Asymmetric AdaBoost for High-dimensional Maximum Score Regression*

Jianghao Chu † Tae-Hwy Lee‡ Aman Ullah§

November 12, 2018

Abstract

Adaptive Boosting (AdaBoost) introduced by Freund and Schapire (1996) has gained enormous success in binary classification/prediction. In this paper, we introduce Asymmetric AdaBoost for solving high-dimensional binary classification/prediction problem with state-dependent loss functions. Asymmetric AdaBoost produces a nonparametric classifier via minimizing the "asymmetric exponential risk" which is a convex surrogate of the nonconvex score risk. The convex risk function gives huge computation advantage over nonconvex risk functions, e.g. Maximum Score (Manski, 1975, 1985), especially when the data is high-dimensional. The resulting nonparametric classifier is more robust than parametric classifiers whose performance depend on the correct specification of the model. We show that the risk of the classifier that Asymmetric AdaBoost produces approaches the Bayes risk which is the infimum risk can be achieved by all classifiers. Monte Carlo experiments show that Asymmetric AdaBoost performs better than the commonly used LASSO-regularized logistic regression when parametric assumption is violated and sample size is large. We apply the Asymmetric AdaBoost to predict the direction of changes in real personal income using data from McCracken and Ng (2016).

Key Words: AdaBoost, binary classification/prediction, convex relaxation, exponential risk.

JEL Classification: C25 C44 C53 C55

*We thank seminar participants at USC, UCR (Econ, CS), PKU, CAS, CUFE, CMES2018 (Shanghai), AMES2018 (Seoul), JSM2018 (Vancouver), CEC2018 (Irvine), and MEG2018 (Madison).

†Department of Economics, University of California, Riverside, CA 92521. E-mail: jianghao.chu@email.ucr.edu

‡Department of Economics, University of California, Riverside, CA 92521. E-mail: tae.lee@ucr.edu

§Department of Economics, University of California, Riverside, CA 92521. E-mail: aman.ullah@ucr.edu

1 Introduction

Many important variables in economics as well as other areas are binary, e.g. whether the economy is in recession or expansion, the stock market is going up or going down, and a mortgage application should be approved or denied.

Let $y \in \{1, -1\}$ be a binary variable, e.g. $y = 1$ if the economy is in expansion and $y = -1$ if the economy is in recession. And let $G(x)$ be a classifier of y. This paper investigates the problem of classification/prediction that minimizes a weighted (asymmetric) misclassification probability

$$\begin{aligned} R_\tau(G) &= \mathbb{E}\left[\tau(x) \times 1_{(y=-1,G(x)=1)} + (1-\tau(x)) \times 1_{(y=1,G(x)=-1)}\right] && (1) \\ &= \mathbb{E}_x[\tau(x) \Pr(y=-1, G(x)=1|x) + (1-\tau(x)) \Pr(y=1, G(x)=-1|x)], && (2) \end{aligned}$$

where the first expectation is taken over y and x, and the symbol $1_{(\cdot)}$ is the indicator function which takes the value 1 if the logical conditions inside the parenthesis are satisfied and takes the value 0 otherwise. The utility-based weight function $\tau(x)$ assigns different penalties conditioning on the state variable y and characteristics x as shown in Section 3.1. In addition, we allow the characteristics x to be high-dimensional, and both the conditional distribution of y given x and the functional form of the classifier $G(x)$ to be of unknown forms.

The problem of binary classification/prediction have been investigated extensively in the past. However, the existing studies varies to a great extent by their focuses, assumptions and data. The focus on binary classification/prediction in most disciplines is often to minimize the unweighted (symmetric) misclassification probability

$$R(G) = \mathbb{E}\left[\frac{1}{2} \times 1_{(y \neq G(x))}\right] = \frac{1}{2} \Pr(y \neq G(x)), \quad (3)$$

where $\tau(x) = \frac{1}{2}$ for all x. Hence, wrong predictions are given the same penalty regardless of whether it is a false positive (FP, type I error) or false negative (FN, type II error) classification/prediction. However, economic theory suggests that the optimal prediction is related with decision theory which aims at maximizing the utility of economic agents. Economists find that people prefer to avoid costly mistakes at the price of making more cost-less ones. For example, people are more willingly to arrive at the airport early than late (Granger, 1999). People are more willingly to overestimate the peak water of a dam than to underestimate it (Zellner, 1986). Such incentives promote the use of an state-dependent (asymmetric) loss in the estimation process of the optimal classifier which takes into account the economic agent's utility function and gives

higher penalty on costly mistakes and lower penalty on cost-less ones. Elliott and Lieli (2013) propose a utility based classifier called maximum utility estimator by using the maximum score approach of Manski (1985) but include utility which lead to a state-dependent (asymmetric) loss function.

In terms of assumptions, Lahiri and Yang (2012) categorize the methods of binary classification/prediction into probability estimation and point estimation. Probability estimation assumes the true class of probability distribution is known to the researcher, such as logit and probit models (Gaddum, 1933; Bliss, 1934a,b). However, such information is seldom available in practice. Klein and Spady (1993) relax the assumption on the true distribution by using a semiparametric linear single-index kernel regression. Diaconis and Freedman (1993) propose Bayesian approach for nonparametric binary regression. Point estimation, on the other hand, does not require knowledge on the true probability distribution. Manski (1975, 1985) use the maximum score approach by maximizing a score function similar to (1). However, all of the above mentioned methods still assumes that the optimal classifier is a function of the linear combination of x. Freund and Schapire (1997) introduce the AdaBoost which iteratively combines weak classifiers into a strong classifier that does not assume the optimal classifier to be a function of the linear combination of x. Other methods include regression tree, K-nearest neighbor and deep neural network which relax the linear assumption (Breiman, 1984; Altman, 1992).

Given the availability of high-dimensional data, modern econometric methods differs from traditional ones in the sense that they try to incorporate high-dimensional data for the use of binary classification/prediction. Tibshirani (1996) introduce regularized logistic regression which adds an L_1 penalty on the coefficients of independent covariates and solve the problem of excessive independent covariates by shrinking the coefficients of unimportant covariates to zero. Freund and Schapire (1997) introduce the AdaBoost which takes a functional descent procedure and selects the independent variables sequentially instead of all at once.

This paper takes the modern approach of econometrics that does not impose any assumption on the conditional probability of y given x and the functional form of the optimal classifier. We propose the Asymmetric AdaBoost which minimizes our asymmetric exponential loss via functional gradient descent and builds a strong (optimal) classifier by iteratively combining weak classifiers. The resulted strong classifier can encamps a large class of functions even if the weak classifiers are restricted to follow a given parametric form. Moreover, we use component-wise algorithm and select only one independent variable at a time to solve the issue of high-dimensionality.

The rest of the paper is organized as follows. In Section 2, we provide a brief introduction of the symmetric binary classification/prediction problem and two methods, namely maximum score and AdaBoost, for solving the problem. In Section 3, we look into the problem of prediction with state-dependent losses and introduce a new "asymmetric exponential risk" function based on the utility functions. We also propose a new algorithm that minimizes the "asymmetric exponential risk" and builds up a nonparametric classifier. In Section 4, we examine the finite sample properties of Asymmetric AdaBoost via Monte Carlo simulations. Section 5 predicts the direction of changes in Real Personal Income. Section 6 concludes. All technical derivations and proofs are presented in the Appendix.

2 Exponential Loss

In this section, we introduce the symmetric binary prediction problem, i.e. $t(y,x) = \frac{1}{2}$, and the maximum score approach (Manski, 1975) for solving the problem. We then introduce the exponential loss which is a convex surrogate of the score loss and the AdaBoost algorithm that minimizes the exponential loss via Newton-like updates (Friedman et al., 2000).

2.1 Binary Classification and Maximum Score

Given a binary variable $y \in \{1,-1\}$ and covariates $x \in \chi$. The random variables $(x,y) \sim \mathcal{P}$, where $\mathcal{P}$ is an unknown distribution on $\chi \times \{1,-1\}$. Our goal is to construct a classifier $G : x \to y$ such that it has the smallest symmetric misclassification probability (3).

Manski (1975) proposes to obtain the classifier by

$$G(x) = \arg\max_{G} \mathbb{E}\left[yG(x)\right], \tag{4}$$

which is called the maximum score approach. We maximize (4) with respect to $G(x) \in \{1,-1\}$,

$$\max_{G} \mathbb{E}\left[yG(x)|x\right] = \left[\Pr(y=1|x) - \Pr(y=-1|x)\right] G(x). \tag{5}$$

Hence, $G(x)$ takes the same sign as $\Pr(y=1|x) - \Pr(y=-1|x)$ when (4) is maximized, i.e.

$$G^*(x) = \begin{cases} 1 & \Pr(y=1|x) > \Pr(y=-1|x) \\ -1 & otherwise, \end{cases} \tag{6}$$

or equivalently,

$$G^*(x) = \begin{cases} 1 & \Pr(y=1|x) > 0.5 \\ -1 & otherwise. \end{cases} \tag{7}$$

Remark 1. We refer to the problem and the risk functions in Section 2 as "symmetric" since the optimal decision rule is $\Pr(y=1|x)>0.5$, i.e. the optimal classifier uses 0.5 as the threshold.

