Asymmetric AdaBoost for High-dimensional Maximum Score Regression*

Jianghao Chu † Tae-Hwy Lee‡ Aman Ullah§

August 28, 2023

Abstract

Carter Hill's numerous contributions (books and articles) in econometrics stand out especially in pedagogy. An important aspect of his pedagogy is to integrate "theory and practice" of econometrics, as coined into the titles of his popular books. The new methodology we propose in this paper is consistent with these contributions of Carter Hill. In particular, we bring the maximum score regression of Manski (1975, 1985) to high dimension in theory and show that the "Asymmetric AdaBoost" provides the algorithmic implementation of the high dimensional maximum score regression in practice. Recent advances in machine learning research have not only expanded the horizon of econometrics by providing new methods but also provided the algorithmic aspects of many of traditional econometrics methods. For example, Adaptive Boosting (AdaBoost) introduced by Freund and Schapire (1996) has gained enormous success in binary/discrete classification/prediction. In this paper, we introduce the "Asymmetric AdaBoost" and relate it to the maximum score regression in the algorithmic perspective. The Asymmetric AdaBoost solves 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 non-convex 0-1 risk. The convex risk function gives a huge computational advantage over non-convex risk functions of Manski (1975, 1985) especially when the data is high-dimensional. The resulting nonparametric classifier is more robust than the parametric classifiers whose performance depends 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 of risk that can be achieved by all classifiers. Monte Carlo experiments show that the 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 business cycle turning points as in Ng (2014).

Key Words: Maximum Score Regression, High Dimension, Asymmetric AdaBoost, Convex Relaxation, Exponential Risk.

JEL Classification: C25, C44, C53, C55

*We thank seminar participants at University of Southern California, UC Riverside (Economics), UC Riverside (Data Science Center), Peking University, Chinese Academy of Sciences, Central University of Finance and Economics, Chinese Meeting of Econometric Society, Asian Meeting of Econometric Society, Joint Statistical Meetings, California Econometrics Conference, and Midwest Econometrics Group.

†JPMorgan Chase & Co, Jersey City, NJ 07310. E-mail: jianghao.chu@jpmchase.com

‡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

Data with a large number of variables relative to the sample size, namely high-dimensional data, are becoming more and more prevalent in empirical economics as well as statistics and computer science. One of the most successful applications of high-dimensional data in economics as well as other sciences is to construct empirical models for the forecasting of binary outcomes and making binary decisions. Examples in forecasting include predicting firm solvency, the legitimacy of credit card transactions, directional forecasts of financial prices, whether a loan is paid off or not, or whether an introduced foreign plant species will become invasive or not. Such forecasts are often translated into decisions which are binary in character, e.g. the loan is granted or it is not, the student is admitted to the school or not, the candidate is hired or not hired, the surgery is undertaken or it is not, importation of a foreign plant species is allowed or not. Various traditional statistical approaches to binary classification are available in the literature, from discriminant analysis, logit or probit models to less parametric estimates of the conditional probability model for the outcome variable such as semiparametric single-index models (Ichimura, 1993; Klein and Spady, 1993).

Typically, most estimation techniques used for binary classification do not make use of the loss function implicit in the underlying decision/prediction problem. For example logit and probit models are estimated to maximize the likelihood of the model, irrespective of the relative usefulness of true positives or true negatives. Nonparametric methods seek the best fit for the conditional probability based on the loss function (typically squared error) rather than the appropriate loss function for the decision problem. In most applications, the relative costs of making errors, false negatives and false positives, are rarely balanced in the way that could be used to motivate these approaches. In detecting credit card fraud, "wasting" resources on checking that the customer has control over their credit card is perhaps less costly than failing to do so when their credit card number has been stolen. Elliott and Lieli (2013) point out that even with local misspecifications that are difficult to detect using standard specification tests, parametric models of the conditional probability of a positive outcome can perform arbitrarily poorly when the loss function is ignored at the estimation stage. They further propose the "maximum utility estimator" which is a semiparametric method that requires far less information to attain maximal utility, and through the utilization of the loss

function at the estimation stage has useful properties given any misspecification. The maximum utility estimator builds upon and extends results of Manski (1975, 1985).

This paper extends the method of Manski (1975, 1985) and Elliott and Lieli (2013) to high-dimensional data and model misspecification in the order of independent variables. We consider the prediction of a binary variable $y \in \{1, -1\}$, e.g. $y = 1$ if the economy is in expansion and $y = -1$ if the economy is in recession. 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. $\tau(x)$ is a utility-based weight function that assigns different penalties conditioning on the state variable y and characteristics x as shown in Section 3. 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.

We propose a nonparametric method which minimizes an 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 a given parametric form. Moreover, we use component-wise algorithm and select only one independent variable at each iteration to overcome the issue of high-dimensionality.

There are some prediction problems that do not fit the framework examined here. A forecaster providing forecasts that might be used by a number of different users might not consider the loss function. For example a weather forecaster providing a forecast of whether or not it might rain might simply report an estimate of the conditional probability of rain and let different users interpret the information differently. We will also rule out feedback of the prediction to the conditional probability of the event to be predicted, which means that the methods are not appropriate for predictions of outcomes where there is this type of feedback. Such feedback occurs for example in predicting success of job training programs, where entry to the program affects the chance of getting

a job. However a myriad of problems are not ruled out, where the prediction is not important for the distribution of the outcomes and the econometrician is willing to elicit a loss function.

The rest of the paper is organized as follows. Section 2 introduces the binary choice model and the maximum score approach. Section 3 relate the binary classification problem with decision theory. In Section 4, 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 5, we examine the finite sample properties of Asymmetric AdaBoost via Monte Carlo simulations. Section 6 predicts business cycle turning points as in Ng (2014). Section 7 concludes. All technical derivations and proofs are presented in the Appendix.

2 Maximum Score Regression

In this paper, we consider the binary choice model given by

$$y = \begin{cases} 1 & \text{if } \phi(x) \geq \epsilon \\ -1 & otherwise. \end{cases} \tag{3}$$

where $\phi(\cdot)$ is an unknown function, x is a vector of exogenous variables, ϵ is a random disturbance. We assume that observations $\{x_i, y_i\}$ are independently and identically distributed. However, we do not require any prior knowledge on the functional form of $\phi(\cdot)$ or the distribution of ϵ.

