., U_n)^\intercal$. A regression technique is to construct an estimator $\hat{U}$ of U with the predictors $x_i, i = 1, ..., n$. Many regression procedures are a linear smoother given by:

$$\hat{U} = SU, \tag{14}$$

where S is a $n \times n$ smoothing matrix whose i-th row assigns weights given to each U_i in building the estimate $\hat{U}_i$. Note, the entries of S are related to the predictors x_i but do not contain the observations U. Popular linear smoothers used as base learners in the boosting literature are reviewed. We pay special attention to their properties, which are building blocks in the subsequent convergence analysis.

5.1.1. Linear least squares

Let Z be a $n \times q$ matrix and a subset of X, such that $1 \leq q \leq \min\{n, p\}$. In addition, assume that Z is full column rank. In practice, componentwise least squares, i.e., $q = 1$ is typically employed in boosting. The smoothing or hat matrix S is given by $S = Z(Z^\intercal Z)^{-1}Z^\intercal$. The least squares prediction is given by $\hat{U} = SU$. It is well-known that the eigenvalues of the projection matrix S are either zero or one, and the number of nonzero eigenvalues is equal to the rank of the matrix. Furthermore, the hat matrix is symmetric idempotent.

5.1.2. Componentwise smoothing spline

Without loss of generality, suppose the r-th predictor x_{ir}, $1 \leq r \leq p$, is chosen by the base learner. We drop the subscript r. For observations $(x_i, U_i), i = 1, ..., n$, a smoothing spline $\hat{U} = g(x_i)$ minimizes the penalized residual sum of squares among all functions with two continuous derivatives:

$$\sum_{i=1}^{n}(U_i - g(x_i))^2 + \lambda \int (g''(x))^2 dx,$$

where λ is a pre-specified smoothing parameter. A smoothing spline is a linear operator such that $\hat{U} = SU$ [33]. The smoothing matrix S has two eigenvalues equal to 1, and the other eigenvalues are all strictly between zero and one. In addition to being positive definite, S is also symmetric but not idempotent.

5.1.3. Regression trees

We start with a bin smoother for a single predictor, also known as a regressogram [33]. As in section 5.1.2, suppose the r-th predictor x_{ir}, $1 \leq r \leq p$, is chosen by the base learner. We drop the subscript r again. Consider the observations $(x_i, U_i), i = 1, ..., n$. We define J bins with cutpoints $a_1 < ... < a_{J+1}$ where $a_1 = -\infty, a_{J+1} = \infty$:

$$R_j = \{i : a_j \leq x_i < a_{j+1}\}; \quad j = 1, ..., J.$$

An estimate $\hat{U}$ can be obtained by averaging the U_is over each bin:

$$\hat{U}(x) = \frac{1}{n_j} \sum_{i: x_i \in R_j} U_i, \quad \text{for } x \in R_j, \tag{15}$$

where n_j is the number of observations in R_j. For $x \in R_j$ define $s_i(x) = 1/n_j$ if $x_i \in R_j$ and $s_i(x) = 0$ otherwise. Thus $\hat{U}(x) = \sum_{i=1}^{n} s_i(x)U_i$. Denote a vector $s(x) = (s_1(x), s_2(x), ..., s_n(x))^\intercal$. Therefore we have

$$s(x)^\intercal = (0, 0, \ldots, \frac{1}{n_j}, \frac{1}{n_j}, \ldots, \frac{1}{n_j}, 0, \ldots, 0).$$

Define the vector of fitted values $\hat{U} = (\hat{U}(x_1), ..., \hat{U}(x_n))^\intercal$. It then follows that $\hat{U} = SU$, where S is a $n \times n$ diagonal block matrix with i-th row $s(x_i)^\intercal$, i.e., $S_{ij} = s_j(x_i), i, j = 1, ..., n$. Denote A_j a $n_j \times n_j$ sub-matrix whose every entry is $1/n_j$. In this case the matrix S is a symmetric block matrix whose diagonal blocks are A_j, and the off-diagonal blocks are matrices of zeros:

$$S = \begin{pmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_J \end{pmatrix}. \tag{16}$$

We have the following properties regarding the regressogram.

Proposition 3. *Linear smoothing matrix S of a regressogram defined by (16) is symmetric idempotent thus positive semidefinite. Furthermore, S has the eigenvalues 1 and 0, with multiplicities J and $n - J$, respectively.*

A regression tree with p predictor variables follows the same idea like the bin smoother, although the splitting variables and cut points are optimally chosen. A regression tree splits the whole predictor space into disjoint hyper-rectangles R_j with sides parallel to the coordinate axes. The regression tree has the estimates in the same form as (15). Consequently, a regression tree is a linear operator with smoothing matrix S given by (16). Like the regressogram, we have the following results:

Proposition 4. *Linear smoothing matrix S of a regression tree defined by (16) is symmetric idempotent thus positive semidefinite. Furthermore, S has the eigenvalues 1 and 0, with multiplicities J and $n - J$, respectively.*

5.2. Square integrable space

Before we evaluate the functional gradient boosting, we first review the relevant Hilbert space [13]. Given a measurable predictor set $\mathcal{X}$, a complete vector space $\mathcal{V}$ of response, and measure μ over $\mathcal{X}$, the square integrable function space $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is the set of all equivalence classes of functions $\phi : \mathcal{X} \to \mathcal{V}$ such that the Lebesgue integral $\int_{\mathcal{X}} \|\phi(x)\|_{\mathcal{V}}^2 d\mu$ is finite. This Hilbert space has a natural inner product and norm:

$$\begin{aligned} \langle \phi, \psi \rangle_\mu &= \int_{\mathcal{X}} \langle \phi(x), \psi(x) \rangle_{\mathcal{V}} d\mu, \\ \|\phi\|_\mu^2 = \langle \phi, \phi \rangle_\mu &= \int_{\mathcal{X}} \|\phi(x)\|_{\mathcal{V}}^2 d\mu. \end{aligned} \tag{17}$$

For an observed set of points $x_i, i = 1, ..., n$, the empirical inner product and norm can be correspondingly defined as the vector space operations defined by (17) reduce to the empirical versions.

We need to compute the gradient of functionals over the Hilbert space $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Let $\ell : L^2(\mathcal{X}, \mathcal{V}, \mu) \to \mathbb{R}$ be a functional, its Fréchet derivative $\nabla\ell$ is a linear operator satisfying

$$\lim_{\psi\to 0} \frac{\|\ell(f+\psi) - \ell(f) - \langle \nabla\ell(f), \psi\rangle\|}{\|\psi\|} = 0.$$

5.3. Convergence of functional gradient boosting algorithms

Given a set of observations $(x_i, y_i), i = 1, ..., n$ where $y_i \in \mathbb{R}$ and $x_i = (x_{i1}, ..., x_{ip})^\intercal \in \mathbb{R}^p$, denote $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ the hypothesis space generated by a base learner. We aim to minimize an empirical loss function

$$\operatorname*{argmin}_{f\in\Omega} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) \tag{18}$$

for a functional $\ell : L^2(\mathcal{X}, \mathcal{V}, \mu) \to \mathbb{R}$. Mason et al. [18] frame the boosting as a functional gradient descent method. The Fréchet derivative $\nabla\ell$ implies

$$\ell(f+\psi) = \ell(f) + \langle \nabla\ell(f), \psi\rangle + o(\|\psi\|).$$

Define $U = -\nabla\ell(f)$ and let $\psi = \delta g$. We then have:

$$\ell(f+\delta g) = \ell(f) - \delta\langle U, g\rangle + o(\|g\|). \tag{19}$$

Hence to first order in $\delta > 0$, (19) implies

$$\ell(f+\delta g) = \ell(f) - \delta\langle U, g\rangle.$$

For the greatest reduction in loss values, we should seek the g which maximizes $\langle U, g\rangle$. Now,

$$\begin{aligned} 2\langle U, g\rangle &= \|U\|^2 + \|g\|^2 - \|U-g\|^2 \\ &\geq -\|U-g\|^2, \end{aligned}$$

where the last equality holds only for the trivial case $\|U\| = \|g\| = 0$. Therefore seeking the solution argmax $-\|U-g\|^2$, or argmin$\|U-g\|^2$, might be sub-optimal for greatest loss reduction, compared to directly seeking argmax$\langle U, g\rangle$. However, the former strategy brings in numerous regression type base learners as presented in Section 5.1. In addition, if the solution $g^* = \text{argmin}\|U - g\|^2$ results in $\langle U, g^*\rangle > 0$, the loss ℓ is thus reduced. Some base learners including linear least squares and regression trees, indeed provide the greatest loss reduction. See Proposition 5 for details. The above discussion is closely related to the concept of edge that quantifies the performance of any given set of base learners [13]. If $\forall\nabla\ell(f) \in \Omega$, there exists a function $h \in \Omega$ such that

$$\langle \nabla\ell(f), h\rangle \geq \gamma\|\nabla\ell(f)\|\|h\|. \tag{20}$$

Then the hypothesis space Ω is defined to have an edge $\gamma \in [0,1]$. If Ω is closed under scalar multiplication, letting $U = -\nabla\ell(f), g = -h$, then (20) becomes

$$\langle U, g\rangle \geq \gamma \|U\|\|g\|. \tag{21}$$

The base learners in this paper are all generated from regression techniques, Ω is thus closed under scalar multiplication. Hence we can also claim that (21) holds for every negative gradient U if Ω has an edge γ.

In the literature, edge is related to weak learnability assumption when base learners are binary classifiers [8]. Roughly speaking, if Ω is a space of binary classifier that is more accurate than a random guess, then edge $\gamma > 0$. The relationship between the edge and base learners discussed in Section 5.1.2 can be summarized in the following results:

Proposition 5. *For a linear smoother $g = SU$, where S is a linear smoothing matrix and $U = -\nabla\ell(f)$, the following results hold:*

(i) If S is positive semidefinite, then $\langle U, g\rangle \geq 0$. Furthermore, the hypothesis space Ω has an edge γ for every U.

(ii) In particular, if S is symmetric idempotent, then the following holds:

$$\operatorname{argmax}\langle U, g\rangle = \operatorname{argmin}\|U - g\|^2.$$

Furthermore, the hypothesis space Ω has an edge γ for every U. Finally, for a constant $\zeta \neq 0$, the following equality holds:

$$\frac{\langle U, g\rangle}{\zeta\|g\|^2} = \frac{1}{\zeta}. \tag{22}$$

The equality (22) will be handy in the subsequent analysis. All three base learners described in Section 5.1.2 are linear smoothers with smoothing matrices positive semidefinite. According to Proposition 5, therefore, the hypothesis space Ω has an edge γ for every negative gradient $-\nabla\ell(f)$. Bühlmann and Hothorn [3] observed that seeking $\operatorname{argmax}\langle U, g\rangle$ [18] and seeking $\operatorname{argmin}\|U - g\|^2$ [11] coincide for the componentwise linear least squares base learner. This instance is generalized by Proposition 5 to contain more base learners, for instance, regression trees (cf. Proposition 4).

The functional gradient boosting algorithm, Algorithm 4, aims to find a solution of (18) [18, 11]. Different choices of step-size in line 5 have been proposed, including an iteration-varying number by a line search [11] and a small constant [3]. From Proposition 5, if the base learner is a linear smoother whose smoothing matrix is symmetric idempotent, then maximizing $\langle U, g\rangle$ is essentially the same as minimizing $\|U - g\|^2$ in line 4 of Algorithm 4. However, if the componentwise smoothing spline is used as the base learner, the two methods don't necessarily match since the smoothing matrix S is not idempotent.

In theoretical analysis of algorithms, it is common to assume some conditions on the loss functions. A functional ℓ is ζ-strongly smooth if $\forall \phi, \psi \in \Omega$, for some $\zeta > 0$, the following inequality holds:

$$\ell(\phi) - \ell(\psi) \leq \langle \nabla\ell(\psi), \phi - \psi\rangle + \frac{\zeta}{2}\|\phi - \psi\|^2.$$

Let $f^* = \operatorname{argmin}_{f \in \Omega} \ell(f)$. We first have a convergence result with an iteration-varying step-size.

Theorem 1. *Let $\ell(f)$ be a ζ-strongly smooth functional bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has an edge $\gamma \in (0, 1]$ for every negative gradient $-\nabla \ell(f)$. Suppose a step-size in the $(m+1)$-th boosting iteration is given by*

$$w_{m+1} = \frac{\langle -\nabla \ell(f_m), g_{m+1} \rangle}{\zeta \|g_{m+1}\|^2}, \tag{23}$$

then Algorithm 4 converges to some value, in which case

$$\|\nabla \ell(f_m)\| \to 0 \text{ as } m \to \infty.$$

Furthermore, after M iterations, the following convergence rate is obtained:

$$\min_{0 \le m \le M-1} \|\nabla \ell(f_m)\| \le \frac{1}{\sqrt{M}} \left(\frac{2\zeta}{\gamma^2} \left(\ell(f_0) - \ell(f^*) \right) \right)^{1/2}.$$

The results in Theorem 1 are different from Theorem 12.3 in Mason et al. [18] in that we have adopted the concept of edge. In addition, there are at least two other differences: the base learner here is not restricted to a classifier and may be a regression type base learner; and we may minimize the least squares $\|-\nabla \ell(f_m) - g_{m+1}\|^2$ rather than maximizing $\langle -\nabla \ell(f_m), g_{m+1} \rangle$. Notice if the boosting base learner is a linear operator $g_{m+1} = S(-\nabla \ell(f_m))$, and the associated linear smoothing matrix S is symmetric idempotent, then (23) reduces to a constant $w_{m+1} = \frac{1}{\zeta}$ based on Proposition 5. Finally, Theorem 1 yields very similar results as to a gradient method in function estimation. See, for instance, Nesterov [23].