Note that the risk function (4) is a linear transformation of the symmetric misclassification probability (3),

$$\mathbb{E}[-yG(x)] = 4\times\mathbb{E}\left[\frac{1}{2}\times 1_{(G(x)\neq y)}\right]-1 = 4\times R(G)-1. \tag{8}$$

Hence, the maximum score approach is equivalent to minimizing the symmetric misclassification probability.

From (7), the optimal maximum score classifier, also known as the Bayes classifier, makes classification based on the condition $\Pr(y=1|x)>0.5$. The Bayes classifier achieves the Bayes risk

$$R^* = \inf_G R(G) = \mathbb{E}\min\left\{\frac{1}{2}\Pr\left(y=1|x\right), \frac{1}{2}\Pr\left(y=-1|x\right)\right\}, \tag{9}$$

where the infimum is taken over all possible (measurable) classifiers.

The maximum score approach yields a classifier that minimizes the symmetric misclassification probability. It is superior to many other popular methods, e.g. probit and logit models, in the sense that it does not have to assume that y given x follows a given distribution. However, there are some limitations: The classifier is assumed to take the form $G(x)=\text{sign}[x'\beta]$, i.e. the optimal classifier is the sign of a linear function; The objective function used is nonconvex which lead to computation difficulty especially when the sample size is large; The method does not work if covariates are high-dimensional.

2.2 Convex Surrogate

The objective function (4) is hard to optimize. As a solution to the computation difficulty, Zhang (2004), Bartlett et al. (2006), Chandrasekaran and Jordan (2012) and Friedman et al. (2000) propose to use "convex surrogates" of the nonconvex loss functions for binary classification/prediction.

Since $G(x)\in\{1,-1\}$, it is standard to assume that $G(x)$ is the sign of a real-valued function,[1] i.e.

$$G(x) = \text{sign}[F(x)], \tag{10}$$

where $F(x)\in\mathbb{R}$ and

$$\text{sign}\left[z\right] = \begin{cases} 1 & z>0 \\ -1 & otherwise. \end{cases} \tag{11}$$

[1]Manski (1975) assumes $G(x)$ is the sign of a linear function of x, i.e. $G(x)=\text{sign}[x'\beta]$.

Let $\mathcal{P}(x, y)$ be the joint probability density function of (x, y) on $\chi \times \{\pm 1\}$. For the simplicity of notation, we abuse the notation to define the score risk of the classifier (10) as

$$R(F) = \mathbb{E}\left(\frac{1}{2} \times \frac{1 - y\,\mathrm{sign}[F(x)]}{2}\right) = \mathbb{E}\left(\frac{1}{2} \times 1_{(-yF(x)>0)}\right), \tag{12}$$

which is the same as (3) except that the risk function takes argument F instead of G. We use the same notation since that

$$\mathbb{E}\left(\frac{1}{2} \times 1_{(-yF(x)>0)}\right) = \mathbb{E}\left(\frac{1}{2} \times 1_{(y\neq G(x))}\right), \tag{13}$$

and

$$R^* = \inf_F R(F) = \inf_G R(G) \tag{14}$$

for $G(x) = \mathrm{sign}[F(x)]$ as in (10). The score risk (12) is a linear transformation of (4) and is equivalent to the symmetric misclassification probability (3). Hence, it is not convex and will lead to huge computational difficulty. To overcome the computation issue of (12), we can use a convex surrogate which we call the exponential risk instead,

$$R_\psi(F) = \mathbb{E}\left(\frac{1}{2} e^{-yF(x)}\right), \tag{15}$$

where $\psi(z) = e^z$. The subscript ψ indicates that the exponential risk (15) replaces the nonconvex indicator function in the score risk (12) with the convex exponential function. As shown in Figure 1, the exponential risk (15) is a convex upper bound of the score risk (12) with $R(F) = R_\psi(F)$ only at $F = 0$. Let us denote the optimal exponential risk as

$$R^*_\psi = \inf_F R_\psi(F). \tag{16}$$

Remark 2. The optimal classifier from minimizing the exponential risk also uses $\Pr(y = 1|x) > 0.5$ for the classification rule as in (7). Let

$$\begin{aligned} R_\psi(F(x)) &= E\left[E\left(\frac{1}{2} e^{-yF(x)}|x\right)\right] \\ &= E\left[\frac{1}{2}\Pr(y = 1|x)\, e^{-F(x)} + \frac{1}{2}\Pr(y = -1|x)\, e^{F(x)}\right]. \end{aligned} \tag{17}$$

Taking derivative w.r.t. $F(x)$ and making it equal to zero, we obtain

$$\frac{\partial E\left(e^{-yF(x)}|x\right)}{\partial F(x)} = -\frac{1}{2}\Pr(y = 1|x)\, e^{-F(x)} + \frac{1}{2}\Pr(y = -1|x)\, e^{F(x)} = 0. \tag{18}$$

Hence,

$$F^*(x) = \frac{1}{2}\log\left[\frac{\Pr(y = 1|x)}{\Pr(y = -1|x)}\right]. \tag{19}$$

Moreover, the optimal classifier,

$$G^*(x) = \text{sign}\left[F^*(x)\right] = \begin{cases} 1 & \Pr(y=1|x) > 0.5 \\ -1 & otherwise, \end{cases} \tag{20}$$

follows the classification rule $\Pr(y=1|x) > 0.5$.

Theorem 1 (Bartlett et al., 2006)**.** *For every sequence of measurable functions $F_m : \chi \to \mathbb{R}$ and every probability distribution on $\chi \times \{\pm 1\}$,*

$$R_\psi\left(F_m\right) \to R_\psi^* \quad \textit{implies that} \quad R\left(F_m\right) \to R^*.$$

Proof. This is a special case of Theorem 1 of Bartlett et al. (2006) for the exponential risk. □

Theorem 1 shows in binary classification/prediction, the classifier that minimizes the exponential risk would automatically minimizes the score risk. Hence, the exponential risk can be used in the place of the score risk as the objective function for binary classification/prediction. The use of the convex exponential risk will provide not only dramatic computation improvement but also flexibility in algorithm as we shall see in the next section.

2.3 AdaBoost

In the machine learning literature, the AdaBoost uses the exponential risk for binary classification/prediction. This section introduces the AdaBoost algorithm. Let n be the number of observations and $f_{mj}\left(x_j\right)$ be the weak binary classifier fitted using y and x_j in the mth iteration. The algorithm of the Component-wise Discrete AdaBoost is shown in Algorithm 1.

Algorithm 1 Component-wise AdaBoost

1. Start with weights $w_i = \frac{1}{n}, i = 1, \ldots, n$.
2. For $m = 1$ to M
   (a) For $j = 1$ to k (for each variable)
      i. Fit the classifier $f_{mj}(x_{ij}) \in \{-1, 1\}$ using weights w_i on the training data.
      ii. Compute $err_{mj} = \sum_{i=1}^{n} w_i 1_{(y_i \neq f_{mj}(x_{ji}))}$.
   (b) Find $\hat{j}_m = \arg\min_j err_{mj}$
   (c) Compute $c_m = \log\left(\frac{1-err_{m,\hat{j}_m}}{err_{m,\hat{j}_m}}\right)$.
   (d) Set $w_i \leftarrow w_i \exp[c_m 1_{(y_i \neq f_{m,\hat{j}_m}(x_{\hat{j}_m,i}))}]$, $i = 1, \ldots, n$, and normalize so that $\sum_{i=1}^{n} w_i = 1$.
3. Output the classifier $\text{sign}[F_M\left(x\right)]$ where $F_M\left(x\right) = \sum_{m=1}^{M} c_m f_{m,\hat{j}_m}(x_{\hat{j}_m})$.

Remark 3. Algorithm 1 uses only one explanatory variable at a time to fit a weak classifier $f_{mj}(x_j)$. In the end, Algorithm 1 produces a strong classifier $F_M(x)$ by combining all the weak classifiers that uses different explanatory variables. Hence, Algorithm 1 overcomes the high-dimensional data problem by selecting only one explanatory variable in each iteration and combining the classifiers across iterations. Moreover, the resulted strong classifier is a weighted sum of all weak classifiers which is not required to take any standard functional form.

From the use of a convex risk function, Algorithm 1 is computationally more efficient. Moreover, since the convex exponential risk (15) is differentiable, Algorithm 1 uses functional gradient descent to minimize the exponential risk which will produce a classifier with larger flexibility.

Theorem 2 (Friedman et al., 2000)**.** *Algorithm 1 builds an additive regression model $F_M(x)$ via minimizing the exponential risk (16).*

Proof. See Appendix 7.1. □

Theorem 2 shows Algorithm 1 as minimizing the exponential risk via functional gradient descent. Algorithm 1 is more efficient since it minimizes the convex exponential risk (16) which is better performed than the nonconvex score risk (12). Moreover, the convexity of the exponential loss allows for the use of functional gradient descent which provides more flexibility in the functional form of the resulted classifier.