Manski (1975) proposes to obtain a classifier $G(x) \in \{1, -1\}$ by maximizing the "score"

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

which is called the maximum score approach. Note that

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

Hence, $G(x)$ should take 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 function in this section as "symmetric" since the optimal decision rule is $\Pr(y = 1|x) > 0.5$. Similarly, we refer to the risk functions in Section 4 as "asymmetric" since the the optimal decision rule is not $\Pr(y = 1|x) > 0.5$.

Note that the score function (4) is a linear transformation of the misclassification probability (1) with $\tau = 0.5$,

$$S(G) = \mathbb{E}[yG(x)] = -4 \times \mathbb{E}\left[\frac{1}{2} \times 1_{(G(x)\neq y)}\right] + 1 = -4 \times R_{0.5}(G) + 1. \tag{8}$$

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

Remark 2. Note that the risk function (1) is often referred to as the 0-1 risk since the indicator function takes value 1 when the classification is wrong and 0 otherwise. We would use these names interchangeably with the negative score function used in the maximum score approach in the rest of the paper.

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"

$$R^*_{0.5} = \inf_G R_{0.5}(G) = \mathbb{E} \min\left\{\frac{1}{2}\Pr(y = 1|x), \frac{1}{2}\Pr(y = -1|x)\right\}, \tag{9}$$

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

The maximum score approach yields a classifier that minimizes the misclassification probability (1) with $\tau = 0.5$. 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 non-convex which lead to computation difficulty especially when the sample size is large. Last but not the least, the method does not work if covariates are high-dimensional.

3 Decision Theory for Binary Prediction/Classification

In a more general case, it may not be optimal to use $\Pr(y = 1|x) > 0.5$ as the threshold. 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

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

(10)

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 optimal classification/prediction is $G(x)=1$ if the expected utility of $G(x)=1$ is greater than $G(x)=-1$:

$$\Pr(y=1|x)\,u_{1,1}(x)+\Pr(y=-1|x)\,u_{1,-1}(x)>\Pr(y=1|x)\,u_{-1,1}(x)+\Pr(y=-1|x)\,u_{-1,-1}(x). \tag{11}$$

Hence, $G(x)=1$ if

$$\Pr(y=1|x)>(u_{-1,-1}(x)-u_{1,-1}(x))=1-\tau(x), \tag{12}$$

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

Setting the losses as negative utilities, the above problem can be written as minimizing a state-dependent risk function as follows:

$$R_\tau(G)=\mathbb{E}\left(t(y,x)\,1_{(-yG(x)>0)}\right), \tag{13}$$

where

$$t(y,x)=\begin{cases}\tau(x) & y=1\\ 1-\tau(x) & y=-1\end{cases} \tag{14}$$

is a non-negative function of outcome variable y and characteristics x. Similarly, we denote the risk of classification using the optimal decision rule (12)

$$R_\tau^*=\inf_G R_\tau(G(x))=\mathbb{E}\{\min[t(1,x)\Pr(y=1|x),t(-1,x)\Pr(y=-1|x)]\} \tag{15}$$

as the Bayes risk which is the minimal risk that can be achieved. The risk function (13) is essentially the same as the misclassification probability (1) with argument $F \in \mathbb{R}$ instead of $G \in \{1,-1\}$. As we have shown before, the misclassification probability, namely the 0-1 risk, is a linear transformation of the risk function used in the maximum score approach.

The optimal classifier

$$G^*_\tau(x) = \begin{cases} 1 & \Pr(y=1|x) > 1-\tau(x) \\ -1 & otherwise. \end{cases} \tag{16}$$

uses the classification rule (12) that is a function of the state-dependent utilities of the economic agent and achieves the Bayes risk (15).[1]

Remark 3. 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.

4 Asymmetric Exponential Loss

The score risk (13) is non-convex which lead to high computation cost especially when the sample size is large and/or covariates are high-dimensional. In this section, we introduce a new risk function, namely the asymmetric exponential risk, for solving binary classification/prediction under state-dependent losses. 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.

4.1 Maximum Score

Before we introduce our convex surrogate risk function, we first point out that it is a common practice to assume that the binary classifier $G(x)$ in (4) is taking the sign of a real valued function, i.e. $G(x) = \text{sign}[F(x)]$ where $F(x) \in \mathbb{R}$. Manski (1975) assumes that $G(x) = \text{sign}[x'\beta]$ where the function $F(x) = x'\beta$ is linear in x. It is worth noting that we do not impose the linearity assumption or any other parametric assumption in our method. Hence, the classifier is nonparametric. However, without loss of generality, we also assume $G(x) = \text{sign}[F(x)]$. Note that this assumption does not jeopardize the generality of our classifier as long as the inner function $F(x)$ is flexible enough.

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

$$\max_G E\left(t(y,x)yG\left(x\right)\right). \tag{17}$$

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

Replace $G(x)$ with $\text{sign}[F(x)]$, then (13) becomes

$$R_\tau\left(\text{sign}[F(x)]\right) = \mathbb{E}\left(t\left(y,x\right)1_{(-y\,\text{sign}[F(x)]>0)}\right) = \mathbb{E}\left(t\left(y,x\right)1_{(-yF(x)>0)}\right), \tag{18}$$

where

$$t(y,x) = \begin{cases} \tau(x) & y = \quad 1 \\ 1-\tau(x) & y = -1. \end{cases} \tag{19}$$

It is interesting to find that in (18) the latter equality shows the equivalence of using the score, $y\,\text{sign}[F(x)] \in \{-1,1\}$ and using which is called the margin in the machine learning literature, $yF(x) \in \mathbb{R}$. From Figure 1, it is easy to see that the risk function (18) is non-convex since it includes the indicator function $1_{(-yF(x)>0)}$. The non-convexity would lead to high computation costs and greatly limit the applicability of the risk function especially when the sample size is large and/or the data is high-dimensional.

4.2 Convex Surrogate

Bartlett, Jordan, and McAuliffe (2006) discuss the "convex relaxation" of non-convex risk functions commonly used in the classification literature. It is possible to use the exponential function

$$\psi(x) = e^{-yF(x)}, \tag{20}$$

as used in AdaBoost as a convex surrogate of the non-convex indicator function of the margin $1_{(-yF(x))>0}$. To solve the non-convex optimization problem, we propose to use a new risk function, the asymmetric exponential risk,

$$R_{\psi,\tau}\left(F\right) = \mathbb{E}\left(t(y,x)e^{-yF(x)}\right), \tag{21}$$

which is a convex surrogate of the score risk (13). Similarly, let us denote the optimal asymmetric exponential risk as

$$R^*_{\psi,\tau} = \inf_F R_{\psi,\tau}\left(F\right). \tag{22}$$

The asymmetric exponential risk replaces the non-convex indicator function in the risk (13) with the convex exponential function. As shown in Figure 1, the asymmetric exponential risk (21) is a convex upper bound of the risk (13).