For a gradient method in function estimation, a constant step-size and iteration-varying step-size can be unified in the convergence analysis [23]. For a functional gradient method, however, the properties of a constant step-size in Algorithm 4 have not been well addressed [3]. The following theorem fills the gap. The results may be applied to linear smoother base learners including linear least squares and regression trees.

Algorithm 4 Functional Gradient Boosting Algorithm (FGB)

```
1: **Input:** training samples $\{(x_1, y_1), ..., (x_n, y_n)\}$, iteration count $M$, starting point $f_0(x)$.
2: **for** $m = 1$ to $M$ **do**
3:     Compute the residuals, defined as negative gradient of loss function,
           $U_i = -\frac{\partial \ell(y_i, f)}{\partial f}|_{f=f_{m-1}(x_i)}$
4:     Fit a base learner $g_m$ to the residuals $U_i$ with predictors $x_i$, for $i = 1, 2, ..., n$.
5:     Update the estimated function with a selected step-size $w_m$,
           $f_m = f_{m-1} + w_m g_m.$
6: **end for**
7: **Output:** $f_M$.
```

Theorem 2. *Let $\ell(f)$ be a ζ-strongly smooth functional bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that Ω has an edge $\gamma \in (0, 1]$. Assume that at the $(m+1)$-th iteration the boosting base learner is a linear operator, and the associated linear smoothing matrix S_{m+1} is symmetric idempotent. Given a constant step-size, then Algorithm 4 converges to some value, in which case*

$$\|S_{m+1} \nabla \ell(f_m)\| \to 0 \text{ as } m \to \infty.$$

Furthermore, after M iterations, the following convergence rate is obtained:

$$\min_{0 \le m \le M} \|S_{m+1} \nabla \ell(f_m)\| \le \frac{1}{\sqrt{M}} \left(\frac{2\zeta}{\gamma^2} (\ell(f_0) - \ell(f^*)) \right)^{1/2}.$$

5.4. Convergence of MM boosting algorithms

Given a set of observations $(x_i, y_i), i = 1, ..., n$, where $y_i \in \mathbb{R}$ and $x_i = (x_{i1}, ..., x_{ip})^\intercal \in \mathbb{R}^p$, denote Ω the hypothesis space generated by a base learner. We seek the solution of the problem

$$\operatorname*{argmin}_{f \in \Omega} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \tag{24}$$

for a functional $L : L^2(\mathcal{X}, \mathcal{V}, \mu) \to \mathbb{R}$. From now on, we suppress y in the loss functions. We consider a general MM algorithm. Suppose the objective $L(f)$ is majorized by a surrogate loss $H_k(f)$ at the majorization point $f^{(k-1)}$ in the k-th outer MM loop, $k = 1, 2, ...$, such that

$$L(f) \le H_k(f) \quad \text{for all } f \in \Omega, \quad L(f^{(k-1)}) = H_k(f^{(k-1)}). \tag{25}$$

Note, $H_k(f)$ depends on $f^{(k-1)}$ by the design. The surrogate loss $H_k(f)$ can be minimized or reduced to obtain a new estimate $f^{(k)}$. Therefore the DC surrogate function characterized by (5) is a special case of (25). Denote $\hbar_k(f)$ the error between the surrogate loss and the objective given by $\hbar_k(f) = H_k(f) - L(f)$. The descent property of the MM framework is described below:

Proposition 6. *Suppose a loss functional L is bounded below, and a surrogate functional H_k is defined by (25), then $L(f^{(k)})$ and $H_k(f^{(k)})$ monotonically decrease and converge to the same value. Furthermore, $\hbar_k(f^{(k)}) \to 0$ as $k \to \infty$.*

We consider an implementation of the boosting algorithm in the MM framework, Algorithm 5, which is a generalization of Algorithm 1. We begin with some preliminary results concerning the convergence analysis. Denote $f_m^{(k)}$ the estimate at the m-th inner boosting iteration within the k-th outer MM loop, for $k = 1, 2, ..., m = 0, 1, 2, ..., M$. Note $f_m^{(0)}$ is not defined in Algorithm 5. For notational convenience, however, denote

$$f_M^{(0)} \triangleq f^{(0)}. \tag{26}$$

Algorithm 5 MM Boosting Algorithm (MMBA)

```
1: **Input:** training samples $\{(x_1, y_1), ..., (x_n, y_n)\}$, iteration count $K, M$, starting point $f^{(0)}$.
2: for $k = 1$ to $K$ do
3:     Initialization: $f_0(x) = f^{(k-1)}$.
4:     Construct the surrogate convex loss $H_k(f)$ majorizing $L$ at $f_0(x)$.
5:     for $m = 1$ to $M$ do
6:         Compute the residuals, defined as negative gradient of loss function
               $U_i = -\frac{\partial H_k(f)}{\partial f}|_{f=f_{m-1}(x_i)}$
7:         Fit a base learner $g_m$ to the residuals $U_i$ with predictors $x_i$, for $i = 1, 2, ..., n$.
8:         Update the estimated function with a selected step-size $w_m$,
               $f_m = f_{m-1} + w_m g_m.$
9:     end for
10:    Update the current estimate $f^{(k)} = f_M(x)$.
11: end for
12: Output: $f^{(K)}$.
```

Notice that $f_0^{(k)}$ is the initial boosting estimate at the k-th MM outer loop. The equality at the majorization points between the surrogate and the objective is stated in the next lemma.

Lemma 3. *For a loss functional L and surrogate loss functional H_k defined by (25), the following results hold for Algorithm 5:*

$$H_k(f_0^{(k)}) = H_k(f_M^{(k-1)}) = L(f_M^{(k-1)}). \tag{27}$$

As a consequence, (27) also holds for Algorithm 1 which is a special case of Algorithm 5.

A concept of strongly convex functional is often needed in the convergence analysis. A functional ℓ over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is η-strongly convex if $\forall \phi, \psi \in \Omega$, for some $\eta > 0$, the following inequality holds:

$$\ell(\phi) - \ell(\psi) \geq \langle \nabla \ell(\psi), \phi - \psi \rangle + \frac{\eta}{2} \|\phi - \psi\|^2. \tag{28}$$

For the sum of convex loss and concave loss functions (2), utilizing the special form of the surrogate loss (4), the following lemma suggests that we may only need to focus on the convex loss concerning the properties of the surrogate loss.

Lemma 4. *Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is linearly majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$ and $h_k(f)$ is given by (3). Then the following results hold:*

(i) If $\ell(f)$ is ζ-strongly smooth, then $H_k(f)$ is ζ-strongly smooth.
(ii) If $\ell(f)$ is η-strongly convex, then $H_k(f)$ is η-strongly convex.

Let $f^* = \text{argmin}_{f \in \Omega} L(f)$. We first have a convergence result of Algorithm 5 with an iteration-varying step-size.

Theorem 5. *Consider a loss functional $L(f)$ bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $L(f)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$. Assume that $H_k(f)$ is a ζ-strongly smooth functional over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has an edge $\gamma \in (0, 1]$ for every negative gradient $-\nabla H_k(f)$. Given a starting point $f^{(0)}$ and suppose a step-size w_{m+1} in the $(m+1)$-th boosting iteration within the k-th outer MM loop is given by*

$$w_{m+1} = \frac{\langle -\nabla H_k(f_m^{(k)}), g_{m+1} \rangle}{\zeta \| g_{m+1} \|^2}, \tag{29}$$

then Algorithm 5 converges to some value, in which case:

$$\|\nabla H_k(f_m^{(k)})\| \to 0 \text{ as } \begin{cases} k \to \infty, \forall m \\ m \to \infty, \forall k. \end{cases}$$

Furthermore, the following convergence rate is obtained:

$$\min_{0 \le m \le M-1, 1 \le k \le K} \|\nabla H_k(f_m^{(k)})\| \le \frac{1}{\sqrt{MK}} \left(\frac{2\zeta}{\gamma^2} \left(L(f^{(0)}) - L(f^*) \right) \right)^{1/2}.$$

Notice if the boosting base learner is a linear operator, and the associated linear smoothing matrix is symmetric idempotent, then the step-size (29) reduces to a constant $w_{m+1} = \frac{1}{\zeta}$ based on Proposition 5. With Theorem 5, the following results for the DC framework directly follow from Lemma 4.

Corollary 5.1. *Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$, and $h_k(f)$ is given by (3). Assume that $\ell(f)$ is a ζ-strongly smooth functional over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has an edge $\gamma \in (0, 1]$ for every negative gradient $-\nabla H_k(f)$. Given a starting point $f^{(0)}$ and suppose a step-size w_{m+1} in the $(m+1)$-th boosting iteration within the k-th outer DC loop is given by (29), then Algorithm 1 converges to some value, in which case:*

$$\|\nabla H_k(f_m^{(k)})\| \to 0 \text{ as } \begin{cases} k \to \infty, \forall m \\ m \to \infty, \forall k. \end{cases}$$

Furthermore, the following convergence rate is obtained:

$$\min_{0 \le m \le M-1, 1 \le k \le K} \|\nabla H_k(f_m^{(k)})\| \le \frac{1}{\sqrt{MK}} \left(\frac{2\zeta}{\gamma^2} \left(L(f^{(0)}) - L(f^*) \right) \right)^{1/2}.$$

For a constant step-size, we have the following results which may be applied to linear smoother base learners including linear least squares and regression trees.

Theorem 6. *Suppose a loss functional $L(f)$ bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$. Assume that $H_k(f)$ is a ζ-strongly smooth functional over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has* *an edge $\gamma \in (0,1]$ for every negative gradient $-\nabla H_k(f)$. Suppose that at the $(m+1)$-th boosting iteration within the k-th outer MM loop, the boosting base learner is a linear operator, and the associated linear smoothing matrix $S_{m+1}^{(k)}$ is symmetric idempotent. Given a starting point $f^{(0)}$ and a constant step-size, then Algorithm 5 converges to some value, in which case:*

$$\|S_{m+1}^{(k)} \nabla H_k(f_m^{(k)})\| \to 0 \text{ as } \begin{cases} k \to \infty, \forall m \\ m \to \infty, \forall k. \end{cases}$$

Furthermore, the following convergence rate is obtained:

$$\min_{0 \le m \le M, 1 \le k \le K} \|S_{m+1}^{(k)} \nabla H_k(f_m^{(k)})\| \le \frac{1}{\sqrt{MK}} \left(\frac{2\zeta}{\gamma^2} \left(L(f^{(0)}) - L(f^*) \right) \right)^{1/2}.$$

With Theorem 6, the following results for the DC framework directly follow from Lemma 4.

Corollary 6.1. *Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ bounded below $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$, and $h_k(f)$ is given by (3). Assume that $\ell(f)$ is a ζ-strongly smooth functional over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has an edge $\gamma \in (0,1]$ for every negative gradient $-\nabla H_k(f)$. Suppose that at the $(m+1)$-th boosting iteration within the k-th outer DC loop, the boosting base learner is a linear operator, and the associated linear smoothing matrix $S_{m+1}^{(k)}$ is symmetric idempotent. Given a starting point $f^{(0)}$ and a constant step-size, then Algorithm 1 converges to some value, in which case:*

$$\|S_{m+1}^{(k)} \nabla H_k(f_m^{(k)})\| \to 0 \text{ as } \begin{cases} k \to \infty, \forall m \\ m \to \infty, \forall k. \end{cases}$$

Furthermore, the following convergence rate is obtained:

$$\min_{0 \le m \le M, 1 \le k \le K} \|S_{m+1}^{(k)} \nabla H_k(f_m^{(k)})\| \le \frac{1}{\sqrt{MK}} \left(\frac{2\zeta}{\gamma^2} \left(L(f^{(0)}) - L(f^*) \right) \right)^{1/2}.$$

In Proposition 6, if we choose $f_\dagger^{(k)} = \operatorname{argmin}_{f \in \Omega} H_k(f)$, Proposition 6 implies that $L(f_\dagger^{(k)})$ and $H_k(f_\dagger^{(k)})$ monotonically decrease and converge to some value, say $L(f^\ddagger)$. In general and in particular if L is nonconvex, there is no guarantee that $L(f^\ddagger) = L(f^*)$. Nevertheless, we can explore how Algorithm 5 performs when $L(f^\ddagger) = L(f^*)$. For convenience of analysis we study Algorithm 6. Unlike Algorithm 5 with two layers of loops, Algorithm 6 is a simplified version with only one loop. This follows the same strategy as Krause and Singer [14]. We first prepare some results before the convergence analysis is conducted. In particular, we generalize strong convexity to the following condition, which will be useful in a constant step-size analysis:

Algorithm 6 Simple MM Boosting Algorithm (SMMBA)

```
1: Input: training samples $\{(x_1, y_1), ..., (x_n, y_n)\}$, iteration count $K$, starting point $f^{(0)}$.
2: for $k = 1$ to $K$ do
3:     Construct the surrogate convex loss $H_k(f)$ majorizing the objective function $L$ at $f^{(k-1)}$.
4:     Compute the residuals, defined as negative gradient of loss function
           $$U_i = -\frac{\partial H_k(f)}{\partial f}|_{f=f^{(k-1)}}$$
5:     Fit a base learner $g_k$ to the residuals $U_i$ with predictors $x_i$, for $i = 1, 2, ..., n$.
6:     Update the estimated function with a selected step-size $w_k$,
           $$f^{(k)} = f^{(k-1)} + w_k g_k.$$
7: end for
8: Output: $f^{(K)}$.
```

Condition 1. *For a loss functional ℓ over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ and $\forall \phi, \psi \in \Omega$,*

$$\ell(\phi) - \ell(\psi) \geq \langle S\nabla\ell(\psi), \phi - \psi\rangle + \frac{\eta}{2}\|\phi - \psi\|^2 \tag{30}$$

holds for a symmetric idempotent matrix S and some constant $\eta > 0$.