Theorem 3 (Bartlett and Traskin, 2007)**.** *Algorithm 1 stopped at step $M_n = n^{1-\epsilon}$ returns a sequence of classifiers F_{M_n} almost surely satisfying*

$$R_\psi(F_{M_n}) \to R_\psi^* \quad as \quad n \to \infty. \tag{21}$$

Proof. This is Theorem 1 of Bartlett and Traskin (2007). □

Theorem 3 shows that Algorithm 1 is consistent in the sense that the risk of the classifier produced converges to the optimal exponential risk (16). The classifier produced by Algorithm 1 will minimize the exponential risk (16). Moreover, by Theorem 1, the classifier will also achieve the Bayes risk (9).

Theorem 4. *Given the conditions in Theorem 3, Algorithm 1 stopped at iteration $M_n = n^{1-\epsilon}$ where $\epsilon \in (0, 1)$ returns a sequence of classifiers F_{M_n} almost surely satisfying*

$$R\left(F_{M_n}\right) \to R^* \text{ as } n \to \infty. \tag{22}$$

Proof. Combining Theroems 1 and 3 gives the above result. □

Algorithm 1 solves the three aforementioned problems of the maximum score approach. The obtained classifier always takes $\Pr(y=1|x) > 0.5$ as the classification rule. However, econometric methods such as the maximum score approach is able to take into account the state-dependent utilities of economic agent which may require a classification/prediction rule other than $\Pr(y=1|x) > 0.5$. In the next section, we generalize the exponential risk to deal with such problems.

3 Asymmetric Exponential Loss

In this section, we introduce a new risk function, namely the asymmetric exponential risk, for solving binary classification/prediction under state-dependent losses. The state-dependent risk function is a weighted version of the score risk (12) as follows:

$$R_\tau(F) = \mathbb{E}\left(t(y,x)\,1_{(-y\,\mathrm{sign}[F(x)]>0)}\right) = \int t(y,x)\,1_{(-y\,\mathrm{sign}[F(x)]>0)}\mathcal{P}(x,y)\,\mathrm{d}\,y\,\mathrm{d}\,x, \tag{23}$$

where

$$t(y,x) = \begin{cases} \tau(x) & y = 1 \\ 1-\tau(x) & y = -1. \end{cases} \tag{24}$$

is a non-negative function of outcome variable y and characteristics x. The score risk (23) is the counterpart of (1) with argument $F \in \mathbb{R}$ instead of $G \in \{1,-1\}$. Similarly, let

$$R_\tau^* = \inf_F R_\tau(F) = \mathbb{E}\{\min[t(1,x)\Pr(y=1|x), t(-1,x)\Pr(y=-1|x)]\} \tag{25}$$

be the Bayes risk.

We also propose a new algorithm, that we call the Asymmetric AdaBoost, which produces a nonparametric classifier by minimizing the asymmetric exponential risk. Our new algorithm is computationally efficient and is able to handle binary classification/prediction problem with high-dimensional covariates.

3.1 Binary Classification with State-dependent Utilities

Granger and Pesaran (2000) discuss the idea of using decision theory to evaluate classification/prediction accuracy in a two-state two-action decision problem. Assume the payoff matrix is

	$y=1$	$y=-1$
$G(x)=1$	$u_{1,1}(x)$	$u_{1,-1}(x)$
$G(x)=-1$	$u_{-1,1}(x)$	$u_{-1,-1}(x)$

(26)

where $u_{i,j}(x)$ is the state dependent utility of making prediction i when the realized value is j under circumstances x. Without loss of generality, we assume that $u_{1,1}(x) - u_{-1,1}(x) + u_{-1,-1}(x) - u_{1,-1}(x) = 1$. It is natural to also assume that all utilities are bounded and taking the correct decision i corresponding to realized state j is beneficial: $\tau(x) \equiv u_{1,1}(x) - u_{-1,1}(x) > 0$ and $1 - \tau(x) \equiv u_{-1,-1}(x) - u_{1,-1}(x) > 0$.

The expected utility of $G(x) = 1$ is

$$\Pr(y = 1|x)\, u_{1,1}(x) + \Pr(y = -1|x)\, u_{1,-1}(x). \tag{27}$$

The expected utility of $G(x) = -1$ is

$$\Pr(y = 1|x)\, u_{-1,1}(x) + \Pr(y = -1|x)\, u_{-1,-1}(x). \tag{28}$$

$G(x) = 1$ gives a higher utility if

$$\Pr(y = 1|x)\, u_{1,1}(x) + \Pr(y = -1|x)\, u_{1,-1}(x) > \Pr(y = 1|x)\, u_{-1,1}(x) + \Pr(y = -1|x)\, u_{-1,-1}(x). \tag{29}$$

Hence,

$$\Pr(y = 1|x) > u_{-1,-1}(x) - u_{1,-1}(x) = 1 - \tau(x), \tag{30}$$

is the sufficient condition for $G(x) = 1$ to be the optimal choice.

The optimal decision rule depends only on $\tau(x) = u_{1,1}(x) - u_{-1,1}(x)$ and $1 - \tau(x) = u_{-1,-1}(x) - u_{1,-1}(x)$. Hence, without loss of generality, we can construct a problem with the same optimal classifier with $u'_{1,1}(x) = 0$, $u'_{-1,1}(x) = -\tau(x)$, $u'_{-1,-1}(x) = 0$ and $u'_{1,-1}(x) = -(1 - \tau(x))$. Since the loss can be seen as the negative payoff, the constructed problem have a loss matrix as follows:

	$y = 1$	$y = -1$
$G(x) = 1$	0	$1 - \tau(x)$
$G(x) = -1$	$\tau(x)$	0

(31)

where the risk can be summarized as (23). However, by the same arguments of the previous section, the score risk (23) is nonconvex. Hence, better methods may be proposed using convex surrogates.

In this general case, the optimal classification rule is no longer $\Pr(y = 1|x) > 0.5$. The optimal classifier (also known as the Bayes classifier)

$$G^*_\tau(x) = \begin{cases} 1 & \Pr(y = 1|x) > 1 - \tau(x) \\ -1 & otherwise. \end{cases} \tag{32}$$

uses the classification rule (29) that is a function of the state-dependent utilities of the economic agent and achieves the Bayes risk (24).[2]

Remark 4. We refer to the binary classification/prediction problem with state-dependent losses as asymmetric since the optimal classification rule is $\Pr(y=1|x) > 1-\tau(x)$, i.e. the threshold is $1-\tau(x)$ instead of 0.5 as in the symmetric case. By the same argument, we name our proposed risk function in the next section as "asymmetric" exponential risk and refer to the exponential risk function in Section 2 as "symmetric" exponential risk.

3.2 Convex Surrogate

The score risk (23) is nonconvex which lead to high computation cost especially when the sample size is large and/or covariates are high-dimensional. To solve the problem, we propose to use a new risk function, the asymmetric exponential risk,

$$R_{\psi,\tau}(F) = \mathbb{E}\left(t(y,x)e^{-yF(x)}\right) = \int t(y,x)\, e^{-yF}\mathcal{P}(x,y)\,\mathrm{d}\,y\,\mathrm{d}\,x, \tag{34}$$

which is a convex surrogate of the score risk (23). Similarly, let us denote the optimal asymmetric exponential risk as

$$R^*_{\psi,\tau} = \inf_F R_{\psi,\tau}(F). \tag{35}$$

Similar to the previous sections, the asymmetric exponential risk replaces the nonconvex indicator function in the score risk (23) with the convex exponential function. As shown in Figure 1, the asymmetric exponential risk (33) is a convex upper bound of the score risk (23).

Note that the optimal classifier from minimizing the asymmetric exponential risk (33) also uses $\Pr(y=1|x) > 1-\tau(x)$ for the classification rule as in (31). Take the derivative of

$$\begin{aligned} R_{\psi,\tau}(F(x)) &= \mathbb{E}\left[\mathbb{E}\left(t(y,x)e^{-yF(x)}|x\right)\right] \\ &= \mathbb{E}\left[\tau(x)\Pr(y=1|x)\, e^{-F(x)} + (1-\tau(x))\Pr(y=-1|x)\, e^{F(x)}\right]. \end{aligned}$$

w.r.t. $F(x)$ and making it equal to zero, we obtain

$$\frac{\partial E\left(e^{-yF(x)}|x\right)}{\partial F(x)} = -\tau(x)\Pr(y=1|x)\, e^{-F(x)} + (1-\tau(x))\Pr(y=-1|x)\, e^{F(x)} = 0. \tag{36}$$

[2]Manski (1975, 1985) propose the maximum score estimator to solve the above binary classification problem from minimizing a linear transformation of the score risk

$$\max_G E(t(y,x)yG(x)). \tag{33}$$

Elliott and Lieli (2013) also use a similar estimator which they call the maximum utility estimator.

Hence,

$$F_\tau^*(x) = \frac{1}{2}\log\left[\frac{\tau(x)\Pr(y=1|x)}{(1-\tau(x))\Pr(y=-1|x)}\right]. \tag{37}$$

Moreover, the optimal classifier,

$$G_\tau^*(x) = \text{sign}[F^*(x)] = \begin{cases} 1 & \Pr(y=1|x) > 1-\tau(x) \\ -1 & otherwise, \end{cases} \tag{38}$$

follows the classification rule $\Pr(y=1|x) > 1-\tau(x)$.