Note that the optimal classifier from minimizing the asymmetric exponential risk (21) also uses $\Pr(y=1|x) > 1-\tau(x)$ for the classification rule as in (16). Take the derivative of

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

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

$$\frac{\partial R_{\psi,\tau}\left(F\left(x\right)\right)}{\partial F\left(x\right)} = -\tau(x)\Pr\left(y=1|x\right)e^{-F(x)} + \left(1-\tau(x)\right)\Pr\left(y=-1|x\right)e^{F(x)} = 0. \tag{23}$$

Hence,

$$F_{\tau}^{*}\left(x\right) = \frac{1}{2}\log\left[\frac{\tau(x)\Pr\left(y=1|x\right)}{\left(1-\tau(x)\right)\Pr\left(y=-1|x\right)}\right]. \tag{24}$$

Moreover, the optimal classifier $\text{sign}[F^{*}(x)]$ follows the classification rule $\Pr(y=1|x) > 1-\tau(x)$ as in (16) since $\tau(x)\Pr\left(y=1|x\right) > \left(1-\tau(x)\right)\Pr\left(y=-1|x\right)$ if $\Pr(y=1|x) > 1-\tau(x)$.

In fact, the excess misspecification probability, $R_{\tau}\left(\text{sign}[F]\right) - R_{\tau}^{*}$, is bounded from above by the excess asymmetric exponential risk, $R_{\psi,\tau}\left(F\right) - R_{\psi,\tau}^{*}$. Hence, the excess misspecification probability would go to zero as the excess asymmetric exponential risk goes to zero. Solving the convex surrogate problem $R_{\psi,\tau}$ would solve the maximum score problem R_{τ} that is widely used in decision theory such as the two-state two-action decision problem mentioned before. Therefore, we replace the non-convex risk function with a convex surrogate which could be minimized more efficiently and provide improvement with large samples and high-dimensional data.

4.3 Asymmetric AdaBoost as Newton-like Optimization

In this section, we introduce a new numerical algorithm that is able to efficiently solve the convex surrogate problem, which we call the Asymmetric AdaBoost. 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 1.

Algorithm 1 builds an additive regression model $F_{M}\left(x\right)$ via Newton-like updates for minimizing the asymmetric exponential risk (21). A detailed comparison of the Asymmetric AdaBoost algorithm and Newton-like minimization via the functional gradient descent is provided in shown below.

Algorithm 1 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. For $m = 1$ to M
   (a) Fit the classifier $f_m(x) \in \{-1, 1\}$ using weights w_i on the training data.
   (b) Compute $err_m = \mathbb{E}_w[1_{y \neq f_m(x)}]$, $c_m = \log\left(\frac{1-err_m}{err_m}\right)$.
   (c) Set $w_i \leftarrow w_i \exp[-c_m y_i f_m(x_i)]$, $i = 1, \ldots, n$, and normalize so that $\sum_{i=1}^{n} w_i = 1$.
3. Output the classifier from the sign of $F_M(x) = \sum_{m=1}^{M} c_m f_m(x)$, $\text{sign}[F_M(x)]$.

We follow the steps of Friedman et al. (2000) and start with the asymmetric exponential risk function

$$R_{\psi,\tau}(F(x)) = \mathbb{E}\left(t(y,x)\, e^{-yF(x)}\right). \tag{25}$$

First, we 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 in previous iterations $F_m(x)$. The updated classifier would be

$$F_{m+1}(x) = F_m(x) + c_{m+1} f_{m+1}(x). \tag{26}$$

The risk for the updated classifier is

$$R_{\psi,\tau}(F_m(x) + c_{m+1} f_{m+1}(x)) = \mathbb{E}\left(t(y,x)\, e^{-y(F_m(x) + c_{m+1} f_{m+1}(x))}\right). \tag{27}$$

Expand (27) w.r.t. $f_{m+1}(x)$

$$\mathbb{E}\left(t(y,x)\, e^{-y(F_m(x) + c_{m+1} f_{m+1}(x))}\right) \tag{28}$$

$$\approx \mathbb{E}\left(t(y,x)\, e^{-yF_m(x)}\left(1 - y c_{m+1} f_{m+1}(x) + \frac{y c_{m+1}^2 f_{m+1}^2(x)}{2}\right)\right) \tag{29}$$

$$= \mathbb{E}\left(t(y,x)\, e^{-yF_m(x)}\left(1 - y c_{m+1} f_{m+1}(x) + \frac{c_{m+1}^2}{2}\right)\right), \tag{30}$$

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

$$\max_f \mathbb{E}\left(e^{-yF_m(x)} t(y,x)\, y f_{m+1}(x) \,|x\right), \tag{31}$$

for any $c_{m+1} > 0$. Then we re-write the above maximization as

$$\max_f \mathbb{E}_w \left(t\left(y,x\right) y f_{m+1}\left(x\right) | x \right). \tag{32}$$

Here the notation $\mathbb{E}_w(\cdot|x)$ refers to a weighted conditional expectation, where $w \equiv w(x,y) \equiv e^{-yF_m(x)}$, and

$$\mathbb{E}_w\left(g(x,y)|x\right) := \frac{\mathbb{E}\left(w(x,y)g(x,y)|x\right)}{\mathbb{E}\left(w(x,y)|x\right)}. \tag{33}$$

We solve the maximization problem

$$\begin{aligned}
& \max_f \mathbb{E}_w \left(t\left(y,x\right) y f_{m+1}\left(x\right) | x \right) && (34)\\
= \;& P_w\left(y=1|x\right) t\left(1,x\right) f_{m+1}\left(x\right) - P_w\left(y=-1|x\right) t\left(-1,x\right) f_{m+1}\left(x\right) && (35)\\
= \;& \left[P_w\left(y=1|x\right) t\left(1,x\right) - P_w\left(y=-1|x\right) t\left(-1,x\right)\right] \; f_{m+1}\left(x\right), && (36)
\end{aligned}$$

by taking $f_{m+1}(x)$ the same sign as $P_w\left(y=1|x\right) t\left(1,x\right) - P_w\left(y=-1|x\right) t\left(-1,x\right)$. Thus,

$$f_{m+1}\left(x\right) = \begin{cases} 1, & P_w\left(y=1|x\right) t\left(1,x\right) - P_w\left(y=-1|x\right) t\left(-1,x\right) > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{37}$$

Next, we look for the optimal learning rate c_{m+1} for each iteration. The optimal learning rate is also related to the weight function $t(y,x)$. The summary of our findings is shown in Theorem 1.