Clearly, if (28) holds, then (30) holds, in which case S is the identity matrix. However, Condition 1 doesn't necessarily imply strongly convex. The following conclusion is analogous to that of a strongly convex functional:

Proposition 7. *If Condition 1 holds for a functional ℓ, $\forall \psi \in \Omega$, the following inequality holds:*

$$\nabla\ell(\psi)^\intercal S\nabla\ell(\psi) \geq 2\eta\left(\ell(\psi) - \ell(\psi^*)\right), \tag{31}$$

where ψ^* is a minimization point of ℓ.

Replacing S with the identity matrix, (31) simplifies to a standard result for strongly convex ℓ (cf. p. 460 in Boyd and Vandenberghe [2]):

$$\|\nabla\ell(\psi)\|^2 \geq 2\eta\left(\ell(\psi) - \ell(\psi^*)\right). \tag{32}$$

The next theorem describes the performance of Algorithm 6.

Theorem 7. *Suppose a loss functional $L(f)$ bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$. Furthermore, $H_k(f)$ is a ζ-strongly smooth functional over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has an edge $\gamma \in (0, 1]$ for every negative gradient $-\nabla H_k(f)$. Given a starting point $f^{(0)}$, consider two cases below:*

(i) Assume that $H_k(f)$ is a η-strongly convex functional, and a step-size w_{k+1} is given by

$$w_{k+1} = \frac{\langle -\nabla H_{k+1}(f^k), g_{k+1}\rangle}{\zeta\|g_{k+1}\|^2}. \tag{33}$$

(ii) Assume that functional H_{k+1} satisfies Condition 1, and a constant step-size is given.

After K iterations of Algorithm 6, the bound on $L(f^{(K)}) - L(f^*)$ is given by

$$\begin{aligned} L(f^{(K)}) - L(f^*) \leq (1 - \frac{\gamma^2\eta}{\zeta})^K \left(L(f^{(0)}) - L(f^*)\right) \\ + \frac{\gamma^2\eta}{\zeta}\sum_{k=1}^{K}(1 - \frac{\gamma^2\eta}{\zeta})^{K-k}\left(H_k(f_\dagger^{(k)}) - L(f^*)\right). \end{aligned} \tag{34}$$

Furthermore, utilizing Proposition 6 to assume that the errors between the surrogate and objective are bounded by

$$H_k(f_\dagger^{(k)}) - L(f^*) \leq \beta(1 - \frac{\gamma^2\eta}{\zeta})^k, \tag{35}$$

where $\beta > 0$ is a constant, then the following convergence rate is obtained:

$$L(f^{(K)}) - L(f^*) \leq (1 - \frac{\gamma^2\eta}{\zeta})^K \left(L(f^{(0)}) - L(f^*) + K\beta\frac{\gamma^2\eta}{\zeta}\right). \tag{36}$$

Notice if the boosting base learner is a linear operator, and the associated linear smoothing matrix is symmetric idempotent, then the step-size (33) reduces to a constant $w_{k+1} = \frac{1}{\zeta}$ based on Proposition 5. Grubb and Bagnell [13] developed a convergence result on strongly smooth and strongly convex loss functions for Algorithm 4. Compared with Theorem 3 in Grubb and Bagnell [13], the right hand side of (34) has an additional second error term that is a bound characterizing the errors between the surrogate loss and the objective introduced at each iteration. Regardless whether L is nonconvex, a convergence rate of $O(K(1 - \frac{\gamma^2\eta}{\zeta})^K$ is obtained when $H_k(f_\dagger^{(k)})$ converges to $L(f^*)$ at the rate of $O((1 - \frac{\gamma^2\eta}{\zeta})^K)$ for K iterations. The latter rate is the same as boosting convergence rate for minimizing a ζ-strongly smooth and η-strongly convex functional [13]. While the assumption (35) seems difficult to verify, it suggests that the convergence rate of Algorithm 6 and hence Algorithm 5 relies on how tight a surrogate loss approximates the objective. Intuitively we should seek surrogate loss functions as close as possible to the original objective, yet easier to minimize than the latter.

With Theorem 7, the following results concerning an iteration-varying step-size for the DC framework can be developed, again with the aid of Lemma 4.

Corollary 7.1. *Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ bounded below over $L^2(\mathcal{X}, \mathcal{V}, \mu)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$, and $h_k(f)$ is given by (3). Assume that $\ell(f)$ is a ζ-strongly smooth and η-strongly convex functional over $L^2(\mathcal{X}, \mathcal{V}, \mu)$. Assume that $\Omega \subset L^2(\mathcal{X}, \mathcal{V}, \mu)$ has an edge $\gamma \in (0, 1]$ for every negative gradient $-\nabla H_k(f)$. Given a starting point $f^{(0)}$ and suppose a step-size w_{k+1} is provided by (33),* *after K iterations of Algorithm 6, the bound on $L(f^{(K)}) - L(f^*)$ is given by*

$$L(f^{(K)}) - L(f^*) \leq (1 - \frac{\gamma^2\eta}{\zeta})^K \left(L(f^{(0)}) - L(f^*)\right) + \frac{\gamma^2\eta}{\zeta}\sum_{k=1}^{K}(1 - \frac{\gamma^2\eta}{\zeta})^{K-k}\left(H_k(f_\dagger^{(k)}) - L(f^*)\right).$$

Furthermore, utilizing Proposition 6 to assume that the errors between the surrogate and objective are bounded by

$$H_k(f_\dagger^{(k)}) - L(f^*) \leq \beta(1 - \frac{\gamma^2\eta}{\zeta})^k,$$

where $\beta > 0$ is a constant, then the following convergence rate is obtained:

$$L(f^{(K)}) - L(f^*) \leq (1 - \frac{\gamma^2\eta}{\zeta})^K \left(L(f^{(0)}) - L(f^*) + K\beta\frac{\gamma^2\eta}{\zeta}\right).$$

5.5. Examples

In the previous sections, general convergence theories are provided for boosting algorithms including FGB, MMBA, DCBA and SMMBA. In this section, we apply these theories to some standard and robust loss functions in Table 1. Recall $y = \{+1, -1\}$.

5.5.1. Logistic loss

We first consider the logistic loss $\ell(y, f)$ given by

$$\ell(y, f) = \log(1 + \exp(-yf)). \tag{37}$$

The partial second derivative is given by

$$\begin{aligned}\frac{\partial^2\ell(y,f)}{\partial f^2} &= \frac{\exp(yf)}{(1+\exp(yf))^2} \\ &= \frac{1}{1/\exp(yf) + 2 + \exp(yf)} \\ &\leq \frac{1}{4}.\end{aligned} \tag{38}$$

The last inequality is trivially obtained by substituting $a = \sqrt{1/\exp(yf)}, b = \sqrt{\exp(yf)}$ in the following inequality:

$$a^2 + b^2 \geq 2ab.$$

It is straightforward to argue that the logistic loss is $\frac{1}{4}$-strongly smooth (cf. pp. 460-461 in Boyd and Vandenberghe [2]). This, together with other assumptions in the theorems, Theorem 1 and 2 thus hold for Algorithm 4 when applying to the logistic loss.

Consider the truncated logistic loss or difference logistic loss (cf. Table 2) $L(f,s) = \ell(f) + \ell_s(f)$, where $\ell(f)$ is given by (37) and $\ell_s(f) = -\big(\log(1+\exp(-u)) - \log(1+\exp(-s))\big)_+$ or $\ell_s(f) = -\log(1+\exp(-u-s))$ with $u = yf$. Since $\ell(f)$ is $\frac{1}{4}$-strongly smooth, along with other assumptions in the corollaries, Corollary 5.1 and 6.1 hold for Algorithm 1 when applying to the truncated logistic loss and difference logistic loss.

Note, for fixed y, (38) implies

$$\lim_{f\to\infty} \frac{\partial^2 \ell(y,f)}{\partial f^2} \to 0.$$

Hence the strongly convex condition in Corollary 7.1 is not satisfied for the logistic loss. However, some minor modifications shown below can restrict the support to a compact set, which in turn can force the logistic loss strongly convex. To maintain numerical stability, it has been suggested to clap the range of f such that $f \in [-\tau, \tau]$ for some constant $\tau > 0$ [12]. Therefore we have $yf \in [-\tau, \tau]$ since $y \in \{-1, 1\}$, which leads to

$$\begin{aligned}\frac{\partial^2 \ell(y,f)}{\partial f^2} &= \frac{1}{1/\exp(yf) + 2 + \exp(yf)} \\ &\geq \frac{1}{2 + 2\exp(\tau)}.\end{aligned}$$

The last inequality is obtained from $1/\exp(yf) \leq \exp(\tau), \exp(yf) \leq \exp(\tau)$. Hence $\ell(y,f)$ is $\frac{1}{2+2\exp(\tau)}$-strongly convex. This, in conjunction with other assumptions in the corollary, Corollary 7.1 holds for Algorithm 6 when applying to the truncated logistic loss and difference logistic loss.

5.5.2. *Exponential loss*

Consider the exponential loss $\ell(y,f)$ given by

$$\ell(y,f) = \exp(-yf). \tag{39}$$

The second derivative is given by $\frac{\partial^2 \ell(y,f)}{\partial f^2} = \exp(-yf)$. Therefore $\ell(y,f)$ is smooth and convex, but neither strongly smooth nor strongly convex. Again, we can clap the range of f, such that $f \in [-\tau, \tau]$ for some constant $\tau > 0$. Therefore we have $yf \in [-\tau, \tau]$ as before, which leads to

$$\exp(-\tau) \leq \frac{\partial^2 \ell(y,f)}{\partial f^2} \leq \exp(\tau).$$

With the restricted range $f \in [-\tau, \tau]$, $\ell(y,f)$ is thus strongly smooth and strongly convex. Combining with other assumptions, Theorem 1 and 2 hold for Algorithm 4 when applying to the restricted exponential loss.

Consider the truncated exponential loss $L(f,s) = \ell(f) + \ell_s(f)$, where $\ell(f)$ is given by (39) and $\ell_s(f) = -\max\big(0, \exp(-u) - \exp(-s)\big), u = yf, f \in [-\tau, \tau]$. As above, combining with other assumptions, Corollary 5.1 and 6.1 thus hold for Algorithm 1 when applying to the truncated exponential loss. Similarly, combining with other assumptions, Corollary 7.1 holds for Algorithm 6 when applying to the truncated exponential loss.

6. A simulation study

In the simulated examples, we study performance of the proposed robust boosting algorithms and compare with the standard non-robust boosting methods and existing robust boosting algorithms. Random data generations are similar to those in Wu and Liu [35], Park and Liu [24]. Example 1 and 2 are binary classification problems while example 3 is a multi-class problem. These examples include both linear and nonlinear classifiers.

1. In example 1, predictor variables (x_1, x_2) are uniformly sampled from a unit disk $x_1^2 + x_2^2 \leq 1$ and $y = 1$ if $x_1 \geq x_2$ and -1 otherwise.
2. In example 2, predictor variables (x_1, x_2) are uniformly sampled from a unit disk $x_1^2 + x_2^2 \leq 1$ and $y = 1$ if $(x_1 - x_2)(x_1 + x_2) < 0$ and -1 otherwise.
3. In example 3, we consider a 3-class problem. Predictor variables (x_1, x_2) are uniformly sampled from a unit disk $x_1^2 + x_2^2 \leq 1$. For a J-class problem, the class label $y = \left\lfloor \frac{J\theta}{2\pi} \right\rfloor$, where $\lfloor \cdot \rfloor$ is the integer part function, and θ is the radian phase angle measured counter-clockwisely.

In each example, we also generate another 18 noise variables from uniform[-1, 1]. Different level of outliers are introduced to the data. We randomly select v percent of the date and switch their class labels for binary problems; or switch their class labels to other classes with equal probabilities in example 3. We consider v=0, 5, 10 and 20. We generated random samples of training/tuning/test samples. Training data were used for model estimation, tuning samples were used to select optimal boosting iteration in the DCBA, and test data were used to evaluate classification accuracy. For the binary classification problems, RobustBoost requires choosing two parameters, the error goal and margin goal. The error goal is the proportion of the outlier observations, which can be easy to set up in the simulated data [7]. The margin goal was chosen based on the tuning data. For example 3, we also evaluated the multi-class BrownBoost and pre-specified the tuning parameter following McDonald et al. [20]. With the simulated data, the performance of an algorithm can be compared with the optimal Bayes errors. In example 1 and 2, the training/tuning/test sample sizes are $n = 200/200/10,000$ from 100 random samples. In example 3, the training/tuning/test sample sizes are $100/1,000/10,000$ from 50 random samples.

The DCBA in Table 4 was evaluated. For example 1 and 3, componentwise linear least squares were base learner, while componentwise smoothing splines were base learner in example 2. The following candidate values of the truncation location s were investigated. For truncated exponential loss, $s =$

$0, -\log 2, -\log 3$; for truncated hinge loss, $s = 0, -1, -2$; for truncated logistic loss, $s = 0, -\log 3, -\log 7$; for logistic difference loss, $s = \log 2, \log 4, \log 8$ except for $s = \log 16$ for example 2 and v=0 and 5. For truncated multi-class hinge loss, $s = 0, 0.5, 1, 1.5, 2$. With candidate values of s described above, we only report the results from which the best prediction was obtained. The results are summarized in Tables 5 to 10. With no contamination, each DCBA and its corresponding non-robust boosting algorithm perform similarly. With data contaminated with outliers, the DCBA is more accurate than its non-robust counterpart in prediction and variable selection. Overall, the DCBA provides smaller misclassification errors and eliminate more noise variables. Compared to existing robust boosting algorithms, the DCBA outperforms RobustBoost in example 1. This is also the case in example 2 except for THingeBoost which has slightly poorer accuracy than RobustBoost but maintains more parsimonious variable selection. In example 3, the robust multi-class HingeBoost performs better than BrownBoost with smaller errors and number of selected variables. To further understand the dynamics of robust boosting with contaminated data, the DCBA is compared with non-robust counterparts in Figure 3 to 6. For each DCBA, we plot the evolutions of misclassification error on the test data as well as the number of variables selected, versus the boosting iteration. Clearly, the DCBA performs better than their non-robust counterparts. While we illustrate the patterns using data generated in example 1 with 20% contamination, similar conclusions hold for other examples as well as other data contamination scenarios (figures not shown).

