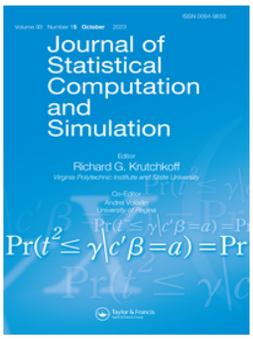

Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/gscs20

# Robust logistic regression with shift parameter estimation

Bokyoung Shin & Seokho Lee

**To cite this article:** Bokyoung Shin & Seokho Lee (2023) Robust logistic regression with shift parameter estimation, Journal of Statistical Computation and Simulation, 93:15, 2625-2641, DOI: 10.1080/00949655.2023.2201008

**To link to this article:** https://doi.org/10.1080/00949655.2023.2201008

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
2023, VOL. 93, NO. 15, 2625–2641
https://doi.org/10.1080/00949655.2023.2201008



# Robust logistic regression with shift parameter estimation

Bokyoung Shin and Seokho Lee

Department of Statistics, Hankuk University of Foreign Studies, Yongin, Republic of Korea

**ABSTRACT**

We investigate a shift parameter approach to logistic regression for robust classification. Shift parameter moves margin to the minimum of loss function. For robust estimation, margin-based logistic regression requires its own version of thresholding-type estimate which is different from residual-based regression. We discuss shift parameter estimation desirable to robust classification and propose some penalty functions producing such shift parameter estimates. Comparing to existing robust logistic regression methods requiring non-convex optimization or label transition modelling, our proposal is implemented in a simple alternating optimization: the classifier is obtained as a solution of conventional logistic regression with an offset and shift parameter is individually estimated in a closed form. We discuss some robust properties of the method and demonstrate its performance in linear and nonlinear classification with synthetic and real-world examples.

**ARTICLE HISTORY**
Received 3 July 2022
Accepted 3 April 2023

**KEYWORDS**
Label noise; logistic regression; outlier; robust classification; shift parameter

## 1. Introduction and motivation

Consider binary classification problem with class label $y_i \in \{-1, +1\}$ and covariate $\mathbf{x}_i \in \mathbb{R}^p$ for $i = 1, \ldots, n$. Logistic regression aims to find a classifier $f(\mathbf{x})$ by minimizing the empirical risk

$$R(f) = \sum_{i=1}^{n} \log(1 + \exp(-y_i f(\mathbf{x}_i))). \tag{1}$$

This comes from negative log likelihood, assuming $(y_i + 1)/2$ follows Bernoulli distribution with a success probability $1/(1 + \exp(-f(\mathbf{x}_i)))$ . After estimating the classifier, $\hat{f}$, the associated decision rule is defined as $\hat{y} = \text{sign}(\hat{f}(\mathbf{x}))$. The hyper-surface $\{\mathbf{x} : f(\mathbf{x}) = 0\}$ is called decision boundary. Note that $yf(\mathbf{x}) > 0$ implies correct classification and $yf(\mathbf{x}) < 0$ indicates misclassification. The term $u = yf(x)$ is called margin [1]. As the residual shows how badly an observation follows the mean response in regression problem, the margin is an indicator whether or not the individual observation is outlying according to $f$ in classification. An observation located far from their own class will have a large negative margin. Margin-based classification is performed with a loss function that promotes positive margin (correct classification) and prevents negative margin (misclassification). To meet this

**CONTACT** Seokho Lee lees@hufs.ac.kr

purpose, loss functions are often chosen to be non-increasing in margin $u$. Logistic loss $\ell(u) = \log(1 + \exp(-u))$ in (1) is one of the popular choices.

The classifier from minimizing (1), with logistic loss or other non-increasing loss functions, can be affected by the existence of outliers because loss functions used in conventional margin-based classification are unbounded so that severe outliers will have large negative margins under the correct decision boundary. Thus, most robust classification methods prefer bounded loss functions to reduce unwanted effects from outliers [2,3]. Since boundedness of non-increasing loss requires non-convexity, its optimization is not straightforward and needs specialized algorithms. Another approach to handle outliers is to treat them as mislabelled observations, called label noise, and incorporates stochastic label noise mechanism into data-generating process model [4,5]. This approach requires a specific label transition probability model which, however, is often unknown in practice.