Theorem 1. *The optimal learning rate c_{m+1} in Algorithm 1 is*

$$c_{m+1} = \frac{1}{2}\log\left(\frac{TP \times t\left(1,x\right) + TN \times t\left(-1,x\right)}{FN \times t\left(1,x\right) + FP \times t\left(-1,x\right)}\right) = \frac{1}{2}\log\left(\frac{1-err_{m+1}}{err_{m+1}}\right), \tag{38}$$

where $err_{m+1} = \mathbb{E}_w\left(t\left(y,x\right) \times 1_{(y \neq f_{m+1}(x))}\right)$, $P_w\left(y=1, f_{m+1}\left(x\right)=1\right)$ is the rate of true positive (TP), $P_w\left(y=-1, f_{m+1}\left(x\right)=-1\right)$ is the rate of true negative (TN), $P_w\left(y=1, f_{m+1}\left(x\right)=-1\right)$ is the rate of false negative (FN), $P_w\left(y=-1, f_{m+1}\left(x\right)=1\right)$ is the rate of false positive (FP).

Proof. See Appendix 8.1. □

When choosing the optimal learning rate, Algorithm 1 penalizes False Positive and False Negative classifications differently according to the weight function $t(y,x)$ which is related to the utilities as shown in (14). Hence, the classifier produced would maximize the utilities in the classification problem.

Remark 4. The existing symmetric AdaBoost algorithm starts with $w_i = \frac{1}{n}$ in Step 1 and the optimal learning rate does not penalize FN and FP differently.

Last, we update the current classifier and get ready for the next iteration. In the next iteration, we have

$$F_{m+1}(x) \leftarrow F_m(x) + c_{m+1} f_{m+1}(x). \tag{39}$$

Hence,

$$\begin{aligned} w_{m+1} &= e^{-yF_{m+1}(x)} && (40)\\ &= e^{-y(F_m(x)+c_{m+1}f_{m+1}(x))} && (41)\\ &= w_m \times e^{-c_{m+1}yf_{m+1}(x)}, && (42) \end{aligned}$$

is of identical form as the Newton-like functional gradient descent as shown by Friedman et al. (2000).

4.4 Component-wise Asymmetric AdaBoost

We now provide a version of the Asymmetric AdaBoost which is able to deal with high-dimensional data which we call the Component-wise Asymmetric AdaBoost. 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. 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.
      ii. Compute $err_{mj} = \sum_{i=1}^{n} w_i 1_{(y_i \neq f_{mj}(x_{ij}))}$.
      iii. Compute $c_{mj} = \frac{1}{2}\log\left(\frac{1-err_{mj}}{err_{mj}}\right)$.
   (b) Find $\hat{j}_m = \arg\min_j \sum w_i e^{-c_{mj} y_i f_{mj}(x_{ij})}$.
   (c) Set $w_i \leftarrow w_i \exp[-c_{m\hat{j}} y_i f_{m\hat{j}}\left(x_{i\hat{j}}\right)]$, $i = 1, \ldots, n$, and normalize so that $\sum_{i=1}^{n} w_i = 1$.
3. Output the classifier from the sign of $F_M(x) = \sum_{m=1}^{M} c_m f_{m\hat{j}_m}(x_{\hat{j}_m})$, sign$[F_M(x)]$.

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.

4.5 Asymmetric AdaBoost is Consistent

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 (21) is differentiable, Algorithm 2 uses functional gradient descent to minimize the asymmetric exponential risk which will produce a classifier with larger flexibility. Next, we show 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.

Theorem 2. *Let the assumptions in Bartlett and Traskin (2007) be satisfied. Then 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*

$$R_{\psi,\tau}\left(\text{sign}[F_{M_n}]\right) \to R^*_{\psi,\tau} \text{ as } n \to \infty. \tag{43}$$

Proof. It is a generalization of Bartlett and Traskin (2007) to the asymmetric exponential risk. □

Theorem 2 shows that the classifier produced by Algorithm 2 will minimize the exponential risk (21). Hence, Asymmetric AdaBoost is consistent in terms that the risk of the produced classifier

is minimized.

In addition, we generalize the convex relaxation result of Bartlett, Jordan, and McAuliffe (2006) for the asymmetric exponential risk.

Theorem 3. *If (43) holds, then* $\lim_{n\to\infty} R_\tau(\text{sign}[F_{M_n}]) = R^*_\tau$.

Proof. See Appendix 8.2. □

In other words, in addition to minimizing the exponential risk, the classifier produced by Algorithm 2 will achieve the Bayes risk (15), hence, solve the maximum score regression. Algorithm 2 is able to solve binary classification/prediction problem with state-dependent losses while maintaining the computational advantage and the flexibility of the functional form.

5 Monte Carlo

In this section, we examine the finite sample properties of the Asymmetric AdaBoost via Monte Carlo simulations and compare its performance with the Logistic Regression with LASSO-penalty. We consider the binary decision problem in Section 3 with $\tau(x) = \tau$.

5.1 DGPs

We construct the following high-dimensional DGPs where y follows Bernoulli distribution and x is high-dimensional. All the DGPs satisfy the sparsity assumption that most of the x's are completely irrelevant or have negligible influence on y.

DGP1 (Linear Logistic Models):

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

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

$$v = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p,$$

where

$$(x_1, x_2, \ldots, x_p)' \sim N(0, I_p), \quad \beta_j = 0.8^j, \ j = 1, \ldots, p$$
$$n = \{100, 1000\}, \quad p = 100.$$

DGP1 is the classical logistic model where the probability of y being 1 depends only on a single index v that is linear in x. This is the underlying model of the Logistic Regression. Hence, we would expect that Logistic Regression would be the best in DGP1. We construct DGP1 to give the most disadvantages to Asymmetric AdaBoost when comparing with Logistic Regression.

DGP2 (Quadratic Logistic Models):

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

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

$$v = \beta_2(x_1^2 - x_2^2) + \beta_3 x_3 + \cdots + \beta_p x_p,$$

where

$$(x_1, x_2, \ldots, x_p)' \sim N(0, I_p), \quad \beta_j = 0.8^j, \ j = 2, \ldots, p$$
$$n = \{100, 1000\}, \quad p = 100.$$

DGP2 is a slight deviation from the classical logistic model in the sense that the single index v in the logistic model is not linear in x_1 and x_2. We take the difference of x_1^2 and x_2^2 so that the expectation of the single index v is 0 and the unconditional probability of $y = 1$ is 0.5, i.e., the data is balanced. We will examine the performance of the Asymmetric AdaBoost with unbalanced data in DGP4. Note that in the simulations, we provide the two methods with x of only the first order. Since the Asymmetric AdaBoost does not depend on any parametric assumptions, we would like to check the robustness of the Asymmetric AdaBoost and the sensitivity of the Logistic Regression when the model is slightly misspecified.