We also evaluate the performance of the adaptive DCBA. We only illustrate selected results. For example 2, we focus on variations of HingeBoost. The results are summarized in Table 7, Table 8 and Figure 7. The adaptive DCBA (ATHingeBoost) performs better than its robust cousin: THingeBoost. It also outperforms the adaptive non-robust cousin: AHingeBoost. AThingeBoost generated more accurate classification and variable selection than the two competitors. For example 3, the results from the adaptive multi-class DCBA are summarized in Table 9 and 10. The adaptive robust boosting (AmHingeBoost)

Table 5
Misclassification error in Example 1

Method	v=0	v=5	v=10	v=20
Bayes				
	0	0.05	0.1	0.2
Nonrobust				
LogitBoost	0.0293(0.0114)	0.0806(0.0151)	0.1335(0.0137)	0.2449(0.0172)
AdaBoost	0.0224(0.0103)	0.0793(0.0159)	0.1344(0.0149)	0.2451(0.0176)
HingeBoost	0.0161(0.0144)	0.0658(0.0152)	0.1206(0.0145)	0.2275(0.0177)
Robust				
TLogitBoost	0.0292(0.0114)	0.0774(0.0140)	0.1275(0.0135)	0.2311(0.0170)
DLogitBoost	0.0320(0.0128)	0.0809(0.0149)	0.1328(0.0152)	0.2376(0.0160)
TAdaBoost	0.0222(0.0102)	0.0652(0.0091)	0.1162(0.0092)	0.2252(0.0140)
THingeBoost	0.0163(0.0145)	0.0643(0.0139)	0.1172(0.0144)	0.2196(0.0153)
RobustBoost	0.0391(0.0156)	0.0893(0.0169)	0.1419(0.0169)	0.243(0.0212)

TABLE 6
Number of selected variables in Example 1

Method	v=0	v=5	v=10	v=20
Nonrobust				
LogitBoost	2(0)	2(0.2)	2.5(0.7)	3.7(1.6)
AdaBoost	2(0)	2.7(1.2)	3.1(1.4)	4.0(2.0)
HingeBoost	2.1(0.4)	2.2(0.6)	2.7(1.2)	3.4(1.6)
Robust				
TLogitBoost	2(0)	2(0)	2.1(0.3)	2.4(1.1)
DLogitBoost	2(0)	2(0)	2(0.1)	2.5(0.8)
TAdaBoost	2(0)	3.1(1.4)	3.8(1.9)	5.0(2.0)
THingeBoost	2.1(0.4)	2.2(0.5)	2.4(0.8)	2.8(1.4)
RobustBoost	7.6(5.1)	7.1(4.7)	8.7(5.6)	9.6(6.0)

TABLE 7
Misclassification error in Example 2

Method	v=0	v=5	v=10	v=20
Bayes				
	0	0.05	0.1	0.2
Nonrobust				
LogitBoost	0.0482(0.0228)	0.1041(0.0207)	0.1595(0.0209)	0.2737(0.0230)
AdaBoost	0.0543(0.0241)	0.1076(0.0225)	0.1622(0.0224)	0.2741(0.0237)
HingeBoost	0.0673(0.0271)	0.1238(0.0299)	0.1802(0.0267)	0.2884(0.0252)
AHingeBoost	0.0579(0.0268)	0.1132(0.0254)	0.1670(0.0229)	0.2765(0.0235)
Robust				
TLogitBoost	0.0486(0.0234)	0.0981(0.021)	0.1497(0.0194)	0.2637(0.0219)
DLogitBoost	0.0505(0.0230)	0.1024(0.0208)	0.1553(0.021)	0.2661(0.0238)
TAdaBoost	0.0543(0.0244)	0.1044(0.0210)	0.1542(0.0207)	0.2680(0.0223)
THingeBoost	0.0673(0.0272)	0.1191(0.0272)	0.1712(0.0252)	0.2797(0.0239)
ATHingeBoost	0.0579(0.0269)	0.1083(0.0247)	0.1607(0.0238)	0.2642(0.0227)
RobustBoost	0.0610(0.0189)	0.1180(0.0200)	0.1709(0.0215)	0.2716(0.0250)

TABLE 8
Number of selected variables in Example 2

Method	v=0	v=5	v=10	v=20
Nonrobust				
LogitBoost	2.1(0.3)	3.0(1.2)	3.9(1.9)	4.5(2.6)
AdaBoost	2.0(0.1)	2.6(0.9)	3.1(1.3)	4.1(2.1)
HingeBoost	3.9(2)	4.7(2.5)	5.9(3.1)	6.2(3.4)
AHingeBoost	2(0.1)	2(0.2)	2.2(0.5)	3.6(1.5)
Robust				
TLogitBoost	2.1(0.3)	2.3(0.6)	2.7(1.2)	4.0(2.1)
DLogitBoost	2.0(0.1)	2.4(0.7)	2.5(0.8)	3.5(1.7)
TAdaBoost	2(0.1)	2.3(0.7)	2.8(1.4)	3.9(1.9)
THingeBoost	3.9(2)	4.2(2.2)	5.3(2.5)	5.9(2.9)
ATHingeBoost	2(0.1)	2(0.2)	2(0.2)	2.6(0.9)
RobustBoost	8.3(5.2)	9.6(5.1)	8.7(5.4)	10.0(6.1)

Table 9
Misclassification error in Example 3

Method	v=0	v=5	v=10	v=20
Bayes				
	0	0.05	0.1	0.2
Nonrobust				
mHingeBoost	0.0604(0.046)	0.1146(0.0508)	0.1731(0.0556)	0.315(0.0664)
AmHingeBoost	0.0486(0.0619)	0.1063(0.0633)	0.1641(0.0669)	0.315(0.088)
Robust				
mTHingeBoost	0.0558(0.0447)	0.1101(0.0497)	0.1678(0.0541)	0.2949(0.065)
AmTHingeBoost	0.0431(0.0426)	0.095(0.0457)	0.1546(0.0563)	0.2854(0.0808)
BrownBoost	0.0751(0.0141)	0.1360(0.0135)	0.1949(0.0169)	0.3132(0.0233)

Table 10
Number of selected variables in Example 3

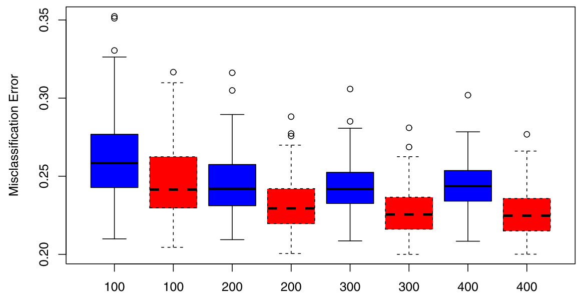
Method	v=0	v=5	v=10	v=20
Nonrobust				
mHingeBoost	4.1(2.6)	4.2(2.5)	4.7(2.4)	6(3.1)
AmHingeBoost	2.5(1.1)	2.5(1)	2.8(1.2)	2.8(1)
Robust				
mTHingeBoost	4.3(2.8)	4.1(2.2)	4.8(2.6)	5.1(2.8)
AmTHingeBoost	2.6(1.2)	2.4(0.7)	2.7(1.1)	2.5(0.7)
BrownBoost	19.9(0.2)	18.3(1.4)	17.1(1.7)	14.3(2.1)

performs better than either robust boosting (mTHingeBoost) or adaptive nonrobust boosting (AmThingeBoost).

7. Data applications

7.1. Analysis of healthcare costs

The Kids' Inpatient Database (KID) is part of a family of databases developed for the US Healthcare Cost and Utilization Project. The KID contains a sample of pediatric discharges from more than 4,100 U.S. community hospitals in 44 states, sampling from 10% of uncomplicated in-hospital births and 80% of other pediatric cases. Acute kidney injury (AKI) is a common hospital complication with a rising incidence and a strong association with mortality and overall morbidity outcomes, especially in critically ill children. We predict inpatient costs for children with AKI who where $\leq$ 18 years old in the 2009 KID. Inclusion and exclusion criteria follow Sutherland et al. [28]. There are total of $n = 10,322$ AKI cases. The cost of AKI inpatient care for a discharge can be estimated by multiplying total hospital charge by the group average all-payer inpatient cost/charge ratio. Although healthcare costs prediction if often considered a regression like problem, there are some merits to be treated as a classification problem. For instance, one can evaluate whether a risk factor improves classification accuracy. In this paper, hospital charges are dichotomized at a threshold of \$50,000. Potential risk factors include patient demographics, characteristics of admission and hospitals, payer information, discharge status and more than 260



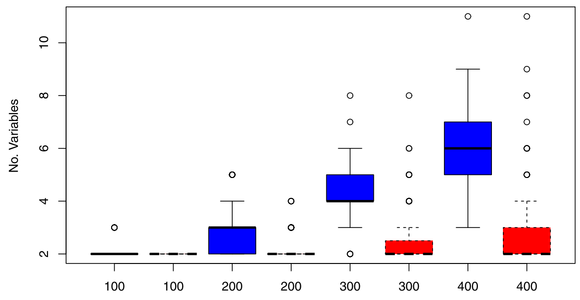
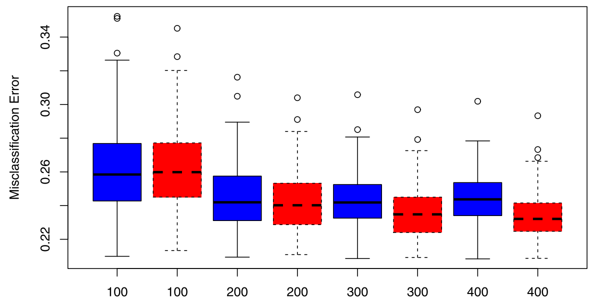


FIG 3. *Misclassification errors and variable selection for LogitBoost (solid blue box plots) and truncated LogitBoost (dashed red box plots) for boosting iterations for example 1, n=200, 20% contamination.*

diagnostic codes classified by Clinical Classification Software. There are $p = 282$ potential predictors which do not contain procedural codes and hospital length of stay.

The four DCBA in Table 4 were applied to the data. Two thirds of the data were randomly selected as training data, and the remaining test data were used to evaluate prediction accuracy. We used trees (stumps) as base learner in boosting algorithms. Different truncation locations were evaluated, and we only illustrate selected results. For truncated LogitBoost, $s = -2$, for difference LogitBoost, $s = 3.3$, for truncated AdaBoost, $s = -1$, and for truncated HingeBoost, $s = -1.05$. For the DCBA and its non-robust counterpart, Figure 8 to 11 plot the evolution of the misclassification error on the test data versus the iteration counter, as the algorithms proceed while working on the test set. These figures also plot the evolution of the number of variables selected versus the iteration counter, as the boosting algorithms proceed while working on the training set. When boosting iterations were selected by five-fold cross validation with the training data, the derived models generated results in Table 11. The best prediction accuracy of 86% was obtained with the truncated AdaBoost (TAdaBoost). The truncated LogitBoost (TLogitBoost) was better than its non-robust counterpart LogitBoost, the difference LogitBoost (DLogitBoost) was comparable to



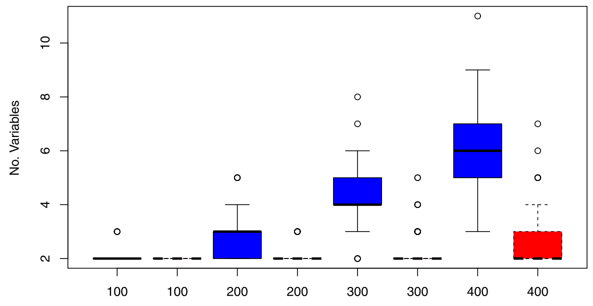


Fig 4. *Misclassification errors and variable selection for LogitBoost (solid blue box plots) and difference LogitBoost (dashed red box plots) for boosting iterations for example 1, n=200, 20% contamination.*

LogitBoost, and truncated Hingeboost (THingeBoost) was slightly better than HingeBoost. In addition, the DCBA provided more sparse variable selection than their non-robust counterparts. For instance, while more accurate in classification, TAdaBoost selected 84 risk factors compared to 127 with AdaBoost. With truncated loss functions, the advantage of DCBA on variable selection is clearly demonstrated in Figure 8 to 11. We used one data realization to illustrate the difference between DCBA and boosting, particularly in Figure 8 to 11. Multiple random data realizations showed similar conclusions, and the results are not presented here. We also performed RobustBoost on the data. We considered different error goals and chose the optimal margin goal based on five-fold cross validation. The best test error and the associated variable selection are presented in Table 11. Except for truncated HingeBoost, the DCBA is more accurate than RobustBoost. In general, RobustBoost selected more risk factors in the predictive model.