Theorem 5. *For every sequence of measurable functions $F_m : x \to \mathbb{R}$ and every probability distribution on $x \times \{\pm 1\}$,*

$$R_{\psi,\tau}(F_m) \to R_{\psi,\tau}^* \text{ implies that } R_\tau(F_m) \to R_\tau^*. \tag{39}$$

Proof. See Appendix 7.2. □

Theorem 5 establishes the relationship between the convex asymmetric exponential risk and the nonconvex score risk that is widely used in decision theories such as the two-state two-action decision problem mentioned before. Therefore, we are able to replace the nonconvex risk function with a convex surrogate which could be minimized more efficiently and provide enormous improvement with large samples and high-dimensional data.

3.3 Asymmetric AdaBoost

In this section, we introduce our algorithm, which we call the Asymmetric AdaBoost, for minimizing our asymmetric exponential risk. We use functional gradient descent to produce a nonparametric classifier. In addition, our algorithm can handle high-dimensional covariates. The algorithm is shown in Algorithm 2.

Algorithm 2 Component-wise Asymmetric AdaBoost

1. Start with weights $w_i = t(y_i, x_i)$, $i = 1, \ldots, n$, and normalize so that $\sum_{i=1}^N w_i = 1$.
2. Steps 2 and 3 are the same as in Algorithm 1.

Remark 5. For the selection of the number of iterations M, a widely used method in the boosting literature is cross-validation. Here we can divide the whole sample into several sections, then take turns to use one section as test sample to evaluate the obtained model while using the other sections as training sample. In the end, we choose the number of iteration that has the least cross-validation loss. Another choice is to use

information criterion, e.g. AICc. The exponential loss can be linked with log-likelihood of logistic models as in Ng (2014).

The Component-wise Asymmetric AdaBoost algorithm uses one explanatory variable at a time to fit a weak classifier $f_{mj}(x_j)$. In the end, the algorithm produces a strong classifier $F_M(x)$ by combining all the weak classifiers that uses different explanatory variables. Hence, the Component-wise Asymmetric AdaBoost overcomes the high-dimensional data problem by selecting only one explanatory variable in each iteration and combining the weak classifiers across iterations. Moreover, the resulted strong classifier is a weighted sum of weak classifiers which is not required to satisfy any parametric assumption. To better understand the new algorithm, we follow the steps of Friedman et al. (2000) to explain our Asymmetric AdaBoost.

Theorem 6. *Algorithm 2 builds an additive regression model $F_M(x)$ via Newton-like updates for minimizing the asymmetric exponential risk (33).*

Proof. See Appendix 7.3. □

As in the previous sections, from the use of a convex risk function, Algorithm 2 is computationally more efficient. Moreover, since the convex exponential risk (15) is differentiable, Algorithm 2 uses functional gradient descent to minimize the asymmetric exponential risk which will produce a classifier with larger flexibility.

Theorem 7. *Let assumption 1 be satisfied. Then the Algorithm 2 stopped at iteration $M_n = n^{1-\epsilon}$ where $\epsilon \in (0,1)$ returns a sequence of classifiers F_{M_n} almost surely satisfying*

$$R_{\psi,\tau}\left(F_{M_n}\right) \to R^*_{\psi,\tau} \text{ as } n \to \infty. \tag{40}$$

Proof. See Appendix 7.4. □

Theorem 7 shows that Algorithm 2 is consistent in the sense that the risk of the classifier obtained will converge to the optimal asymmetric exponential risk as the sample size goes to infinity, i.e. the classifier produced by Algorithm 2 will minimize the exponential risk (16). Moreover, by Theorem 5, the classifier will also achieve the Bayes risk (24).

Theorem 8. *Given the conditions in Theorem 7, the Algorithm 2 stopped at iteration $M_n = n^{1-\epsilon}$ where $\epsilon \in (0,1)$ returns a sequence of classifiers F_{M_n} almost surely satisfying*

$$R_{\tau}\left(F_{M_n}\right) \to R^*_{\tau} \text{ as } n \to \infty. \tag{41}$$

Proof. Combining Theorems 5 and 7 gives the above result. □

Algorithm 2 encamps Algorithm 1 by letting $t(y,x) = \frac{1}{2}$. Hence, Algorithm 2 is able to solve binary classification/prediction problem with state-dependent losses while maintaining the computation advantage and function form flexibility of Algorithm 1. In addition, Algorithm 2 can deal with high-dimensional x.

4 Monte Carlo

In this section, we examine the finite sample properties of the Asymmetric AdaBoost via Monte Carlo simulations. We consider the binary decision problem in Section 3.1 with $\tau(x) = \tau$.

4.1 DGPs

We construct the following DGPs where y follows Bernoulli distribution.

DGP1&2 (Logistic Model):

$$\Pr(y=1|x) = \frac{1}{1+e^{-v}}.$$

$$v = \begin{cases} \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p & \text{DGP1 (linear model)} \\ \beta_2 x_1^2 - \beta_2 x_2^2 + \beta_3 x_3 + \cdots + \beta_p x_p & \text{DGP2 (quadratic model)} \end{cases}$$

where

$$(x_1, x_2, \ldots, x_p)' \sim N(0, I_p), \quad \beta_j = 0.8^j, \ j = 1, \ldots, p$$

$$n = \{1000, 100\}, \quad p = 100$$

DGP3 (Circle Model):

$$\Pr(y=1|x) = \begin{cases} 1 & v < 8 \\ \frac{28-v}{20} & 8 \le v \le 28 \\ 0 & v > 28 \end{cases}.$$

Let x be a $p \times 1$ vector.

$$v = \sqrt{x_1^2 + x_2^2}$$

where

$$x_j \sim \mathrm{U}[-28, 28], \quad j = 1, \ldots, p$$

$$n = \{1000, 100\}, \quad p = 100$$

The probability, $\Pr(y=1|x)$, in the DGP3 is shown in Figure 2. A major difference between DGP3 and the other DGPs is that $\Pr(y=1) < 0.5$ in DGP3. Hence, the data is unbalanced, i.e. there are more events of

$y = -1$ than $y = 1$. We have this setup since in many situations we are more interested in predicting an event that is less common than its complementary, e.g. recession over expansion.

To construct the training and testing samples, we randomly generate x using the above distribution and calculate $\Pr(y = 1|x)$. To generate the random variable y based on x, we first generate a random variable ϵ that follows uniform distribution between $[0, 1]$. Next, we compare ϵ with $\Pr(y = 1|x)$. There is a probability of $\Pr(y = 1|x)$ that ϵ is smaller than $\Pr(y = 1|x)$ and a probability $1 - \Pr(y = 1|x)$ otherwise. Hence, we set

$$y = \begin{cases} 1 & \epsilon < \Pr(y = 1|x) \\ -1 & \epsilon > \Pr(y = 1|x). \end{cases} \tag{42}$$

To evaluate the algorithms, first we train our classifier with the training data of size $n = \{100, 1000\}$. Then, we use a testing dataset that contains $n' = 10000$ new observations to test the out-of-sample performance of the methods.

We report the following sample version of score risk for the tested methods,

$$R_{\tau,n'}(F) = \tau \sum_{y_i=1} 1_{(y_i \neq \text{sign}[F(x_i)])} + (1-\tau) \sum_{y_i=-1} 1_{(y_i \neq \text{sign}[F(x_i)])}.$$

We also report the sample Bayes risk as the benchmark for comparison,

$$R^*_{\tau,n'} = \sum_{i=1}^{n'} \min\{\tau \Pr(y = 1|x_i), (1-\tau) \Pr(y = -1|x_i)\}.$$

4.2 Comparing with Asymmetric Logistic Regression

Apart from Asymmetric AdaBoost, we consider logistic regression as an alternative method to obtain a classifier of y. In the alternative method, we use $Y = \frac{y+1}{2}$ for simplification. Because of the high-dimensional construction of our problem, we minimizes the negative logistic log-likelihood with a LASSO-penalty as below

$$\beta = \arg\min_{\beta} -\sum_{i=1}^{n} \left[Y_i (x_i\beta) - \log\left(1 + e^{x_i\beta}\right)\right] + \lambda |\beta|_1. \tag{43}$$

We use the *glmnet* package provided by Hastie and Qian for this alternative method. We use the estimated β to construct a logistic probability model for y. Then, get the classifications by plugging the estimated logistic probability into the Bayes classifier (31).

4.3 Results

The simulation results are listed in the tables. We report in total 3 tables.

In table 1, the DGP1 is a linear logistic model. Logistic regression has advantage over Asymmetric AdaBoost both when n is small and large. This is expected since logistic regression has the correct parametric assumption in this case. Moreover, we see that as the sample size increases, the loss of the Asymmetric AdaBoost converges to the sample Bayes risk.