DGP3 (Cubic Logistic Models):

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

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

$$v = x_1^3 - 4x_1,$$

where

$$(x_1, x_2, \ldots, x_p)' \sim N(0, I_p), \quad n = \{100, 1000\}, \quad p = 100.$$

In DGP3, we deviate further from the classical logistic model by having the single-index v to be a third-order polynomial of x_1. DGP3 is to test the performance of the Asymmetric AdaBoost and the Logistic Regression when the parametric assumptions of the Logistic Regression are invalid.

DGP4 (Circle Model, Mease, Wyner, and Buja (2007)):

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

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

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

where

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

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

The probability, $\Pr(y=1|x)$, in the DGP4 is shown in Figure 2. A major difference between DGP4 and the other DGPs is that $\Pr(y=1) \approx 0.1 < 0.5$ in DGP4. 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. recessions over expansions.

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

$$y = \begin{cases} 1 & \Pr(y=1|x) > \epsilon \\ -1 & \Pr(y=1|x) < \epsilon. \end{cases} \tag{44}$$

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 the 0-1 risk of the tested methods,

$$\hat{R}_{\tau,n'}(\mathrm{sign}[F]) = \frac{\tau}{n'} \sum_{y_i=1} 1_{(y_i \neq \mathrm{sign}[F(x_i)])} + \frac{(1-\tau)}{n'} \sum_{y_i=-1} 1_{(y_i \neq \mathrm{sign}[F(x_i)])}. \tag{45}$$

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

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

The above procedure is repeated for 1000 times and the average over the 1000 repetitions is reported in the tables.

5.2 Alternative Method: Asymmetric Logistic Regression

Apart from Asymmetric AdaBoost, we consider the 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

$$\beta = \arg\min_{\beta} - \sum_{i=1}^{n} \left[Y_i \left(x_i \beta \right) - \log \left(1 + e^{x_i \beta} \right) \right] + \lambda \left| \beta \right|_1 . \tag{46}$$

In particular, we use the standard *glmnet* package of Friedman et al. (2010) for the Logistic Regression. 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 (16).

5.3 Results

The simulation results are reported in Tables 1 to 4. In Table 1, the DGP1 is a linear logistic model. In this case, the Logistic Regression has absolute 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 which is infeasible in practice. However, even in this case, we see that the advantage of the Logistic Regression over the Asymmetric AdaBoost is limited and as the sample size increases, the loss of the Asymmetric AdaBoost converges to the sample Bayes risk which suggests that the Asymmetric AdaBoost is consistent.

In Table 2, the DGP2 is still the logit model. Hence, the Logistic Regression still has inherited advantages over the Asymmetric AdaBoost. However, we introduce a small deviation from DGP1 by letting the single index, v, in the logistic function be quadratic in x_1 and x_2. In this case, the logistic regression is partially 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. This is expected as our method is nonparametric and nonparametric methods generally perform worse in the tails when samples in the area are few. Moreover, Logistic Regression is also not highly biased. 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.

When the sample size increases, the Asymmetric AdaBoost have smaller variance and the losses are closer to the sample Bayes risk. The Logistic Regression, on the other hand, is still biased and has higher losses than the Asymmetric AdaBoost except in the two far 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 even if the true model only deviates slightly from the parametric assumptions of the Logistic Regression.

In Table 3, the DGP3 deviates further from the classical logistic model. The Asymmetric AdaBoost performs strictly better than the Logistic Regression. When n is small, we see that the Asymmetric AdaBoost outperforms the Logistic Regression except in the two tails ($\tau = 0.1$ and $\tau = 0.9$) where insufficient samples are available. This is a general limitation of all nonparametric methods since nonparametric methods. However, the performance of the Asymmetric AdaBoost surpasses the Logistic Regression in the tails when the sample size become larger. In practice, when the true DGP is not the logistic model, the Asymmetric AdaBoost is definitely more reliable.

In Table 4, the DGP4 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, the 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 as we will see in the application section. 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 is more robust if the true DGP is not the logistic model especially when 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, such as recessions over expansions, when the data is unbalanced.

6 Application

In this section, we predict the NBER business cycle turning points using both the Asymmetric AdaBoost and the Logistic Regression with LASSO-penalty. We use the 132 independent variables from the data of Jurado, Ludvigson, and Ng (2015). After removing the observations with missing values and taking the one, two and three lagged values of each independent variable and the dependent variable, the remained sample period ranges from April 1964 to July 2011 with 568 observations and 399 independent variables. We use a rolling sample scheme and make three-period ahead predictions of economic recessions as in Ng (2014). We use rolling sample size (n) of 60, 120 and 240. The average losses from all rolling samples under different degrees of asymmetry

$$\hat{R}_{\tau,n}\left(\text{sign}[F]\right) = \frac{\tau}{n'}\sum_{y_i=1} 1_{(y_i \neq \text{sign}[F(x_i)])} + \frac{(1-\tau)}{n'}\sum_{y_i=-1} 1_{(y_i \neq \text{sign}[F(x_i)])}$$

where n' is the total number of rolling samples are reported in Table 5.

In the application, we see that the Asymmetric AdaBoost has smaller losses than the Logistic Regression. Both the Asymmetric AdaBoost and the Logistic Regression are consistent in the sense that the forecasting error decreases as the rolling sample size increases. Algorithm-wise, we have removed the rolling samples that contain less than two months of recessions. These samples account for 119, 2, 0 of the total rolling samples in the cases where the rolling sample sizes are 60, 120 and 240. When the number of recessions in the rolling sample is less than two, the standard package for Logistic Regression with LASSO-penalty reports an error and fails to produce the result(Friedman et al., 2010). More specifically, if there is no recession contained in the rolling sample, the maximum likelihood of the Logistic Regression would be 0, i.e., the coefficients of the Logistic Regression

would explode to infinity. In addition, we can not use cross-validation to choose the penalty term λ for Logistic Regression since the cross-validation process involves randomly resampling the rolling samples and frequently results in less than two recessions in the cross-validation samples even when $n = 240$. Instead, we tried different values of λ for Logistic Regression and reported all of them in Table 5. In almost all cases, the Asymmetric AdaBoost significantly outperforms the Logistic Regression which strongly suggests that the parametric assumptions of Logistic Regression are invalid in this application.