7.2. Classifying breast cancer clinical status

The data contains 278 breast cancer samples with 164 estrogen receptor positive cases [27]. Following Zhang et al. [37], Li and Bradic [16], we conducted

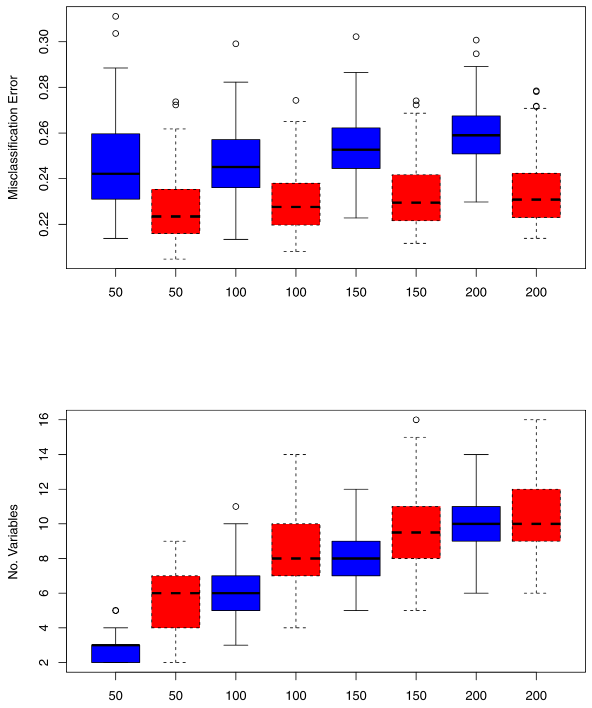


Fig 5. *Misclassification errors and variable selection for AdaBoost (solid blue box plots) and truncated AdaBoost (dashed red box plots) for boosting iterations for example 1, n=200, 20% contamination.*

a pre-screening procedure to reduce computation demand. From 22,283 genes, the dataset was reduced to 3000 genes with the largest absolute values of the two-sample t-statistics. The remaining genes were then standardized. We randomly split the data to training and test data. The training data consists 50/50 positive/negative estrogen receptor status, and the rest was the test data. The positive/negative outcome status in the training data was randomly exchanged with percentage v=0, 5, 10, 15. We used componentwise linear least squares as base learner in boosting algorithms. While different truncation location parameters were evaluated, we only report results from selected truncation parameters due to space limitation. For TLogitBoost, DLogitBoost, TAdaBoost and THingeBoost, truncation location $s = -0.2, 0.8, -0.2, -0.5$, respectively. A five-fold cross-validation was implemented to select the optimal boosting iterations with the training data. The above data generation and analysis were repeated 100 times. The results are summarized in Table 12 and 13. For comparison, the test errors of RobustBoost in Li and Bradic [16] are also reproduced in Table 12. Note, variable selection results of RobustBoost were not presented in that paper. For non-contaminated data, all methods show similar prediction while hinge loss based boosting algorithms select more variables. The same dataset was studied in Zhang et al. [37], Li and Bradic [16] where the best test error in ten methods

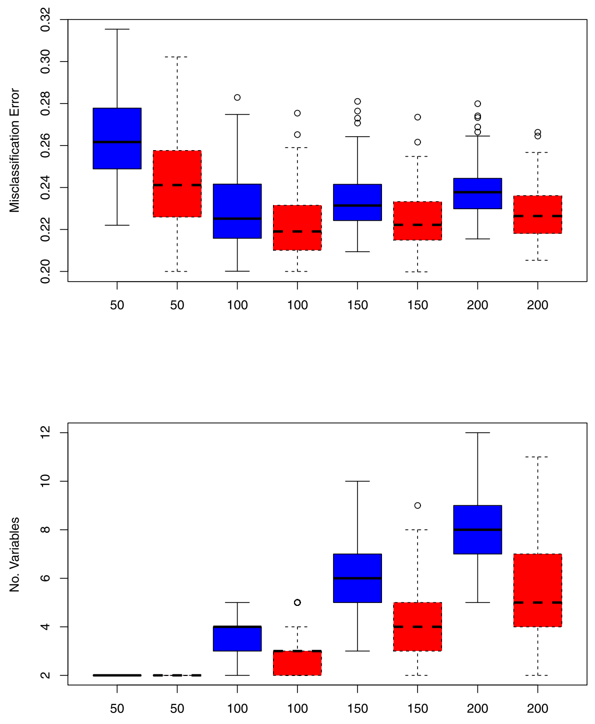


Fig 6. *Misclassification errors and variable selection for HingeBoost (solid blue box plots) and truncated HingeBoost (dashed red box plots) for boosting iterations for example 1, n=200, 20% contamination.*

was 0.0945, which is larger than six methods in Table 12 with v=0. For contaminated data with outliers, the truncated loss functions improve prediction accuracy. The DCBA has better test error than its non-robust boosting counterpart. For instance, with 15% outliers, THingeBoost has test error 0.1197 compared to 0.1379 with HingeBoost. As the level of outlier increases, classification is more difficult, and more variables are selected. The robust methods in Li and Bradic [16] found the best test errors 0.1116 (v=5), 0.1217 (v=10) and 0.1329 (v=15), respectively. Clearly, the proposed TLogitBoost and THingeBoost have better classification accuracy.

8. Discussion

The results from the simulation studies and data example considered in this paper strongly suggest that robust gradient boosting is an important alternative to the traditional boosting in many applications. In particular, the numerical results suggest that robust boosting can provide more accurate classification and variable selection.

In this paper, the convergence analysis is centered on smooth loss functions. However, the hinge loss and truncated hinge loss are both nonsmooth functions.

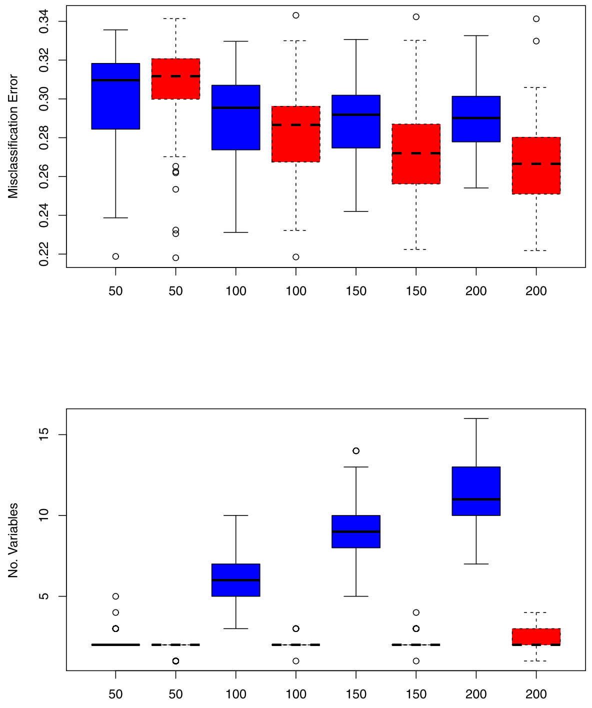


FIG 7. *Misclassification errors and variable selection for adaptive HingeBoost (solid blue box plots) and adaptive truncated HingeBoost (dashed red box plots) for boosting iterations for example 2, n=200, 20% contamination.*

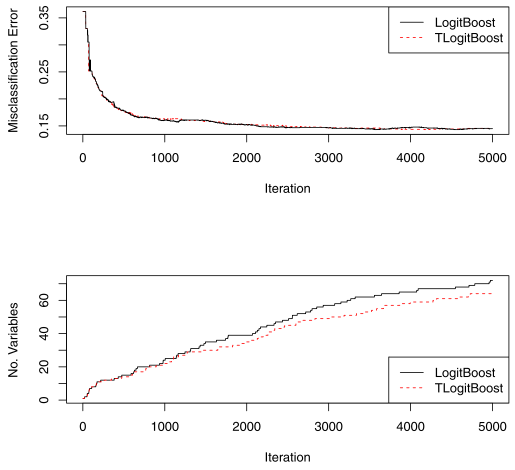
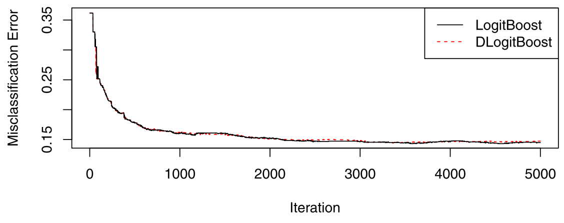


FIG 8. *Misclassification errors and variable selection for LogitBoost (solid black) and truncated LogitBoost (dashed red) for boosting iterations for AKI data.*



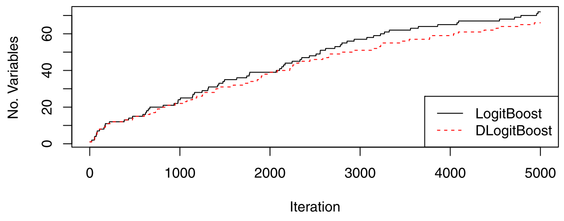
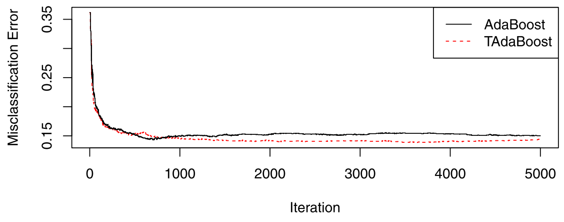


Fig 9. *Misclassification errors and variable selection for LogitBoost (solid black) and difference LogitBoost (dashed red) for boosting iterations for AKI data.*



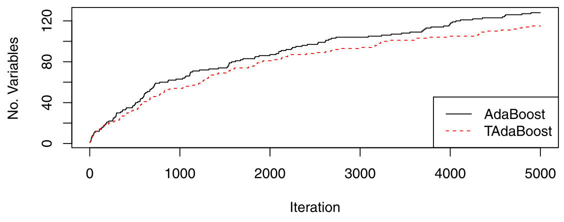
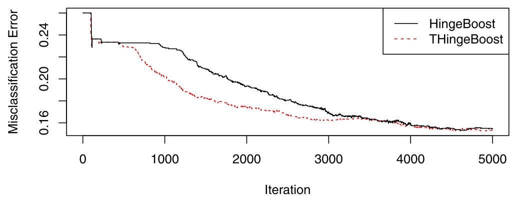


Fig 10. *Misclassification errors and variable selection for AdaBoost (solid black) and truncated AdaBoost (dashed red) for boosting iterations for AKI data.*



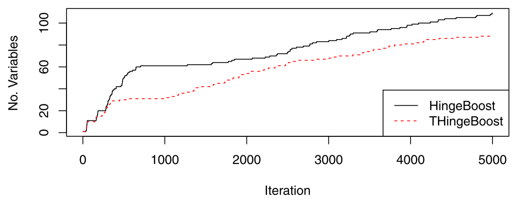


FIG 11. *Misclassification errors and variable selection for HingeBoost (solid black) and truncated HingeBoost (dashed red) for boosting iterations for AKI data.*

TABLE 11
Classification of AKI Costs

Method	Misclassification Error	No. Variables
Nonrobust		
LogitBoost	0.1455	64
AdaBoost	0.1508	127
HingeBoost	0.1550	104
Robust		
TLogitBoost	0.1431	59
DLogitBoost	0.1455	57
TAdaBoost	0.1407	84
THingeBoost	0.1532	88
RobustBoost	0.1521	119

There is a lack of convergence analysis on such loss functions for functional gradient algorithm, or Algorithm 4. Indeed, Algorithm 4 may break down for nonsmooth loss functions [13]. Further analysis of the proposed DCBA on nonsmooth loss functions is needed to be done.

Asymptotic theories of various boosting algorithms have been elaborated in the literature, for instance, see an early review Bühlmann and Hothorn [3]. However, there are only limited asymptotic theories developed for Algorithm 4 despite its wide applications particularly in the statistical community [3, 19]. Much of the previous results have focused on special loss functions such as the least squares loss [3], or special algorithms like AdaBoost. As an extension of Algorithm 4, asymptotic consistency of Algorithm 1 and 5 remains a future research topic. Theorem 7 provides convergence rate analysis for Algorithm 6. It is also interesting to investigate the convergence speed of Algorithm 1 and 5.

The gradient descent view of boosting with the aid of difference of convex

Table 12
Misclassification error in breast cancer data

Method	v=0	v=5	v=10	v=15
Nonrobust				
LogitBoost	0.0907(0.0145)	0.1027(0.0218)	0.1341(0.0444)	0.1735(0.0703)
AdaBoost	0.0923(0.0156)	0.1072(0.0237)	0.1333(0.0407)	0.1585(0.0473)
HingeBoost	0.0931(0.0158)	0.0996(0.0230)	0.1165(0.0414)	0.1379(0.0492)
Robust				
TLogitBoost	0.0884(0.0131)	0.0920(0.0182)	0.1125(0.0453)	0.1326(0.0646)
DLogitBoost	0.0898(0.0163)	0.0997(0.0265)	0.1316(0.0562)	0.1708(0.0787)
TAdaBoost	0.0978(0.0198)	0.1097(0.0269)	0.1289(0.0344)	0.1511(0.0425)
THingeBoost	0.0937(0.0149)	0.0957(0.0173)	0.1076(0.0295)	0.1197(0.0411)
RobustBoost	0.1024(0.0210)	0.1182(0.0295)	0.1222(0.0304)	0.1329(0.0313)

Table 13
Number of selected variables in breast cancer data

Method	v=0	v=5	v=10	v=15
Nonrobust				
LogitBoost	2.5(1.8)	3.6(2.2)	5.3(2.9)	6.9(3.6)
AdaBoost	4.1(3.8)	6.6(5.6)	8.6(6.0)	11.7(7.0)
HingeBoost	6.6(7.8)	8.9(9.1)	8.8(8.4)	12.4(9.8)
Robust				
TLogitBoost	1.8(1.1)	1.8(1.0)	2.1(1.3)	2.4(1.5)
DLogitBoost	1.4(0.8)	1.7(0.9)	2.5(1.5)	2.9(1.4)
TAdaBoost	4.5(3.7)	7.6(5.4)	8.8(5.7)	8.7(6.2)
THingeBoost	8.4(9.0)	9.4(9.5)	9.6(8.7)	11.2(9.0)

allow us to design boosting algorithms for a variety of robust loss functions. This new approach can be extended to other nonconvex loss functions including but not limited to truncated loss. We anticipate that more innovative boosting algorithms will be developed coupling with other MM algorithms in addition to the difference of convex algorithm.