In table 2, the DGP2 is still the logit model. However, the single index, v, in the logistic function is quadratic in x_1 and x_2. In this case, the logistic regression is biased since it assumes that the single index is a linear function of the covariates. When n is small, we see that the results are neck and neck. The Asymmetric AdaBoost works better when τ is close to 0.5 and the logistic regression works better when τ is away from 0.5. Both methods are far behind the Bayes risk since the Asymmetric AdaBoost without parametric assumption has larger variance and the logistic regression with wrong parametric assumption is biased. Moreover, when the sample size increases, the Asymmetric AdaBoost will have smaller variance and the losses are closer to the sample Bayes risk. The logistic regression is still biased and has higher losses than the Asymmetric AdaBoost except in the two tails. This shows that the Asymmetric AdaBoost that produces a nonparametric classifier will suffer from higher variance if the sample size is small. But, as the sample size increases, the Asymmetric AdaBoost will produce an unbiased classifier and achieve lower losses than logistic regression which is biased.

In table 3, the DGP3 is unbalanced. The event $y = 1$ is significantly fewer than $y = -1$. We can see that the Asymmetric AdaBoost works better when the minority of the events is penalized more heavily. The Asymmetric AdaBoost has lower losses on the right-hand side where $y = 1$ is penalized more heavily, and higher losses on the left-hand side where $y = -1$ is penalized more heavily. In the unbalanced DGP, the logistic regression only focuses on the event that is the majority. However, Asymmetric AdaBoost still tries to model both events. Hence, if one is interested in predicting the less common event, e.g. recession over expansion, the Asymmetric AdaBoost will give lower losses. Moreover, as the sample size increases, we see that the Asymmetric AdaBoost converges to the Bayes risk on both sides and catches up with logistic regression on the left-hand side.

In summary, the Asymmetric AdaBoost is consistent in the sense that the losses of the classifier produced converges to the sample Bayes risk as the sample size increases. Compared with the logistic regression, the Asymmetric AdaBoost has smaller losses when the logistic regression is misspecified and the sample size if large. Moreover, the Asymmetric AdaBoost is better than the logistic regression if one is more interested in

predicting the less common events when the data is unbalanced.

5 Application

In this section, we predict the direction of change in Real Personal Income using data of McCracken and Ng (2016). We use all variables from the dataset except New Orders for Consumer Goods, New Orders for Nondefense Capital Goods, Trade Weighted U.S. Dollar Index: Major Currencies and VXO where the number of missing values are over 40 and Consumer Sentiment Index where data frequency is different from others. We construct the direction of change in Real Personal Income as the dependent variable. The data is monthly data from January, 1960 to September, 2018. There are in total 705 observations and 124 explanatory variables including a lag variable of the dependent variable. We use a rolling sample scheme and test the one-period ahead prediction. We use window width of 200 and 500. The results are shown in Table 4.

In the application, we see that the Asymmetric AdaBoost gives competitive results as penalized logistic regression with LASSO. Even though we do not know the Bayes risk in this case, the resulted losses are much smaller than 0.5 which is the expected loss for random guessing. In summary, both Asymmetric AdaBoost and penalized logistic regression with LASSO would be suitable for the application.

6 Conclusions

In this paper, we introduce a new Asymmetric AdaBoost algorithm which produces an additive regression model from maximizing a new risk function, namely the asymmetric exponential risk function. The new Asymmetric AdaBoost algorithm is based on the asymmetric exponential risk function, which maps into a binary decision making problem given a utility function. Furthermore, by carefully establishing the asymmetry in the risk function in accordance to the binary decision making, we show that our Asymmetric AdaBoost algorithm is closely related to the maximum score regression (Manski 1975, 1985) and the binary prediction literature in economics (Granger and Pesaran 2000, Lee and Yang 2006, Lahiri and Yang 2012, and Elliot and Lieli 2013), all of which however deal with low-dimensional predictor space. Asymmetric AdaBoost can handle the maximum score and binary prediction when the predictors are high-dimensional. Theoretical results show that Asymmetric AdaBoost will converge to Bayes risk as $n \to \infty$. Simulation results show that Asymmetric AdaBoost is a competitive approach in binary classification/prediction.

7 Appendix

7.1 Proof of Theorem 2

We use greedy method to minimize the exponential risk function iteratively.

Step 1 is to look for f_{m+1}. Suppose we have finished m iterations, the current classifier is denoted as $F_m(x) = \sum_{s=1}^{m} c_s f_s(x)$. In the next iteration, we are seeking an update $c_{m+1} f_{m+1}(x)$ for the function fitted from previous iterations $F_m(x)$. The updated classifier would take the form

$$F_{m+1}(x) = F_m(x) + c_{m+1} f_{m+1}(x).$$

The risk for $F_{m+1}(x)$ will be

$$\begin{aligned} R(F_{m+1}(x)) &= R(F_m(x) + c_{m+1} f_{m+1}(x)) \\ &= \mathbb{E}\left[e^{-y(F_m(x)+c_{m+1}f_{m+1}(x))}\right]. \end{aligned} \tag{44}$$

Expand w.r.t. $f_{m+1}(x)$,

$$\begin{aligned} R(F_{m+1}(x)) &\approx \mathbb{E}\left[e^{-yF_m(x)}\left[1 - yc_{m+1}f_{m+1}(x) + \frac{y^2 c_{m+1}^2 f_{m+1}^2(x)}{2}\right]\right] \\ &= \mathbb{E}\left[e^{-yF_m(x)}\left(1 - yc_{m+1}f_{m+1}(x) + \frac{c_{m+1}^2}{2}\right)\right]. \end{aligned}$$

The last equality holds since $y \in \{-1, 1\}$, $f_{m+1}(x) \in \{-1, 1\}$, and $y^2 = f_{m+1}^2(x) = 1$. $f_{m+1}(x)$ only appears in the second term in the parenthesis, so minimizing the risk function (43) w.r.t. $f_{m+1}(x)$ is equivalent to maximizing the second term in the parenthesis which results in the following conditional expectation

$$\max_f \mathbb{E}\left[e^{-yF_m(x)} y c_{m+1} f_{m+1}(x) | x\right].$$

For any $c > 0$ (we will prove this later), we can omit c_{m+1} in the above objective function

$$\max_f \mathbb{E}\left[e^{-yF_m(x)} y f_{m+1}(x) | x\right].$$

To compare it with Algorithm 1, here we define weight $w = w(y, x) = e^{-yF_m(x)}$. Later we will see that this weight w is equivalent to that shown in Algorithm 1. So the above optimization can be seen as maximizing a weighted conditional expectation

$$\max_f \mathbb{E}_w\left[y f_{m+1}(x) | x\right] \tag{45}$$

where $\mathbb{E}_w(y|x) := \frac{\mathbb{E}(wy|x)}{\mathbb{E}(w|x)}$ refers to a weighted conditional expectation. Note that (44)

$$\begin{aligned} & \mathbb{E}_w\left[yf_{m+1}(x)|x\right] \\ = \; & P_w(y=1|x) f_{m+1}(x) - P_w(y=-1|x) f_{m+1}(x) \\ = \; & \left[P_w(y=1|x) - P_w(y=-1|x)\right] f_{m+1}(x), \end{aligned} \tag{46}$$

where $P_w(y|x) = \frac{\mathbb{E}(w|y,x)\Pr(y|x)}{\mathbb{E}(w|x)}$. Solve the maximization problem (44). Since $f_{m+1}(x)$ only takes 1 or -1, it should be positive whenever $P_w(y=1|x) - P_w(y=-1|x)$ is positive and -1 whenever $P_w(y=1|x) - P_w(y=-1|x)$ is negative. Hence, the solution for $f_{m+1}(x)$ is

$$f_{m+1}(x) = \begin{cases} 1 & P_w(y=1|x) - P_w(y=-1|x) > 0 \\ -1 & \text{otherwise.} \end{cases} \tag{47}$$

Step 2 is to look for c_{m+1}. Minimize the risk function (43) w.r.t. c_{m+1}

$$\begin{aligned} c_{m+1} &= \arg\min_{c_{m+1}} \mathbb{E}_w\left(e^{-c_{m+1}yf_{m+1}(x)}\right) \\ \mathbb{E}_w\left(e^{-c_{m+1}yf_{m+1}(x)}\right) &= P_w(y=f_{m+1}(x))\, e^{-c_{m+1}} + P_w(y\neq f_{m+1}(x))\, e^{c_{m+1}} \\ \frac{\partial \mathbb{E}_w\left(e^{-cyf_{m+1}(x)}\right)}{\partial c} &= -P_w(y=f_{m+1}(x))\, c_{m+1}e^{-c_{m+1}} + P_w(y\neq f_{m+1}(x))\, c_{m+1}e^{c_{m+1}} \end{aligned}$$

Let

$$\frac{\partial \mathbb{E}_w\left(e^{-c_{m+1}yf_{m+1}(x)}\right)}{\partial c_{m+1}} = 0,$$

and we have

$$P_w(y=f_{m+1}(x))\, c_{m+1}e^{-c_{m+1}} = P_w(y\neq f_{m+1}(x))\, c_{m+1}e^{c_{m+1}}.$$

Solve for c_{m+1}, we obtain

$$c_{m+1} = \frac{1}{2}\log\frac{P_w(y=f_{m+1}(x))}{P_w(y\neq f_{m+1}(x))} = \frac{1}{2}\log\left(\frac{1-err_{m+1}}{err_{m+1}}\right),$$

where $err_{m+1} = P_w(y\neq f_{m+1}(x))$ is the error rate of $f_{m+1}(x)$. Note that $c_{m+1} > 0$ as long as the error rate is smaller than 50%, i.e. the assumption $c_{m+1} > 0$ holds for any learner that is better than random guessing.