7 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 and application results show that Asymmetric AdaBoost is a competitive approach in binary classification/prediction.

8 Appendix

8.1 Proof of Theorem 1

After solving $f_{m+1}(x)$, we minimize the risk function (27) w.r.t. c_{m+1},

$$\begin{aligned} c_{m+1} &= \arg\min_{c} R_{\psi,\tau}\left(F_m(x) + c f_{m+1 m+1}(x)\right) && (47)\\ &= \arg\min_{c} \mathbb{E}\left(t(y,x)\, e^{-y(F_m(x)+c_{m+1}f_{m+1}(x))}\right) && (48)\\ &= \arg\min_{c} \mathbb{E}_w\left(t(y,x)\, e^{-y c_{m+1} f_{m+1}(x)}\right) && (49)\end{aligned}$$

Then

$$\begin{aligned} & \mathbb{E}_w\left(t(y,x)\, e^{-y c_{m+1} f_{m+1}(x)}\right) && (50)\\ = \;& P_w(y=1, f_{m+1}(x)=1)\, t(1,x)\, e^{-c_{m+1}} + P_w(y=-1, f_{m+1}(x)=-1)\, t(-1,x)\, e^{-c_{m+1}} && (51)\\ & + P_w(y=1, f_{m+1}(x)=-1)\, t(1,x)\, e^{c_{m+1}} + P_w(y=-1, f_{m+1}(x)=1)\, t(-1,x)\, e^{c_{m+1}} && (52)\end{aligned}$$

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

$$\begin{aligned} & \frac{\partial R_{\psi,\tau}\left(c_{m+1} f_{m+1}(x)\right)}{\partial c_{m+1}} && (53)\\ = \;& -P_w(y=1, f_{m+1}(x)=1)\, t(1,x)\, e^{-c_{m+1}} - P_w(y=-1, f_{m+1}(x)=-1)\, t(-1,x)\, e^{-c_{m+1}} && (54)\\ & + P_w(y=1, f_{m+1}(x)=-1)\, t(1,x)\, e^{c_{m+1}} + P_w(y=-1, f_{m+1}(x)=1)\, t(-1,x)\, e^{c_{m+1}} && (55)\end{aligned}$$

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

$$\begin{aligned} & P_w(y=1, f_{m+1}(x)=1)\, t(1,x)\, e^{-c_{m+1}} + P_w(y=-1, f_{m+1}(x)=-1)\, t(-1,x)\, e^{-c_{m+1}} && (56)\\ = \;& P_w(y=1, f_{m+1}(x)=-1)\, t(1,x)\, e^{c_{m+1}} + P_w(y=-1, f_{m+1}(x)=1)\, t(-1,x)\, e^{c_{m+1}}, && (57)\end{aligned}$$

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

$$\left[\mathrm{TP} \times t(1,x) + \mathrm{TN} \times t(-1,x)\right] e^{-c_{m+1}} = \left[\mathrm{FN} \times t(1,x) + \mathrm{FP} \times t(-1,x)\right] e^{c_{m+1}}, \quad (58)$$

we obtain the optimal c_{m+1}

$$c_{m+1} = \frac{1}{2}\log\left(\frac{\mathrm{TP}\times t(1,x) + \mathrm{TN} \times t(-1,x)}{\mathrm{FN} \times t(1,x) + \mathrm{FP} \times t(-1,x)}\right) = \frac{1}{2}\log\left(\frac{1-err_{m+1}}{err_{m+1}}\right), \quad (59)$$

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

8.2 Proof of Theorem 3

Notation. Let

$$R(G) = \mathbb{E}\left[\frac{1}{2} \times 1_{(y \neq G(x))}\right] = \frac{1}{2}\Pr(y \neq G(x)), \tag{60}$$

denote the 0-1 risk when $\tau = \frac{1}{2}$ and

$$R^* = \inf_G R(G) = \mathbb{E}\min\left\{\frac{1}{2}\Pr\left(y = 1|x\right), \frac{1}{2}\Pr\left(y = -1|x\right)\right\}, \tag{61}$$

be the minimum risk. Let

$$R_\psi\left(F\right) = \mathbb{E}\left(\frac{1}{2}e^{-yF(x)}\right), \tag{62}$$

be the exponential risk with $t(y, x) = \frac{1}{2}$ and

$$R_\psi^* = \inf_F R_\psi\left(F\right). \tag{63}$$

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

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

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

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

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

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

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

Hence, by Lemma 1,

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

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

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

Therefore,

$$R_\tau(\text{sign}[F_i]) = \int t(y,x)\, 1_{(y \neq \text{sign}[F_i])} \mathcal{P}(x,y)\, dydx \to \int t(y,x)\, 1_{(y \neq \text{sign}[F^*])} \mathcal{P}(x,y)\, dydx = R^*_\tau.$$

The statement in the theorem is proved. □

References

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. Technical report.

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.

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

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software 33*(1), 1–22.

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

Ichimura, H. (1993). Semiparametric Least Squares and Weighted SLS Estimation of Single Index Models. *Journal of Econometrics 58*(1-2), 71–120.

Jurado, K., S. C. Ludvigson, and S. Ng (2015). Measuring Uncertainty. *American Economic Review 105*(3), 1177–1216.

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

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.

Mease, D., A. Wyner, and A. Buja (2007). Cost-weighted boosting with jittering and over/under-sampling: Jous-boost. *Journal of Machine Learning Research 8*, 409–439.

Ng, S. (2014). Viewpoint: Boosting recessions. *Canadian Journal of Economics 47*(1), 1–34.