Appendix A: Proofs

Proof of Proposition 1: Let $\hat{s} = \log(1+\exp(-s))$. We first consider $u \to -\infty$. For the difference logistic loss we have

$$\begin{aligned}
\lim_{u\to-\infty} \{L_{DL}(u,\hat{s})\} &= \lim_{u\to-\infty} \{\log(1+\exp(-u)) - \log(1+\exp(-u-\hat{s}))\} \\
&= \lim_{u\to-\infty} \left\{\log \frac{1+\exp(-u)}{1+\exp(-u-\hat{s})}\right\} \\
&= \log \frac{1}{\exp(-\hat{s})} \\
&= \hat{s} \\
&= \log(1+\exp(-s)).
\end{aligned}$$

For the truncated logistic loss we have

$$\begin{aligned}\lim_{u\to-\infty}\{L_{TL}(u,s)\} &= \lim_{u\to-\infty}\{\min(\log(1+\exp(-u)),\log(1+\exp(-s)))\}\\ &= \log(1+\exp(-s)).\end{aligned}$$

Therefore we have

$$\lim_{u\to-\infty}\{L_{DL}\left(u,\log(1+\exp(-s))\right)-L_{TL}(u,s)\}=0.$$

For the case $u\to\infty$ we have

$$\begin{aligned}\lim_{u\to\infty}\{L_{DL}(u,\hat{s})\} &= \lim_{u\to\infty}\{\log(1+\exp(-u))-\log(1+\exp(-u-\hat{s}))\}\\ &= 0,\\ \lim_{u\to\infty}\{L_{TL}(u,s)\} &= \lim_{u\to\infty}\{\min(\log(1+\exp(-u)),\log(1+\exp(-s)))\}\\ &= 0.\end{aligned}$$

Therefore we have

$$\lim_{u\to\infty}\{L_{DL}\left(u,\log(1+\exp(-s))\right)-L_{TL}(u,s)\}=0.$$

Proof of Proposition 2: Since $E(L_{DL}(Yf(X),s)) = E[E(L_{DL}(Yf(X),s)|X=x)]$, to minimize $E(L_{DL}(Yf(X),s))$, we can instead minimize $E(L_{DL}(Yf(X),s)|X=x)$ for every x. For any fixed x,

$$E(L_{DL}(Yf(X),s)|X=x)=p(x)L_{DL}(f(x),s)+(1-p(x))L_{DL}(-f(x),s),$$

where $p(x)$ is the conditional probability of class $+1$ given x. Without loss of generality, we only consider $p(x)>1/2$. Note if $p(x)>1/2$, we have the minimizer $f^*(x)\ge 0$. Otherwise, suppose $f^*(x)<0$, we then have $f^*(x)<-f^*(x)$, leading to

$$(2p(x)-1)\big(L_{DL}(f^*(x),s)-L_{DL}(-f^*(x),s)\big)>0,$$

since $L_{DL}(f(x),s)$ is a decreasing function of $f(x)$. Equivalently, we obtain

$$\begin{aligned}p(x)L_{DL}\left(f^*(x),s\right)+(1-p(x))\,L_{DL}\left(-f^*(x),s\right) > p(x)L_{DL}\left(-f^*(x),s\right)\\ +(1-p(x))L_{DL}\left(f^*(x),s\right),\end{aligned}$$

which is contradictory to that $f^*(x)$ is the minimizer. Hence $f^*(x)\ge 0$. For any $s>0$ we have

$$\begin{aligned}\frac{d}{df(x)}\Big\{p(x)L_{DL}(f(x),s)+(1-p(x))L_{DL}(-f(x),s)\Big\}|_{f(x)=0} &= \frac{\exp(-s)-1}{2(1+\exp(-s))}\\ &< 0.\end{aligned}$$

Hence $f^*(x)$ can't be 0. Therefore it must be $f^*(x)>0$. In conclusion, $f^*(x)$ has the same sign as $p(x)-1/2$.

Proof of Proposition 3: For the diagonal block matrix S defined by (16), S is a symmetric block matrix whose diagonal blocks are A_j, and the off-diagonal blocks are matrices of zeros. In addition, each A_j is a $n_j \times n_j$ sub-matrix whose every entry is $1/n_j$. Therefore each A_j is symmetric. The above arguments imply that S is symmetric. Furthermore, we have

$$S^2 = \begin{pmatrix} A_1^2 & & & \\ & A_2^2 & & \\ & & \ddots & \\ & & & A_J^2 \end{pmatrix}.$$

It can be easily shown that $A_j, j = 1, ..., J$, are idempotent (cf. Searle [26], p. 322). Thus $S^2 = S$, i.e., S is idempotent.

Denote I the identity matrix. Since S is a diagonal block matrix, the eigenvalues can be computed as below:

$$\det(S - \lambda I) = \det(A_1 - \lambda I) \det(A_2 - \lambda I) ... \det(A_J - \lambda I).$$

The eigenvalues of S are thus just the list of eigenvalues of each A_j. For each A_j, the eigenvalues are 1 and 0, with multiplicities 1 and $n_j - 1$, respectively (cf. Searle [26], p. 322). Therefore the eigenvalues of S are 1 and 0, with multiplicities J and $\sum_{j=1}^{J}(n_j - 1) = n - J$, respectively.

Proof of Proposition 4 is the same as for Proposition 3.

Proof of Proposition 5:

(i) Since $g = SU$ and S is positive semidefinite, we have

$$\begin{aligned} \langle U, g \rangle &= \langle U, SU \rangle \\ &= U^\intercal SU \\ &\geq 0. \end{aligned}$$

Furthermore, if $\gamma = 0$, then (21) holds. Thus Ω has an edge $\gamma \in [0, 1]$ for every negative gradient $U = -\nabla \ell(f)$.

(ii) Since $g = SU$, we have

$$\begin{aligned} \|U - g\|^2 &= \|U - SU\|^2 \\ &= \|U\|^2 - U^\intercal SU - U^\intercal S^\intercal U + U^\intercal S^\intercal SU \\ &= \|U\|^2 - U^\intercal SU \\ &= \|U\|^2 - \langle U, g \rangle. \end{aligned}$$

The third equality is obtained since $S^\intercal = S$ and $S^2 = S$. Furthermore, since $\|U\|^2$ is a constant,

$$\begin{aligned} \operatorname{argmin} \|U - g\|^2 &= \operatorname{argmin} -\langle U, g \rangle \\ &= \operatorname{argmax} \langle U, g \rangle. \end{aligned}$$

Since S is symmetric idempotent, S is positive semidefinite. Thus as in (i), Ω has an edge $\gamma \in [0,1]$ for every U. Finally, for the S defined above, it is simple to prove the following equality:

$$\frac{\langle U, g \rangle}{\zeta \|g\|^2} = \frac{1}{\zeta}. \tag{40}$$

Proof of Theorem 1: The proof is an adaptation of the corresponding proof of the convergence of standard gradient method for minimizing a smooth function [23]. We analyze boosting iterations on minimizing the loss ℓ. By the assumption that ℓ is strongly smooth, we have:

$$\ell(f_{m+1}) \leq \ell(f_m) + \langle \nabla \ell(f_m), f_{m+1} - f_m \rangle + \frac{\zeta}{2} \|f_{m+1} - f_m\|^2.$$

Let $f_{m+1} = f_m + w_{m+1} g_{m+1}$ and consider w_{m+1} given by $w_{m+1} = \frac{\langle -\nabla \ell(f_m), g_{m+1} \rangle}{\zeta \|g_{m+1}\|^2}$. We then have

$$\ell(f_{m+1}) \leq \ell(f_m) - \frac{1}{2\zeta} \frac{\langle -\nabla \ell(f_m), g_{m+1} \rangle^2}{\|g_{m+1}\|^2}.$$

With the edge requirement $\gamma > 0$, we have

$$\langle -\nabla \ell(f_m), g_{m+1} \rangle \geq \gamma \|\nabla \ell(f_m)\| \|g_{m+1}\|.$$

Therefore we have

$$\ell(f_{m+1}) \leq \ell(f_m) - \frac{\gamma^2}{2\zeta} \|\nabla \ell(f_m)\|^2. \tag{41}$$

Since $\ell(f_m)$ monotonically decreases for $m = 0, 1, ..., M$, and ℓ is bounded below, $\ell(f_m)$ converges. Summing up these inequalities (41) for $m = 0, ..., M-1$, we obtain

$$\frac{\gamma^2}{2\zeta} \sum_{m=0}^{M-1} \|\nabla \ell(f_m)\|^2 \leq \ell(f_0) - \ell(f_M) \leq \ell(f_0) - \ell(f^*), \tag{42}$$

where $f^* = \operatorname{argmin} \ell(f)$. From (42), we know

$$\|\nabla \ell(f_m)\| \to 0 \text{ as } m \to \infty.$$

Furthermore, denote

$$\alpha_M = \min_{0 \leq m \leq M-1} \|\nabla \ell(f_m)\|.$$

Then, using (42) we have

$$\alpha_M \leq \frac{1}{\sqrt{M}} \left(\frac{2\zeta}{\gamma^2} \left(\ell(f_0) - \ell(f^*) \right) \right)^{1/2}.$$

Proof of Theorem 2: Again we adapt the convergence analysis in Nesterov [23] for the proof. We analyze boosting iterations on minimizing the loss ℓ. By the assumption that ℓ is strongly smooth, we have

$$\ell(f_{m+1}) \leq \ell(f_m) + \langle \nabla \ell(f_m), f_{m+1} - f_m \rangle + \frac{\zeta}{2} \|f_{m+1} - f_m\|^2.$$

Let $f_{m+1} = f_m + wg_{m+1}, U_m = -\nabla\ell(f_m)$. By the assumption that line 4 of Algorithm 4 returns $g_{m+1} = S_{m+1}U_m$, where S_{m+1} is symmetric idempotent, we have

$$\begin{aligned}
\ell(f_{m+1}) &\leq \ell(f_m) - wU_m^{\intercal}S_{m+1}U_m + \frac{w^2\zeta}{2}\|S_{m+1}U_m\|^2 \\
&= \ell(f_m) - wU_m^{\intercal}S_{m+1}^{\intercal}S_{m+1}U_m + \frac{w^2\zeta}{2}\|S_{m+1}U_m\|^2 \\
&= \ell(f_m) + (-w + \frac{w^2\zeta}{2})\|S_{m+1}U_m\|^2.
\end{aligned}$$

Consider minimizing the following function:

$$\Delta(w) = -w + \frac{w^2\zeta}{2}.$$

Computing the derivative of this function, we obtain the optimal step-size satisfying $\Delta'(w) = \zeta w - 1 = 0$. Therefore $w^* = \frac{1}{\zeta}$ is a minimum of $\Delta(w)$ since $\Delta''(w) = \zeta > 0$. Note $\gamma \in (0, 1]$, hence the one step of the boosting decreases the value of loss at least as follows:

$$\begin{aligned}
\ell(f_{m+1}) &\leq \ell(f_m) - \frac{1}{2\zeta}\|S_{m+1}U_m\|^2 \\
&\leq \ell(f_m) - \frac{\gamma^2}{2\zeta}\|S_{m+1}U_m\|^2.
\end{aligned} \tag{43}$$

Since $\ell(f_m)$ monotonically decreases for $m = 0, 1, ..., M$, and ℓ is bounded below, $\ell(f_m)$ converges. Summing up these inequalities (43) for $m = 0, ..., M-1$, we obtain

$$\frac{\gamma^2}{2\zeta}\sum_{m=0}^{M-1}\|S_{m+1}U_m\|^2 \leq \ell(f_0) - \ell(f_M) \leq \ell(f_0) - \ell(f^*). \tag{44}$$

Hence (44) leads to

$$\|S_{m+1}\nabla\ell(f_m)\| = \|S_{m+1}U_m\| \to 0 \text{ as } m \to \infty.$$

Furthermore, denote

$$\theta_M = \min_{0\leq m\leq M}\|S_{m+1}\nabla\ell(f_m)\|.$$

Then, using (44) we have

$$\theta_M \leq \frac{1}{\sqrt{M}}\left(\frac{2\zeta}{\gamma^2}\left(\ell(f_0) - \ell(f^*)\right)\right)^{1/2}.$$

Proof of Proposition 6: Given $f^{(k-1)}$, $f^{(k)}$ is obtained such that the surrogate loss $H_k(f)$ is reduced. Together with (25), we have

$$L(f^{(k)}) \leq H_k(f^{(k)}) \leq H_k(f^{(k-1)}) = L(f^{(k-1)}). \tag{45}$$

Therefore $L(f^{(k)})$ monotonically decreases for $k = 1, 2, \ldots$. Because L is bounded below, $L(f^{(k)})$ converges to some value $L(f^{\ddagger})$. It also holds that

$$\hbar_k(f^{(k)}) = H_k(f^{(k)}) - L(f^{(k)}) \geq 0.$$

Similarly we have

$$L(f^{(k+1)}) \leq H_{k+1}(f^{(k+1)}) \leq H_{k+1}(f^{(k)}) = L(f^{(k)}). \tag{46}$$

Combining (45) and (46), we have

$$L(f^{(k+1)}) \leq H_{k+1}(f^{(k+1)}) \leq H_k(f^{(k)}) \leq L(f^{(k-1)}).$$

Hence $H_k(f^{(k)})$ monotonically decreases for $k = 1, 2, \ldots$. By the Squeeze Theorem for Convergence Sequences, we have $H_k(f^{(k)}) \to L(f^{\ddagger})$, or

$$\hbar_k(f^{(k)}) = H_k(f^{(k)}) - L(f^{(k)}) \to 0$$

as $k \to \infty$.