Step 3 is to update the learner and get ready for the next iteration. Now we have finished the steps of one iteration and can get our updated classifier by

$$F_{m+1}(x) = F_m(x) + \left(\frac{1}{2}\log\left(\frac{1-err_{m+1}}{err_{m+1}}\right)\right) f_{m+1}(x).$$

Note that in the next iteration, the weight we defined w_{m+1} will be

$$w_{m+1} = e^{-yF_{m+1}(x)} = e^{-y(F_m(x)+c_{m+1}f_{m+1}(x))} = w_m \times e^{-c_{m+1}f_{m+1}(x)y}.$$

Thus the function and weights update are of an identical form to those used in Algorithm 1. □

7.2 Proof of Theorem 5

Proof. Let $F^* = \arg\min_F R_\tau(F)$ be the Bayes classifier. Let $P_w(x,y) = \frac{t(y,x)\mathcal{P}(x,y)}{\int t(y,x)\mathcal{P}(x,y)dydx}$. Then $P_w(x,y)$ defines a probability distribution of (x,y) on $\chi \times \{\pm 1\}$. By definition,

$$\begin{aligned}
R_{\psi,\tau}(F_i) &= \mathbb{E}\left(t(y,x)\, e^{-yF_i}\right) \\
&= \int t(y,x)\, e^{-yF_i}\mathcal{P}(x,y)\, dydx \\
&= \int t(y,x)\,\mathcal{P}(x,y)\, dydx \cdot \int e^{-yF_i}\frac{t(y,x)\,\mathcal{P}(x,y)}{\int t(y,x)\,\mathcal{P}(x,y)\, dydx} dydx \\
&= \int t(y,x)\,\mathcal{P}(x,y)\, dydx \cdot \int e^{-yF_i} P_w(x,y)\, dydx \\
&= C\int e^{-yF_i} P_w(x,y)\, dydx,
\end{aligned}$$

where $C \equiv \int t(y,x)\,\mathcal{P}(x,y)\, dydx$ is positive and bounded. Moreover,

$$R^*_{\psi,\tau} = \inf_{F_i} R_{\psi,\tau}(F_i).$$

Hence, by Theorem 2,

$$R_{\psi,\tau} \to R^*_{\psi,\tau} \quad \text{implies that} \quad \int 1_{(y\neq F_i)} P_w(x,y)\, dydx \to \int 1_{(y\neq F^*)} P_w(x,y)\, dydx.$$

Rewrite the expression in terms of $\mathcal{P}(x,y)$, we have

$$\frac{1}{C}\int 1_{(y\neq F_i)} t(y,x)\,\mathcal{P}(x,y)\, dydx \to \frac{1}{C}\int 1_{(y\neq F^*)} t(y,x)\,\mathcal{P}(x,y)\, dydx.$$

Therefore,

$$R_\tau(F_i) = \int t(y,x)\, 1_{(y\neq F_i)}\mathcal{P}(x,y)\, dydx \to \int t(y,x)\, 1_{\left(y\neq F^*_L\right)}\mathcal{P}(x,y)\, dydx = R^*_\tau.$$

□

7.3 Proof of Theorem 6

We start with the asymmetric exponential risk function

$$R_{\psi,\tau}(F(x)) = \mathbb{E}\left(t(y,x)\, e^{-yF(x)}\right). \tag{48}$$

Step 1 is to look for the optimal $f_{m+1}(x)$ for each iteration. Suppose we have finished m iterations, the current classifier is denoted as $F_m(x) = \sum_{s=1}^{m} c_s f_s(x)$. In the next iteration, we are seeking an update $c_{m+1} f_{m+1}(x)$ for the function fitted by previous iterations $F_m(x)$. The updated classifier would be

$$F_{m+1}(x) = F_m(x) + c_{m+1} f_{m+1}(x). \tag{49}$$

The risk for the updated classifier is

$$R_{\psi,\tau}\left(F_m(x) + c_{m+1} f_{m+1}(x)\right) = \mathbb{E}\left(t(y,x)\, e^{-y(F_m(x)+c_{m+1} f_{m+1}(x))}\right). \tag{50}$$

Expand it w.r.t. $f_{m+1}(x)$

$$\begin{aligned}
& \mathbb{E}\left(t(y,x)\, e^{-y(F_m(x)+c_{m+1} f_{m+1}(x))}\right) && (51)\\
\approx\ & \mathbb{E}\left(t(y,x)\, e^{-yF_m(x)}\left(1 - yc_{m+1} f_{m+1}(x) + \frac{yc_{m+1}^2 f_{m+1}^2(x)}{2}\right)\right) && (52)\\
=\ & \mathbb{E}\left(t(y,x)\, e^{-yF_m(x)}\left(1 - yc_{m+1} f_{m+1}(x) + \frac{c_{m+1}^2}{2}\right)\right), && (53)
\end{aligned}$$

since $y^2 = f_{m+1}^2(x) = 1$ holds for all y and $f_{m+1}(x)$. Only the second term in the bracket contains $f_{m+1}(x)$, so minimizing the above risk function w.r.t. $f_{m+1}(x)$ is equivalent to maximizing the following expectation

$$\max_f \mathbb{E}\left(e^{-yF_m(x)} t(y,x)\, y f_{m+1}(x) \,|x\right), \tag{54}$$

for any $c_{m+1} > 0$. Let weights be $w \equiv e^{-yF_m(x)}$. Then we re-write the above maximization as

$$\max_f \mathbb{E}_w\left(t(y,x)\, y f_{m+1}(x) \,|x\right). \tag{55}$$

Solve the maximization problem

$$\begin{aligned}
& \max_f \mathbb{E}_w\left(t(y,x)\, y f_{m+1}(x) \,|x\right) && (56)\\
=\ & P_w(y=1|x)\, t(1,x)\, f_{m+1}(x) - P_w(y=-1|x)\, t(-1,x)\, f_{m+1}(x) && (57)\\
=\ & \left[P_w(y=1|x)\, t(1,x) - P_w(y=-1|x)\, t(-1,x)\right]\ f_{m+1}(x), && (58)
\end{aligned}$$

$f_{m+1}(x)$ should take the same sign as $P_w(y=1|x)\, t(1,x) - P_w(y=-1|x)\, t(-1,x)$.

The solution is

$$f_{m+1}(x) = \begin{cases} 1, & P_w(y=1|x)\, t(1,x) - P_w(y=-1|x)\, t(-1,x) > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{59}$$

Step 2 is to look for the optimal c_{m+1} for each iteration. After solving $f_{m+1}(x)$, we minimize the risk function (49) w.r.t. c_{m+1},

$$\begin{aligned} c_{m+1} &= \arg\min_{c} R_{\psi,\tau}\left(F_m(x) + c f_{m+1 m+1}(x)\right) && (60)\\ &= \arg\min_{c} \mathbb{E}\left(t(y,x)\, e^{-y(F_m(x)+c_{m+1} f_{m+1}(x))}\right) && (61)\\ &= \arg\min_{c} \mathbb{E}_w\left(t(y,x)\, e^{-y c_{m+1} f_{m+1}(x)}\right) && (62)\end{aligned}$$

Then

$$\begin{aligned} & \mathbb{E}_w\left(t(y,x)\, e^{-y c_{m+1} f_{m+1}(x)}\right) && (63)\\ = \;& P_w(y=1, f_{m+1}(x)=1)\, t(1,x)\, e^{-c_{m+1}} + P_w(y=-1, f_{m+1}(x)=-1)\, t(-1,x)\, e^{-c_{m+1}} && (64)\\ & + P_w(y=1, f_{m+1}(x)=-1)\, t(1,x)\, e^{c_{m+1}} + P_w(y=-1, f_{m+1}(x)=1)\, t(-1,x)\, e^{c_{m+1}} && (65)\end{aligned}$$

The first order condition from taking the derivative w.r.t. c_{m+1}

$$\begin{aligned} & \frac{\partial R_{\psi,\tau}\left(c_{m+1} f_{m+1}(x)\right)}{\partial c_{m+1}} && (66)\\ = \;& -P_w(y=1, f_{m+1}(x)=1)\, t(1,x)\, e^{-c_{m+1}} - P_w(y=-1, f_{m+1}(x)=-1)\, t(-1,x)\, e^{-c_{m+1}} && (67)\\ & + P_w(y=1, f_{m+1}(x)=-1)\, t(1,x)\, e^{c_{m+1}} + P_w(y=-1, f_{m+1}(x)=1)\, t(-1,x)\, e^{c_{m+1}} && (68)\end{aligned}$$

gives the optimal c_{m+1} from solving the following

$$\begin{aligned} & P_w(y=1, f_{m+1}(x)=1)\, t(1,x)\, e^{-c_{m+1}} + P_w(y=-1, f_{m+1}(x)=-1)\, t(-1,x)\, e^{-c_{m+1}} && (69)\\ = \;& P_w(y=1, f_{m+1}(x)=-1)\, t(1,x)\, e^{c_{m+1}} + P_w(y=-1, f_{m+1}(x)=1)\, t(-1,x)\, e^{c_{m+1}}, && (70)\end{aligned}$$

where $P_w(y=1, f_{m+1}(x)=1)$ is the rate of true positive (TP), $P_w(y=-1, f_{m+1}(x)=-1)$ is the rate of true negative (TN), $P_w(y=1, f_{m+1}(x)=-1)$ is the rate of false negative (FN), $P_w(y=-1, f_{m+1}(x)=1)$ is the rate of false positive (FP). Hence, rewriting it as

$$\left[\mathrm{TP} \times t(1,x) + \mathrm{TN} \times t(-1,x)\right] e^{-c_{m+1}} = \left[\mathrm{FN} \times t(1,x) + \mathrm{FP} \times t(-1,x)\right] e^{c_{m+1}}, \quad (71)$$

we obtain the optimal c_{m+1}

$$c_{m+1} = \frac{1}{2}\log\left(\frac{\mathrm{TP}\times t(1,x) + \mathrm{TN} \times t(-1,x)}{\mathrm{FN} \times t(1,x) + \mathrm{FP} \times t(-1,x)}\right) = \frac{1}{2}\log\left(\frac{1-err_{m+1}}{err_{m+1}}\right), \quad (72)$$

where $err_{m+1} = \mathbb{E}_w\left(t(y,x) \times 1_{(y \neq f_{m+1}(x))}\right)$.