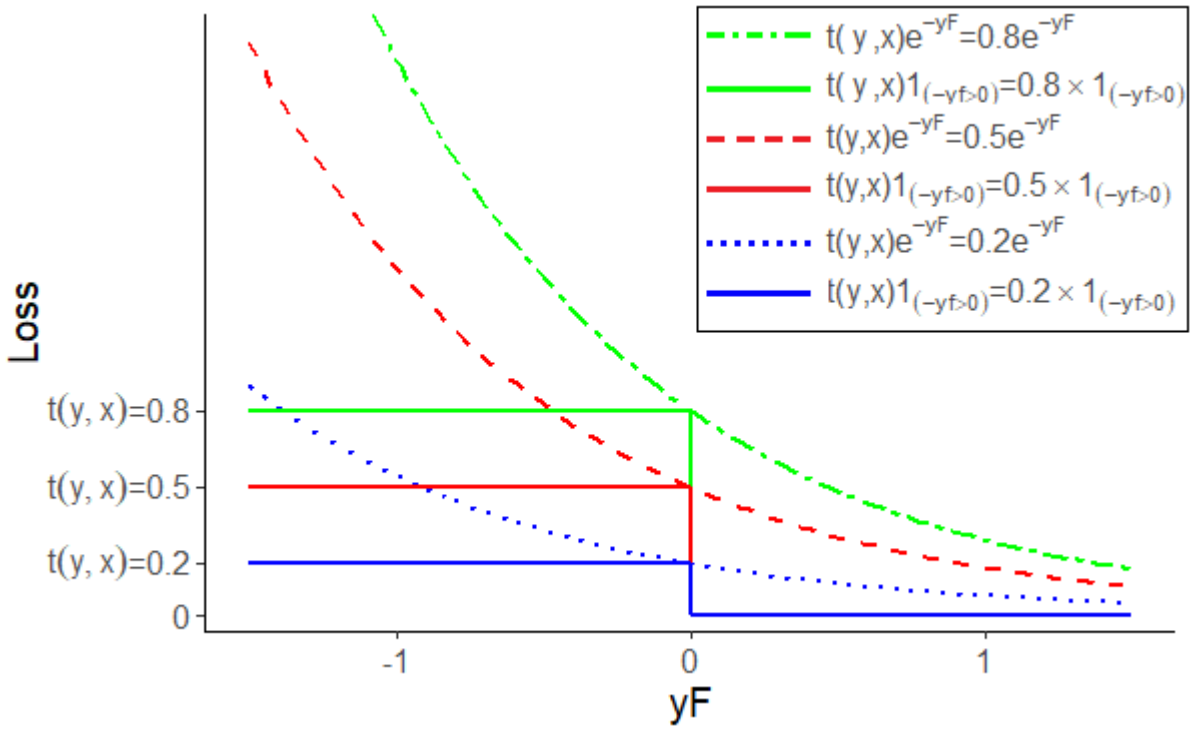

Figure 1: (Asymmetric) Exponential Loss and 0-1 Loss

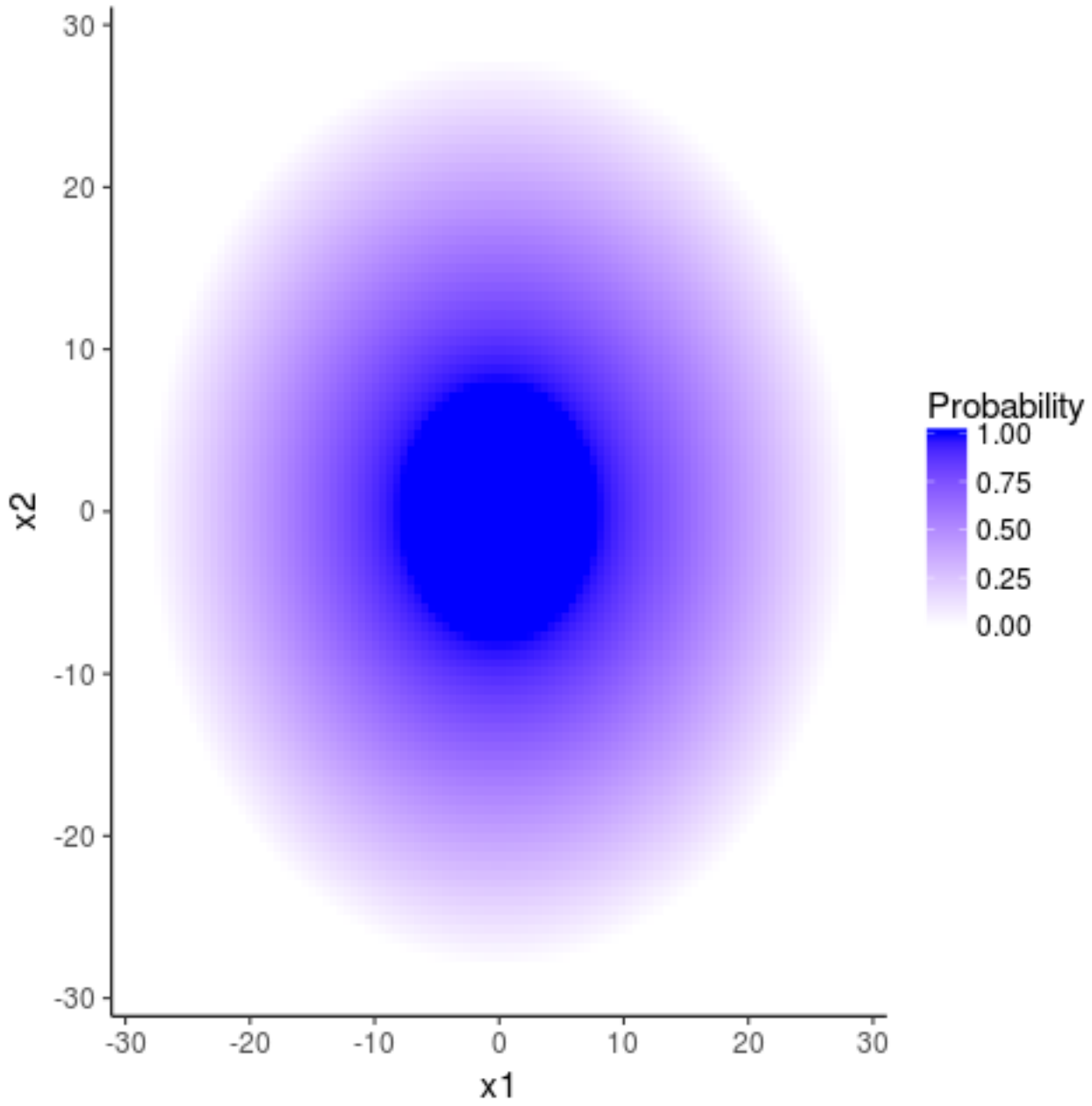

Figure 2: Conditional Probability of the Circle Model

Table 1: Linear Logit Model (DGP1)

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$n = 1000$	AdaBoost	0.0544	0.0997	0.1379	0.1602	0.1712	0.1607	0.1372	0.1005	0.0545
	LASSO	0.0492	0.0934	0.1266	0.1479	0.1550	0.1482	0.1271	0.0933	0.0493
	Bayes Risk	0.0482	0.0885	0.1178	0.1360	0.1419	0.1359	0.1182	0.0886	0.0482
$n = 100$	AdaBoost	0.0774	0.1263	0.1728	0.2001	0.2085	0.1981	0.1739	0.1300	0.0773
	LASSO	0.0509	0.1026	0.1483	0.1814	0.1973	0.1843	0.1482	0.1015	0.0513
	Bayes Risk	0.0482	0.0885	0.1180	0.1357	0.1418	0.1358	0.1179	0.0885	0.0483

Note: The average of the losses of the two methods for predicting y are reported in the table. Bayes Risk is the infeasible optimal risk when the true model is known. τ shows different degrees of asymmetry. n is the sample size of each training sample.