Proof of Lemma 3: We first show that

$$f_0^{(k)} = f_M^{(k-1)} \tag{47}$$

for $k = 1, 2, \ldots$. Denote $f^{(0)}$ the starting point as before. Consider two cases. If $k = 1$, with (26), then $f_M^{(0)} = f^{(0)}$ holds for the initial point, and $f_0^{(k)} = f^{(0)}$ by line 3 in Algorithm 5. Together, for $k = 1$, we have $f_0^{(k)} = f_M^{(k-1)}$. If $k > 1$, then the initial point $f_0^{(k)}$ in the k-th outer MM loop is the value $f_M^{(k-1)}$ obtained in the last (i.e. M-th) boosting inner iteration within the $(k-1)$-th outer MM loop. To summarize, we have shown (47) holds. In addition, $f_M^{(k-1)}$ or equivalently, $f_0^{(k)}$, for $k = 1, \ldots, K$, are majorization points by design in line 4 in Algorithm 5. Combining (47) and (25), we have

$$H_k(f_0^{(k)}) = H_k(f_M^{(k-1)}) = L(f_M^{(k-1)}).$$

Proof of Lemma 4: Consider $H_k(f) = \ell(f) + h_k(f)$, where $h_k(f)$ is given by (3). Denote

$$\nabla \ell_s(f^{(k-1)}) \triangleq \frac{\partial \ell_s(f)}{\partial f}\Big|_{f=f^{(k-1)}}.$$

We then have

$$\begin{aligned} H_k(f) &= \ell(f) + h_k(f) \\ &= \ell(f) + \ell_s(f^{(k-1)}) + \langle \nabla \ell_s(f^{(k-1)}), f - f^{(k-1)} \rangle. \end{aligned} \tag{48}$$

Taking derivative on both sides of the last equation, we have

$$\nabla H_k(f) = \nabla \ell(f) + \nabla \ell_s(f^{(k-1)}). \tag{49}$$

From (48) and (49) we have

$$H_k(\phi) = \ell(\phi) + \ell_s(f^{(k-1)}) + \langle \nabla \ell_s(f^{(k-1)}), \phi - f^{(k-1)} \rangle,$$
$$H_k(\psi) = \ell(\psi) + \ell_s(f^{(k-1)}) + \langle \nabla \ell_s(f^{(k-1)}), \psi - f^{(k-1)} \rangle,$$
$$\nabla H_k(\psi) = \nabla \ell(\psi) + \nabla \ell_s(f^{(k-1)}).$$

With the last three equalities, it is a simple substitution to show that

$$\begin{aligned} & H_k(\phi) - H_k(\psi) - \langle \nabla H_k(\psi), \phi - \psi \rangle \\ &= \ell(\phi) + \langle \nabla \ell_s(f^{(k-1)}), \phi \rangle - \ell(\psi) - \langle \nabla \ell_s(f^{(k-1)}), \psi \rangle \\ &\quad - \langle \nabla \ell(\psi) + \nabla \ell_s(f^{(k-1)}), \phi - \psi \rangle \\ &= \ell(\phi) - \ell(\psi) - \langle \nabla \ell(\psi), \phi - \psi \rangle. \end{aligned} \tag{50}$$

We are ready to prove the two conclusions in the lemma:

(i) Suppose functional ℓ is ζ-strongly smooth, we then have

$$\ell(\phi) - \ell(\psi) \le \langle \nabla \ell(\psi), \phi - \psi \rangle + \frac{\zeta}{2}\|\phi - \psi\|^2. \tag{51}$$

Combining (50) and (51), we obtain

$$H_k(\phi) - H_k(\psi) \le \langle \nabla H_k(\psi), \phi - \psi \rangle + \frac{\zeta}{2}\|\phi - \psi\|^2.$$

This completes the proof that H_k is ζ-strongly smooth.

(ii) Now suppose functional ℓ is η-strongly convex:

$$\ell(\phi) - \ell(\psi) \ge \langle \nabla \ell(\psi), \phi - \psi \rangle + \frac{\eta}{2}\|\phi - \psi\|^2. \tag{52}$$

Combining (50) and (52), we have

$$H_k(\phi) - H_k(\psi) \ge \langle \nabla H_k(\psi), \phi - \psi \rangle + \frac{\eta}{2}\|\phi - \psi\|^2.$$

Thus H_k is η-strongly convex. Indeed, it can be shown more generally, if $\ell(f)$ is η-strongly convex, and $h(f)$ is convex, then $\ell(f)+h(f)$ is η-strongly convex.

Proof of Theorem 5: We analyze boosting iterations on minimizing the surrogate loss H_k. Denote $f_m^{(k)}$ the m-th inner boosting iteration within the k-th outer MM loop, for $k = 1, 2, ..., m = 0, 1, 2, ...$ By the assumption that H_k is strongly smooth, we have

$$H_k(f_{m+1}^{(k)}) \le H_k(f_m^{(k)}) + \langle \nabla H_k(f_m^{(k)}), f_{m+1}^{(k)} - f_m^{(k)} \rangle + \frac{\zeta}{2}\|f_{m+1}^{(k)} - f_m^{(k)}\|^2.$$

Let $f_{m+1}^{(k)} = f_m^{(k)} + w_{m+1} g_{m+1}$ and consider w_{m+1} given by

$$w_{m+1} = \frac{\langle -\nabla H_k(f_m^{(k)}), g_{m+1} \rangle}{\zeta \|g_{m+1}\|^2}.$$

We have

$$H_k(f_{m+1}^{(k)}) \leq H_k(f_m^{(k)}) - \frac{1}{2\zeta}\frac{\langle \nabla H_k(f_m^{(k)}), g_{m+1}\rangle^2}{\|g_{m+1}\|^2}.$$

With the edge requirement $\gamma > 0$, we have

$$\langle -\nabla H_k(f_m^{(k)}), g_{m+1}\rangle \geq \gamma \|\nabla H_k(f_m^{(k)})\| \|g_{m+1}\|.$$

We have

$$H_k(f_{m+1}^{(k)}) \leq H_k(f_m^{(k)}) - \frac{\gamma^2}{2\zeta}\|\nabla H_k(f_m^{(k)})\|^2. \tag{53}$$

Summing up these inequalities (53) for $m = 0, ..., M-1$, we have

$$H_k(f_M^{(k)}) \leq H_k(f_0^{(k)}) - \frac{\gamma^2}{2\zeta}\sum_{m=0}^{M-1}\|\nabla H_k(f_m^{(k)})\|^2. \tag{54}$$

By (25), we have

$$L(f_M^{(k)}) \leq H_k(f_M^{(k)}). \tag{55}$$

Combining Lemma 3 and (54)-(55), we have

$$L(f_M^{(k)}) \leq L(f_M^{(k-1)}) - \frac{\gamma^2}{2\zeta}\sum_{m=0}^{M-1}\|\nabla H_k(f_m^{(k)})\|^2. \tag{56}$$

Since $L(f_M^{(k)})$ monotonically decreases for $k = 1, 2, ...\forall M$, and L is bounded below, $L(f_M^{(k)})$ converges. Summing up these inequalities (56) for $k = 1, ..., K$ and using (26) , we obtain

$$\frac{\gamma^2}{2\zeta}\sum_{k=1}^{K}\sum_{m=0}^{M-1}\|\nabla H_k(f_m^{(k)})\|^2 \leq L(f_M^{(0)}) - L(f_M^{(K)}) \leq L(f^{(0)}) - L(f^*), \tag{57}$$

where $f^* = \operatorname{argmin} L(f)$. Since L is bounded below, together with (57), the sum $\sum_{k=1}^{K}\sum_{m=0}^{M-1}\|\nabla H_k(f_m^{(k)})\|^2$ is absolutely convergent. So the terms can be rearranged in basically any reasonable way without altering the sum. Obviously as the sum converges, the individual terms must go to zero as $m \to \infty, k \to \infty$ in those various ways. As far as Algorithm 5 is concerned, we have

$$\|\nabla H_k(f_m^{(k)})\| \to 0 \text{ as } \begin{cases} k \to \infty, \forall m \\ m \to \infty, \forall k. \end{cases}$$

Furthermore, denote

$$\alpha_{M,K} = \min_{0\leq m\leq M-1, 1\leq k\leq K}\|\nabla H_k(f_m^{(k)})\|.$$

Then, using (57) we have

$$\alpha_{M,K} \leq \frac{1}{\sqrt{MK}}\left(\frac{2\zeta}{\gamma^2}\left(L(f^{(0)}) - L(f^*)\right)\right)^{1/2}.$$

Proof of Corollary 5.1: Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$, and $h_k(f)$ is given by (3). Assume that $\ell(f)$ is a ζ-strongly smooth functional, from Lemma 4, $H_k(f)$ is thus ζ-strongly smooth. Additionally, Algorithm 1 is a special case of Algorithm 5. Together with other assumptions, all conditions in Theorem 5 are satisfied. Thus the same conclusions of Theorem 5 are obtained and restated in the corollary.

Proof of Theorem 6: We analyze boosting iterations on minimizing the surrogate loss H_k. Denote $f_m^{(k)}$ the estimate at the m-th inner boosting iteration within the k-th outer MM loop, for $k = 1, 2, ..., m = 0, 1, 2, ...$ By the assumption that H_k is strongly smooth, we have

$$H_k(f_{m+1}^{(k)}) \leq H_k(f_m^{(k)}) + \langle \nabla H_k(f_m^{(k)}), f_{m+1}^{(k)} - f_m^{(k)} \rangle + \frac{\zeta}{2} \| f_{m+1}^{(k)} - f_m^{(k)} \|^2.$$

Let $f_{m+1}^{(k)} = f_m^{(k)} + w g_{m+1}, U_m^{(k)} = -\nabla H_k(f_m^{(k)})$. By the assumption that line 7 of Algorithm 5 returns a linear smoother g_{m+1} such that $g_{m+1} = S_{m+1}^{(k)} U_m^{(k)}$, and $S_{m+1}^{(k)}$ is symmetric idempotent, we have

$$\begin{aligned} H_k(f_{m+1}^{(k)}) &\leq H_k(f_m^{(k)}) - w U_m^{(k)\intercal} S_{m+1}^{(k)} U_m^{(k)} + \frac{w^2 \zeta}{2} \| S_{m+1}^{(k)} U_m^{(k)} \|^2 \\ &= H_k(f_m^{(k)}) - w U_m^{(k)\intercal} S_{m+1}^{(k)\intercal} S_{m+1}^{(k)} U_m^{(k)} + \frac{w^2 \zeta}{2} \| S_{m+1}^{(k)} U_m^{(k)} \|^2 \\ &= H_k(f_m^{(k)}) + (-w + \frac{w^2 \zeta}{2}) \| S_{m+1}^{(k)} U_m^{(k)} \|^2. \end{aligned}$$

Using the same technique for Theorem 2 and noting $\gamma \in (0, 1]$, the one step of the boosting decreases the value of surrogate loss at least as follows:

$$\begin{aligned} H_k(f_{m+1}^{(k)}) &\leq H_k(f_m^{(k)}) - \frac{1}{2\zeta} \| S_{m+1}^{(k)} U_m^{(k)} \|^2 \\ &\leq H_k(f_m^{(k)}) - \frac{\gamma^2}{2\zeta} \| S_{m+1}^{(k)} U_m^{(k)} \|^2. \end{aligned} \tag{58}$$

Summing up these inequalities (58) for $m = 0, ..., M-1$, we obtain

$$H_k(f_M^{(k)}) \leq H_k(f_0^{(k)}) - \frac{\gamma^2}{2\zeta} \sum_{m=0}^{M-1} \| S_{m+1}^{(k)} U_m^{(k)} \|^2.$$

With the same arguments as in the proof of Theorem 5, in particular, the construction of inequality (56), we have

$$L(f_M^{(k)}) \leq L(f_M^{(k-1)}) - \frac{\gamma^2}{2\zeta} \sum_{m=0}^{M-1} \| S_{m+1}^{(k)} U_m^{(k)} \|^2. \tag{59}$$

Since $L(f_M^{(k)})$ monotonically decreases for $k = 1, 2, ... \forall M$, and L is bounded below, $L(f_M^{(k)})$ converges. Summing up these inequalities (59) for $k = 1, ..., K$

and using (26) , we obtain

$$\frac{\gamma^2}{2\zeta}\sum_{k=1}^{K}\sum_{m=0}^{M-1}\|S_{m+1}^{(k)}U_m^{(k)}\|^2 \le L(f_M^{(0)}) - L(f_M^{(K)}) \le L(f^{(0)}) - L(f^*), \tag{60}$$

where $f^* = \operatorname{argmin} L(f)$. With (60) we have

$$\|S_{m+1}^{(k)}\nabla H_k(f_m^{(k)})\| = \|S_{m+1}^{(k)}U_m^{(k)}\| \to 0 \text{ as } \begin{cases} k\to\infty, \forall m \\ m\to\infty, \forall k. \end{cases}$$

Furthermore, denote

$$\theta_{M,K} = \min_{0\le m\le M, 1\le k\le K}\|S_{m+1}^{(k)}\nabla H_k(f_m^{(k)})\|.$$

Then using (60) we have

$$\theta_{M,K} \le \frac{1}{\sqrt{MK}}\left(\frac{2\zeta}{\gamma^2}\left(L(f^{(0)}) - L(f^*)\right)\right)^{1/2}.$$

Proof of Corollary 6.1: Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$, and $h_k(f)$ is given by (3). Assume that $\ell(f)$ is a ζ-strongly smooth functional, from Lemma 4, $H_k(f)$ is thus ζ-strongly smooth. Moreover, Algorithm 1 is a special case of Algorithm 5. Together with other assumptions, all conditions in Theorem 6 are satisfied. Thus the same conclusions of Theorem 6 are obtained and restated in the corollary.
Proof of Proposition 7: Suppose a functional ℓ satisfies Condition 1, i.e, $\forall\phi,\psi\in\Omega$, assume that

$$\ell(\phi) - \ell(\psi) \ge \langle S\nabla\ell(\psi), \phi-\psi\rangle + \frac{\eta}{2}\|\phi-\psi\|^2. \tag{61}$$

The right hand side of (61) is a convex quadratic function of ϕ (for fixed ψ). Setting the gradient with respect to ϕ equal to zero, we find that $\hat\phi = \psi - \frac{1}{\eta}S\nabla\ell(\psi)$ minimizes the right hand side of (61). Therefore we have

$$\begin{aligned}\ell(\phi) - \ell(\psi) &\ge \langle S\nabla\ell(\psi), \phi-\psi\rangle + \frac{\eta}{2}\|\phi-\psi\|^2 \\ &\ge \langle S\nabla\ell(\psi), \hat\phi-\psi\rangle + \frac{\eta}{2}\|\hat\phi-\psi\|^2 \\ &= -\frac{1}{2\eta}\|S\nabla\ell(\psi)\|^2 \\ &= -\frac{1}{2\eta}\nabla\ell(\psi)^\intercal S^\intercal S\nabla\ell(\psi) \\ &= -\frac{1}{2\eta}\nabla\ell(\psi)^\intercal S\nabla\ell(\psi).\end{aligned} \tag{62}$$

The lase equality holds due to that S is a symmetric idempotent matrix. Since (62) holds for any $\phi \in \Omega$, we have

$$\nabla \ell(\psi)^{\intercal} S \nabla \ell(\psi) \geq 2\eta \left(\ell(\psi) - \ell(\psi^*)\right),$$

where ψ^* is a minimization point of ℓ.