Step 3 is to update the current strong learner and get ready for the next iteration. In the next iteration, we have

$$F_{m+1}(x) \leftarrow F_m(x) + c_{m+1} f_{m+1}(x). \tag{73}$$

Hence

$$\begin{aligned} w_{m+1} &= e^{-yF_{m+1}(x)} && (74)\\ &= e^{-y(F_m(x)+c_{m+1}f_{m+1}(x))} && (75)\\ &= w_m \times e^{-c_{m+1}yf_{m+1}(x)}, && (76) \end{aligned}$$

is of identical form as in Algorithm 2. □

7.4 Proof of Theorem 7

Hereby we provide the proof of Theorem 7 following Bartlett and Traskin (2007) for our asymmetric exponential risk.

Notation. Let $R_{\psi,\tau,n}$ be the sample version of $R_{\psi,\tau}$ with sample size n and the set of k-combinations, $k \in \mathbb{N}$, of functions in $\mathcal{H}$

$$\mathcal{F}^k = \left\{ F | F = \sum_{i=1}^{k} \lambda_i h_i, \lambda_i \in \mathbb{R}, h_i \in \mathcal{H} \right\}. \tag{77}$$

Define the squashing function $\pi_l(\cdot)$ to be

$$\pi_l(x) = \begin{cases} l, & x > l \\ x, & x \in [-l, l] \\ -l, & x < -l. \end{cases} \tag{78}$$

Then the set of truncated functions is

$$\pi_l \circ \mathcal{F} = \left\{ \tilde{F} | \tilde{F} = \pi_l(F), F \in \mathcal{F} \right\}. \tag{79}$$

The set of classifiers based on a class $\mathcal{F}$ is denoted by

$$\mathcal{G} = \{ G(F) | F \in \mathcal{F} \}. \tag{80}$$

Let

$$\varphi_\lambda = \inf_{\alpha \in [-\lambda, \lambda]} t(y, x) e^{-\alpha}. \tag{81}$$

Assumption 1. *Let n be sample size. Let there exist non-negative sequences $t_n \to \infty$, $\zeta_n \to \infty$ and a sequence $\left\{\bar{F_n}\right\}_{n=1}^{\infty}$ of reference functions such that*

$$R_{\psi,\tau}\left(\bar{F_n}\right) \underset{n\to\infty}{\to} R^*_{\psi,\tau}, \tag{82}$$

and suppose that the following conditions are satisfied.

1. *Uniform convergence of t_n-combinations.*

$$\sup_{F\in\pi_{\zeta_n}\circ\mathcal{F}^{t_n}} \left|R_{\psi,\tau}\left(F\right) - R_{\psi,\tau,n}\left(F\right)\right| \overset{a.s.}{\underset{n\to\infty}{\to}} 0. \tag{83}$$

2. *Convergence of empirical exponential risks for the sequence $\left\{\bar{F_n}\right\}_{n=1}^{\infty}$.*

$$\max\left\{0, R_{\psi,\tau,n}\left(\bar{F_n}\right) - R_{\psi,\tau}\left(\bar{F_n}\right)\right\} \overset{a.s.}{\underset{n\to\infty}{\to}} 0. \tag{84}$$

3. *Algorithmic convergence of t_n-combinations.*

$$\max\left\{0, R_{\psi,\tau,n}\left(F_{t_n}\right) - R_{\psi,\tau,n}\left(\bar{F_n}\right)\right\} \overset{a.s.}{\underset{n\to\infty}{\to}} 0. \tag{85}$$

Evidence that Asymmetric AdaBoost and the asymmetric exponential risk satisfies Assumption 1 can be find in Bartlett and Traskin (2007) where they discuss the same topic for symmetric AdaBoost.

Proof of Theorem 7. We follow the procedure of Bartlett and Traskin (2007) with our asymmetric exponential risk function. For almost every outcome ω on the probability space we can define sequences $\epsilon_n^1\left(\omega\right) \to 0$, $\epsilon_n^2\left(\omega\right) \to 0$,$\epsilon_n^3\left(\omega\right) \to 0$, such that for almost all ω the following inequalities are true.

$$\begin{aligned}
R\left(\pi_{\zeta_n}\left(F_{t_n}\right)\right) &\leq R_n\left(\pi_{\zeta_n}\left(F_{t_n}\right)\right) + \epsilon_n^1\left(\omega\right) \to 0 \text{ by (82)} \\
&\leq R_n\left(F_{t_n}\right) + \epsilon_n^1\left(\omega\right) + \varphi_{\zeta_n} \qquad (86) \\
&\leq R_n\left(\bar{F_{t_n}}\right) + \epsilon_n^1\left(\omega\right) + \varphi_{\zeta_n} + \epsilon_n^2\left(\omega\right) \text{ by (84)} \\
&\leq R\left(\bar{F_n}\right) + \epsilon_n^1\left(\omega\right) + \varphi_{\zeta_n} + \epsilon_n^2\left(\omega\right) + \epsilon_n^3\left(\omega\right) \text{ by (83).} \qquad (87)
\end{aligned}$$

Inequality (85) follows from the convexity of $\varphi\left(\cdot\right)$. By choice of the sequence $\left\{\bar{F_n}\right\}_{n=1}^{\infty}$, we have $R\left(\bar{F_n}\right) \to R^*$ and $\varphi_{\zeta_n} \to 0$. And from (86) follows $R\left(\pi_{\zeta_n}\left(F_{t_n}\right)\right) \overset{a.s.}{\to} R^*$. Eventually we can use Lemma 1 to conclude that

$$L\left(G\left(\pi_{\zeta_n}\left(F_{t_n}\right)\right)\right) \overset{a.s.}{\to} L^*. \tag{88}$$

And for $\zeta_n > 0$, we have $G\left(\pi_{\zeta_n}\left(F_{t_n}\right)\right) = G\left(F_{t_n}\right)$, therefore

$$L\left(G\left(F_{t_n}\right)\right) \overset{a.s.}{\to} L^*. \tag{89}$$

Hence, Asymmetric AdaBoost is consistent if stopped after t_n steps. □

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician 46*(3), 175–185.

Bartlett, P. L., M. I. Jordan, and J. D. McAuliffe (2006). Convexity, Classification, and Risk Bounds. *Journal of the American Statistical Association 101*(473), 138–156.

Bartlett, P. L. and M. Traskin (2007). AdaBoost is consistent. *Journal of Machine Learning Research 8*, 2347–2368.

Bliss, C. I. (1934a). The method of probits. *Science 79*(2037), 38–39.

Bliss, C. I. (1934b). The method of probits. *Science 79*(2037), 409–410.

Breiman, L. (1984). *Classification and Regression Trees*. Routledge.

Chandrasekaran, V. and M. I. Jordan (2012). Computational and Statistical Tradeoffs via Convex Relaxation. *Proceedings of the National Academy of Sciences of the United States of America 110*(13), E1181–90.

Diaconis, P. and D. Freedman (1993). Nonparametric Binary Regression: A Bayesian Approach. *The Annals of Statistics 21*(4), 2108–2137.

Elliott, G. and R. P. Lieli (2013). Predicting binary outcomes. *Journal of Econometrics 174*(1), 15–26.

Freund, Y. and R. Schapire (1996). Experiments with a New Boosting Algorithm. Technical report.

Freund, Y. and R. E. Schapire (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences 55*, 119–139.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics 28*(2), 337–407.

Gaddum, J. (1933). Report on Biological Standards III: Methods of Biological Assay Depending on Quantal Response. *Special Report Series of the Medical Research Council 183* (London: Medical Research Council).

Granger, C. W. J. and M. H. Pesaran (2000). Economic and statistical measures of forecast accuracy. *Journal of Forecasting 19*(7), 537–560.

Klein, R. W. and R. H. Spady (1993). An Efficient Semiparametric Estimator for Binary Response Models. *Econometrica 61*(2), 387.