Table 2: Balanced Quadratic Logit Model (DGP2)

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$n = 1000$	AdaBoost	0.0524	0.0958	0.1330	0.1614	0.1736	0.1570	0.1316	0.0951	0.0516
	LASSO	0.0510	0.1021	0.1495	0.1841	0.1949	0.1805	0.1442	0.0977	0.0488
	Bayes Risk	0.0469	0.0866	0.1168	0.1358	0.1422	0.1358	0.1170	0.0867	0.0469
$n = 100$	AdaBoost	0.0765	0.1304	0.1734	0.2017	0.2121	0.2049	0.1769	0.1335	0.0819
	LASSO	0.0501	0.1017	0.1552	0.2076	0.2346	0.2063	0.1541	0.1030	0.0514
	Bayes Risk	0.0468	0.0866	0.1168	0.1357	0.1422	0.1358	0.1168	0.0866	0.0469

Note: The average of the losses of the two methods for predicting y are reported in the table. Bayes Risk is the infeasible optimal risk when the true model is known. τ shows different degrees of asymmetry. n is the sample size of each training sample.

Table 3: Unbalanced Quadratic Logit Model (DGP3)

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$n = 1000$	AdaBoost	0.0442	0.0642	0.0770	0.0837	0.0857	0.0837	0.0771	0.0641	0.0443
	LASSO	0.0500	0.0999	0.1499	0.1999	0.2499	0.1998	0.1499	0.1000	0.0500
	Bayes Risk	0.0402	0.0609	0.0736	0.0807	0.0830	0.0807	0.0736	0.0609	0.0402
$n = 100$	AdaBoost	0.0539	0.0843	0.1148	0.1393	0.1393	0.1392	0.1162	0.0843	0.0535
	LASSO	0.0500	0.0999	0.1500	0.2015	0.2498	0.2023	0.1501	0.0999	0.0500
	Bayes Risk	0.0403	0.0609	0.0736	0.0807	0.0830	0.0807	0.0737	0.0609	0.0402

Note: The average of the losses of the two methods for predicting y are reported in the table. Bayes Risk is the infeasible optimal risk when the true model is known. τ shows different degrees of asymmetry. n is the sample size of each training sample.

Table 4: Circle Model (DGP4)

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$n = 1000$	AdaBoost	0.0402	0.0719	0.0835	0.0893	0.0981	0.1041	0.1058	0.0794	0.0443
	LASSO	0.0358	0.0715	0.1073	0.1430	0.1792	0.2158	0.1937	0.1283	0.0641
	Bayes Risk	0.0276	0.0513	0.0700	0.0833	0.0902	0.0897	0.0814	0.0640	0.0372
$n = 100$	AdaBoost	0.0554	0.0841	0.1090	0.1256	0.1353	0.1387	0.1336	0.1189	0.0807
	LASSO	0.0358	0.0718	0.1082	0.1451	0.1848	0.2272	0.2049	0.1344	0.0658
	Bayes Risk	0.0276	0.0512	0.0700	0.0834	0.0902	0.0897	0.0812	0.0640	0.0372

Note: The average of the losses of the two methods for predicting y are reported in the table. Bayes Risk is the infeasible optimal risk when the true model is known. τ shows different degrees of asymmetry. n is the sample size of each training sample.

Table 5: Loss for Predicting Recessions

	τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$n = 60$	AdaBoost	0.0181	0.0255	0.0318	0.0377	0.0374	0.0405	0.0431	0.0362	0.0348
	LASSO ($\lambda = 0.05$)	0.0229	0.0350	0.0452	0.0499	0.0514	0.0473	0.0422	0.0365	0.0288
	LASSO ($\lambda = 0.1$)	0.0216	0.0370	0.0522	0.0535	0.0488	0.0473	0.0458	0.0411	0.0440
	LASSO ($\lambda = 1$)	0.0211	0.0422	0.0632	0.0843	0.1054	0.1326	0.1707	0.1769	0.1262
	LASSO ($\lambda = 5$)	0.0211	0.0422	0.0632	0.0843	0.1054	0.1326	0.1707	0.1769	0.1262
$n = 120$	AdaBoost	0.0154	0.0245	0.0267	0.0316	0.0334	0.0352	0.0348	0.0276	0.0223
	LASSO ($\lambda = 0.05$)	0.0182	0.0314	0.0370	0.0381	0.0381	0.1641	0.1269	0.0919	0.0511
	LASSO ($\lambda = 0.1$)	0.0168	0.0336	0.0469	0.0444	0.0471	0.0457	0.0365	0.0359	0.0332
	LASSO ($\lambda = 1$)	0.0155	0.0309	0.0464	0.0619	0.0774	0.0928	0.1128	0.1439	0.0659
	LASSO ($\lambda = 5$)	0.0155	0.0309	0.0464	0.0619	0.0774	0.0928	0.1128	0.1439	0.0659
$n = 240$	AdaBoost	0.0115	0.0158	0.0228	0.0237	0.0289	0.0256	0.0271	0.0231	0.0170
	LASSO ($\lambda = 0.05$)	0.0112	0.0213	0.0295	0.0402	0.0442	0.1121	0.0871	0.0621	0.0378
	LASSO ($\lambda = 0.1$)	0.0112	0.0225	0.0338	0.0426	0.0472	0.0408	0.0274	0.0274	0.0320
	LASSO ($\lambda = 1$)	0.0112	0.0225	0.0338	0.0451	0.0564	0.0676	0.0789	0.1500	0.0853
	LASSO ($\lambda = 5$)	0.0112	0.0225	0.0338	0.0451	0.0564	0.0676	0.0789	0.1500	0.0853

Note: The average losses of the three period ahead prediction on recessions are reported in the table. τ shows different degrees of asymmetry.