Proof of Theorem 7: Since the surrogate loss functional H_{k+1} is strongly smooth by assumption, we have:

$$H_{k+1}(f^{(k+1)}) \leq H_{k+1}(f^{(k)}) + \langle \nabla H_{k+1}(f^{(k)}), f^{(k+1)} - f^{(k)} \rangle + \frac{\zeta}{2} \|f^{(k+1)} - f^{(k)}\|^2. \tag{63}$$

We consider two cases:

(i) Let $f^{(k+1)} = f^{(k)} + w_{k+1} g_{k+1}$ with $w_{k+1} = \frac{\langle -\nabla H_{k+1}(f^k), g_{k+1} \rangle}{\zeta \|g_{k+1}\|^2}$. Substituting $f^{(k+1)}$ in (63), we have

$$H_{k+1}(f^{(k+1)}) \leq H_{k+1}(f^{(k)}) - \frac{1}{2\zeta} \frac{\langle \nabla H_{k+1}(f^{(k)}), g_{k+1} \rangle^2}{\|g_{k+1}\|^2}.$$

With the edge requirement $\gamma > 0$, we have

$$\langle -\nabla H_{k+1}(f^{(k)}), g_{k+1} \rangle \geq \gamma \|\nabla H_{k+1}(f^{(k)})\| \|g_{k+1}\|.$$

We then have

$$H_{k+1}(f^{(k+1)}) \leq H_{k+1}(f^{(k)}) - \frac{\gamma^2}{2\zeta} \|\nabla H_{k+1}(f^{(k)})\|^2.$$

Assume that functional H_{k+1} is η-strongly convex. Therefore applying (32) leads to

$$\|\nabla H_{k+1}(f^{(k)})\|^2 \geq 2\eta \left(H_{k+1}(f^{(k)}) - H_{k+1}(f_{\dagger}^{(k+1)}) \right),$$

where $f_{\dagger}^{(k+1)}$ is the minimum point of surrogate loss H_{k+1}. Therefore we have

$$H_{k+1}(f^{(k+1)}) \leq H_{k+1}(f^{(k)}) - \frac{\gamma^2 \eta}{\zeta} \left(H_{k+1}(f^{(k)}) - H_{k+1}(f_{\dagger}^{(k+1)}) \right). \tag{64}$$

(ii) Let $f^{(k+1)} = f^{(k)} + w g_{k+1}, U^{(k)} = -\nabla H_k(f^{(k)})$. By the assumption that line 5 of Algorithm 6 returns a linear smoother g_{k+1} such that $g_{k+1} = S_{k+1} U_k$, and S_{k+1} is symmetric idempotent, substituting $f^{(k+1)}$ in (63), we have

$$\begin{aligned}
H_k(f^{(k)}) &\leq H_k(f^{(k)}) - w U^{(k)\intercal} S^{(k)} U^{(k)} + \frac{w^2 \zeta}{2} \|S^{(k)} U^{(k)}\|^2 \\
&= H_k(f^{(k)}) - w U^{(k)\intercal} S^{(k)\intercal} S^{(k)} U^{(k)} + \frac{w^2 \zeta}{2} \|S^{(k)} U^{(k)}\|^2 \\
&= H_k(f^{(k)}) + (-w + \frac{w^2 \zeta}{2}) \|S^{(k)} U^{(k)}\|^2.
\end{aligned}$$

Using the same technique for Theorem 2 and noting $\gamma \in (0,1]$, the one step of the boosting decreases the value of surrogate loss at least as follows:

$$\begin{aligned} H_k(f^{(k+1)}) &\le H_k(f^{(k)}) - \frac{1}{2\zeta}\|S^{(k)}U^{(k)}\|^2 \\ &\le H_k(f^{(k)}) - \frac{\gamma^2}{2\zeta}\|S^{(k)}U^{(k)}\|^2. \end{aligned}$$

Assume that functional H_{k+1} satisfies Condition 1. Now using Proposition 7, we have

$$\|S^{(k)}U^{(k)}\|^2 \ge 2\eta\left(H_{k+1}(f^{(k)}) - H_{k+1}(f_\dagger^{(k+1)})\right).$$

Therefore again we have

$$H_{k+1}(f^{(k+1)}) \le H_{k+1}(f^{(k)}) - \frac{\gamma^2\eta}{\zeta}\left(H_{k+1}(f^{(k)}) - H_{k+1}(f_\dagger^{(k+1)})\right). \quad (65)$$

To summarize, inequalities (64) and (65) are the same despite the two different assumptions. Using (25), we know

$$L(f^{(k+1)}) \le H_{k+1}(f^{(k+1)}), \quad H_{k+1}(f^{(k)}) = L(f^{(k)}).$$

Combining with (65), hence the following holds:

$$L(f^{(k+1)}) \le L(f^{(k)}) - \frac{\gamma^2\eta}{\zeta}\left(L(f^{(k)}) - H_{k+1}(f_\dagger^{(k+1)})\right).$$

Subtracting the optimal value from both sides we get

$$\begin{aligned} L(f^{(k+1)}) - L(f^*) &\le L(f^{(k)}) - L(f^*) - \frac{\gamma^2\eta}{\zeta}\left(L(f^{(k)}) - H_{k+1}(f_\dagger^{(k+1)})\right) \\ &= L(f^{(k)}) - L(f^*) \\ &\quad - \frac{\gamma^2\eta}{\zeta}\left(L(f^{(k)}) - L(f^*) + L(f^*) - H_{k+1}(f_\dagger^{(k+1)})\right) \\ &= (1 - \frac{\gamma^2\eta}{\zeta})\left(L(f^{(k)}) - L(f^*)\right) \\ &\quad + \frac{\gamma^2\eta}{\zeta}\left(H_{k+1}(f_\dagger^{(k+1)}) - L(f^*)\right). \end{aligned} \quad (66)$$

Recursively applying the bound in (66) starting at $k = 0$ gives the bound on $L(f^{(K)}) - L(f^*)$ after K iterations:

$$\begin{aligned} L(f^{(K)}) - L(f^*) &\le (1 - \frac{\gamma^2\eta}{\zeta})^K\left(L(f^{(0)}) - L(f^*)\right) \\ &\quad + \frac{\gamma^2\eta}{\zeta}\sum_{k=1}^{K}(1 - \frac{\gamma^2\eta}{\zeta})^{K-k}\left(H_k(f_\dagger^{(k)}) - L(f^*)\right). \end{aligned} \quad (67)$$

Furthermore, assume that for $\beta > 0$ the following holds:

$$H_k(f_\dagger^{(k)}) - L(f^*) \leq \beta(1 - \frac{\gamma^2\eta}{\zeta})^k. \tag{68}$$

Plugging the inequality (68) into (67), we arrive the desired result (36).

Proof of Corollary 7.1: Suppose a loss functional $L(f) = \ell(f) + \ell_s(f)$ is majorized by a surrogate loss $H_k(f)$ at $f^{(k-1)}$, where $H_k(f) = \ell(f) + h_k(f)$, and $h_k(f)$ is given by (3). Assume that $\ell(f)$ is a ζ-strongly smooth and η-strongly convex functional, from Lemma 4, $H_k(f)$ is then ζ-strongly smooth and η-strongly convex. Together with other assumptions, all conditions in Theorem 7 are satisfied. Thus the same conclusions of Theorem 7 are obtained and restated in the corollary.

Acknowledgement

The author would like to thank the editor and two referees for their valuable suggestions and encouragement which have led to a much improved paper.

References

[1] Black, M. J. and Rangarajan, A. (1996). On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91.

[2] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press. MR2061575

[3] Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting (with discussion). *Statistical Science*, 22(4):477–505. MR2420454

[4] Bühlmann, P. and Hothorn, T. (2010). Twin boosting: improved feature selection and prediction. *Statistics and Computing*, 20:119–138. MR2610767

[5] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39(1):1–38. MR0501537

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

[7] Freund, Y. (2009). A more robust boosting algorithm. *arXiv preprint* https://arxiv.org/abs/0905.2138.

[8] Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37. MR2920188

[9] Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156. MR2920188

[10] Friedman, B., Henke, R. M., and Wier, L. M. (2010). Most expensive hospitalizations, 2008. *Agency for Health Care Policy and Research (US)*. http://www.hcup-us.ahrq.gov/reports/statbriefs/sb97.pdf.

[11] Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232. MR1873328

[12] Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics*, 28(2):337–407. MR1790002

[13] Grubb, A. and Bagnell, J. A. (2011). Generalized boosting algorithms for convex optimization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1209–1216. Omnipress.

[14] Krause, N. and Singer, Y. (2004). Leveraging the margin more carefully. In *Proceedings of the Twenty-first International Conference on Machine Learning*, page 63, Banff, Canada. ACM.

[15] Lange, K. (2013). *Optimization*. Springer, New York, second edition. MR3052733

[16] Li, A. H. and Bradic, J. (2015). Boosting in the presence of outliers: adaptive classification with non-convex loss functions. http://arxiv.org/pdf/1510.01064v1.pdf.

[17] Lin, Y. (2004). A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73–82. MR2064687

[18] Mason, L., Baxter, J., Bartlett, P., and Frean, M. (2000). Functional gradient techniques for combining hypotheses. In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 221–246, Cambridge, MA. MIT Press. MR1820960

[19] Mayr, A., Binder, H., Gefeller, O., and Schmid, M. (2014). The evolution of boosting algorithms: From machine learning to statistical modelling (together with the companion review and an invited discussion). *Methods of Information in Medicine*, 53(6):419–427.

[20] McDonald, R. A., Hand, D. J., and Eckley, I. A. (2004). A Multiclass Extension to the Brownboost Algorithm. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(05):905–931.

[21] McLachlan, G. and Krishnan, T. (2007). *The EM Algorithm and Extensions*, volume 382. John Wiley & Sons. MR2392878

[22] Moturu, S. T., Johnson, W. G., and Liu, H. (2007). Predicting future high-cost patients: a real-world risk modeling application. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 202–208. IEEE.

[23] Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media. MR2142598

[24] Park, S. Y. and Liu, Y. (2011). Robust penalized logistic regression with truncated loss functions. *Canadian Journal of Statistics*, 39(2):300–323. MR2839482

[25] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464. MR0468014

[26] Searle, S. R. (1982). *Matrix Algebra Useful for Statistics (Wiley Series in*

Probability and Statistics). Wiley-Interscience. MR0670947

[27] Shi, L., Campbell, G., Jones, W., Campagne, F., et al. (2010). The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature Biotechnology*, 28(8):827–838. https://goo.gl/8bdBDE.

[28] Sutherland, S. M., Ji, J., Sheikhi, F. H., Widen, E., Tian, L., Alexander, S. R., and Ling, X. B. (2013). AKI in hospitalized children: epidemiology and clinical associations in a national cohort. *Clinical Journal of the American Society of Nephrology*, 8(10):1661–1669.

[29] Tao, P. D. and An, L. T. H. (1997). Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355. MR1479751

[30] Vapnik, V. (1996). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York. MR1719582

[31] Wang, Z. (2011). HingeBoost: ROC-based boost for classification and variable selection. *The International Journal of Biostatistics*, 7(1):1–30. MR2775080

[32] Wang, Z. (2012). Multi-class HingeBoost: Method and application to the classification of cancer types using gene expression data. *Methods of Information in Medicine*, 51(2):162–167.

[33] Wassermann, L. (2006). *All of Nonparametric Statistics*. Springer Science & Business Media, New York. MR2172729

[34] Wu, Y. and Liu, Y. (2007a). On multicategory truncated-hinge-loss support vector. In *Prediction and Discovery: AMS-IMS-SIAM Joint Summer Research Conference, Machine and Statistical Learning: Prediction and Discovery, June 25–29, 2006, Snowbird, Utah*, volume 443, page 49. American Mathematical Soc.

[35] Wu, Y. and Liu, Y. (2007b). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983. MR2411659

[36] Yang, M., Xu, L., White, M., Schuurmans, D., and Yu, Y.-l. (2010). Relaxed clipping: A global training method for robust regression and classification. In *Advances in Neural Information Processing Systems*, pages 2532–2540.

[37] Zhang, X., Wu, Y., Wang, L., and Li, R. (2016). Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):53–76.