Lahiri, K. and L. Yang (2012). Forecasting binary outcomes. In G. Elliott and A. Timmermann (Eds.), *Handbook of Economic Forecasting*, Volume 2, pp. 1025–1106. SSRN.

Manski, C. F. (1975). Maximum score estimation of the stochastic utility model of choice. *Journal of Econometrics 3*(3), 205–228.

Manski, C. F. (1985). Semiparametric analysis of discrete response. Asymptotic properties of the maximum score estimator. *Journal of Econometrics 27*(3), 313–333.

McCracken, M. W. and S. Ng (2016). FRED-MD: A Monthly Database for Macroeconomic Research. Technical Report 4.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

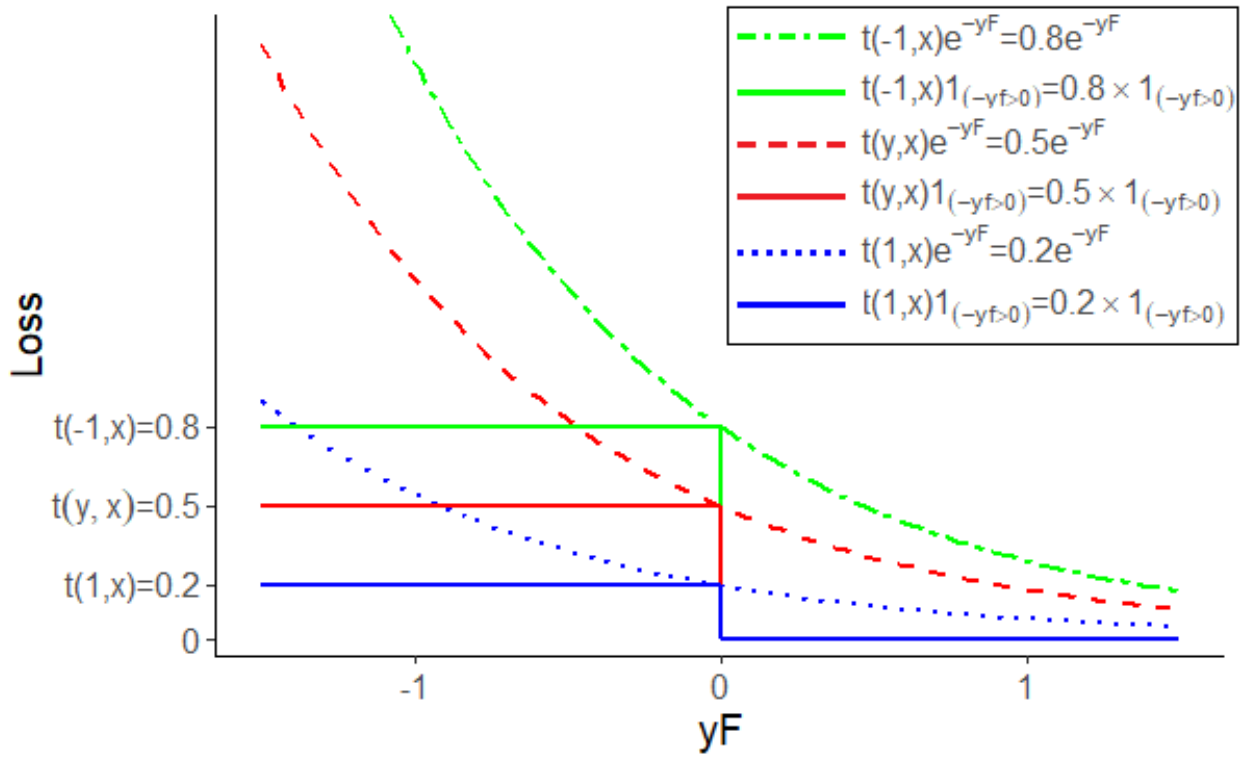

Figure 1: (Asymmetric) Exponential Loss and Score Loss

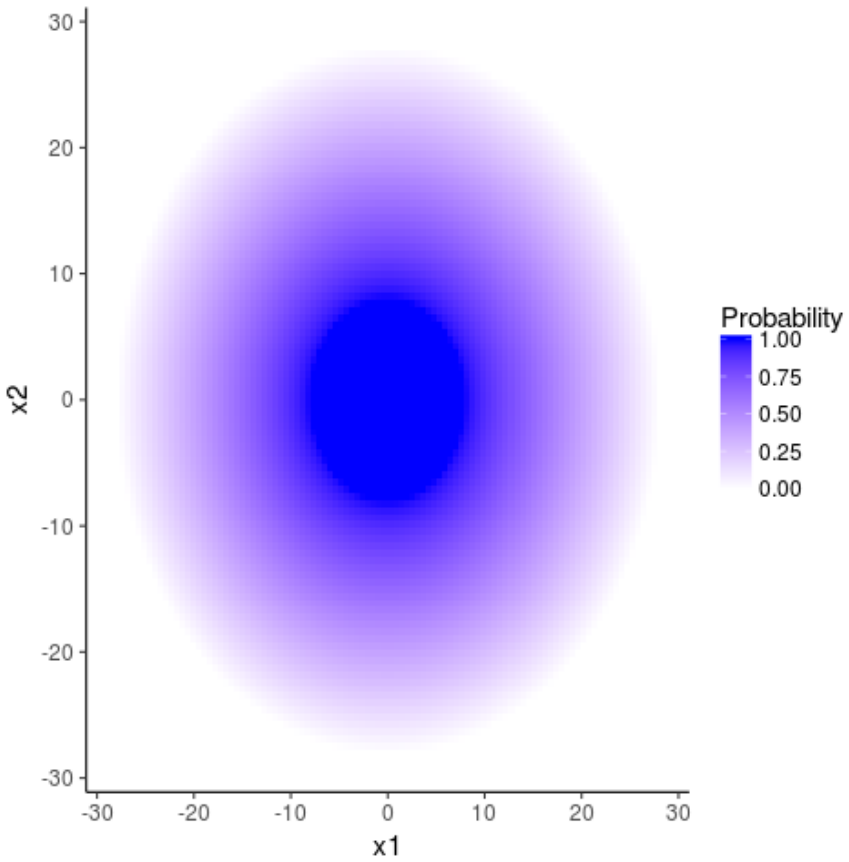

Figure 2: Conditional Probability of the Circle Model

Table 1: Linear Logit Model

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	AdaBoost	544.49	997.15	1379.57	1602.51	1712.86	1607.31	1372.14	1005.47	545.34
$n = 1000$	LASSO	492.64	934.23	1266.53	1479.09	1550.39	1482.48	1271.21	933.40	493.81
	Bayes Risk	482.03	885.91	1178.71	1360.64	1419.00	1359.56	1182.63	886.65	482.28
	AdaBoost	774.05	1263.55	1728.17	2001.50	2085.16	1981.78	1739.04	1300.41	773.65
$n = 100$	LASSO	509.96	1026.17	1483.54	1814.39	1973.22	1843.52	1482.58	1015.68	513.55
	Bayes Risk	482.66	885.49	1180.18	1357.64	1418.54	1358.78	1179.72	885.95	483.10

Table 2: Quadratic Logit Model

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	AdaBoost	524.15	958.94	1330.60	1614.64	1736.21	1570.27	1316.51	951.87	516.89
$n = 1000$	LASSO	510.42	1021.05	1495.94	1841.09	1949.09	1805.75	1442.74	977.64	488.46
	Bayes Risk	469.27	866.57	1168.82	1358.86	1422.40	1358.41	1170.30	867.15	469.06
	AdaBoost	765.08	1304.56	1734.17	2017.23	2121.00	2049.07	1769.74	1335.79	819.19
$n = 100$	LASSO	501.26	1017.77	1552.46	2076.06	2346.12	2063.88	1541.83	1030.40	514.38
	Bayes Risk	468.92	866.33	1168.62	1357.71	1422.94	1358.86	1168.68	866.11	469.20

Table 3: Circle Model

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	AdaBoost	402.10	719.11	835.78	893.85	981.62	1041.31	1058.47	794.18	443.48
$n = 1000$	LASSO	358.01	715.63	1073.40	1430.46	1792.02	2158.52	1937.20	1283.70	641.60
	Bayes Risk	276.30	513.59	700.54	833.49	902.57	897.67	814.02	640.97	372.48
	AdaBoost	554.64	841.00	1090.67	1256.00	1353.53	1387.48	1336.96	1189.13	807.87
$n = 100$	LASSO	358.70	718.68	1082.37	1451.54	1848.22	2272.30	2049.76	1344.72	658.73
	Bayes Risk	276.88	512.74	700.44	834.47	902.52	897.30	812.37	640.81	372.76

Table 4: Application Result

Window Width	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
200	AdaBoost	0.117	0.136	0.153	0.131	0.123	0.094	0.074	0.046	0.026
	LASSO	0.091	0.137	0.139	0.133	0.117	0.097	0.071	0.052	0.024
500	AdaBoost	0.119	0.145	0.159	0.139	0.129	0.103	0.075	0.056	0.028
	LASSO	0.081	0.136	0.144	0.135	0.124	0.099	0.083	0.053	0.028