As a different way from bounded loss approach and label noise approach, adding shift parameter (individual intercept, case-specific parameter) to classifier is known to make classification procedure robust to outliers [6–8]. In this approach, the objective function to be minimized is

$$R(f, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \log(1 + \exp\{-y_i f(\mathbf{x}_i) + \gamma_i\}) + \sum_{i=1}^{n} p_\lambda(\gamma_i). \tag{2}$$

Here, $\gamma_i$ are shift parameters to be individually estimated for each observation. The penalty on $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)^T$ promotes sparsity so that a small portion of $\gamma_i$ is estimated nonzero. The sparsity of $\boldsymbol{\gamma}$ naturally motivates using $L_1$ penalty, $p_\lambda(x) = \lambda|x|$. References [7,8] show that $L_1$ penalization yields the estimator of $\gamma_i$ in the form of on margin. For robust classification with functional data, Ref. [6] additionally considers other sparsity-inducing penalizations, such as SCAD (Smoothly clipped absolute deviation) [9] and MCP (Minimax concave penalty) [10]. The individual shift parameters help to reduce the loss, adjusting the large negative margin by subtracting it by a negative intercept when the observation is misclassified toward the opposite class. In the case of observations classified correctly, the corresponding shift parameters are estimated as zero, so no adjustment is applied. Thus, $\ell(u_i)$ is shifted by $\gamma_i$ and $\ell(u_i - \gamma_i)$ is smaller than $\ell(u_i)$ if $u_i = y_i f(\mathbf{x}_i)$ is large negative so that $\gamma_i < 0$. This mechanism makes the classification procedure robust by preventing loss function being too high for possible outliers in training data. This shift parameter approach enjoys computational advantage over aforementioned approaches. Alternating algorithm can be applied to find $f$ and $\boldsymbol{\gamma}$ simultaneously. Given $\boldsymbol{\gamma}$, $f$ is obtained from standard logistic regression with an offset value $-y_i\gamma_i$ since $y_i f(\mathbf{x}_i) - \gamma_i = y_i\{f(\mathbf{x}_i) - y_i\gamma_i\}$ with $y_i^2 = 1$. Once $f$ is obtained, the estimation of $\gamma_i$ is separable so that $\gamma_i$ can be individually obtained as a closed-form solution [6,7].

The idea that individual shift parameter leads to robust estimation is motivated from wavelet denoising by wavelet thresholding [11,12]. Applying it to robust regression, Ref. [12] proposes the below optimization for robust estimation and outlier detection in regression

$$\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i) - \gamma_i)^2 + \sum_{i=1}^{n} p_\lambda(\gamma_i). \tag{3}$$

Note that the squared loss $\ell(u) = u^2$ has individual parameter $\gamma_i$ with the residual $u_i = y_i - f(\mathbf{x}_i)$, so that the original loss $\ell(u_i)$ becomes shifted as $\ell(u_i - \gamma_i)$. Reference [12] shows that imposing $L_1$ penalty on $\boldsymbol{\gamma}$ leads to soft thresholding solution as $\hat{\gamma}_i = \Theta_{\text{soft}}(u_i; \lambda) = \text{sign}(u_i)(|u_i| - \lambda)_+$. And the resulting regression coefficient estimate becomes the robust estimate from Huber's procedure, which is also found independently by Ref. [7]. Reference [12] recommends $L_0$ penalty on $\boldsymbol{\gamma}$ for improved robustness, with which the individual parameter is estimated as hard thresholding $\hat{\gamma}_i = \Theta_{\text{hard}}(u_i; \lambda) = u_i I(|u_i| > \lambda)$. Both approaches adjust the residual by individual parameter in order to move large residuals toward 0 at which the squared loss achieves its minimum. This mechanism reduces the harmful effects on estimation of $f$ from severe outliers and achieves robustness in regression.

In the case of logistic regression with individual parameter in (2), logistic loss $\ell(u) = \log(1 + e^{-u})$ of margin $u = yf(\mathbf{x})$ has individual parameter $\gamma_i$ which makes the original loss $\ell(u_i)$ become shifted as $\ell(u_i - \gamma_i)$ as in regression. Thus, we expect adjusting margin by shift parameter and makes the estimation robust in logistic regression as in least-squares-based regression. For logistic regression, however, there exist some notable features different from regression.

(1) The shift parameter obtained in logistic regression is not same as that in regression even if the same penalty function is applied. Shift parameter optimization depends not only on the choice of penalty function but also on the loss function used in the procedure. Thus, logistic regression requires its own penalty function for soft or hard thresholding on margin.
(2) The shift parameter of thresholding residual in regression is not helpful for robust estimation in margin-based logistic regression. In regression, excessively large residuals are shifted toward 0 by subtracting the shift parameter from them so that the squared loss becomes minimized for outliers. On the other hand, logistic loss takes its minimum at $+\infty$, so that the large negative margin appeared in severe misclassified observations should be moved further than 0 toward $+\infty$ in order to achieve improved robustness.

The rest of this paper is organized as follows. In Section 2, we develop and propose a set of penalty functions tailored to robust logistic regression. The resulting shift parameters for margin adjustment will push the excessive negative margin toward $+\infty$, rather than shrinking residuals to 0 in regression. Section 3 provides its implementation. The robustness and prediction performance of the proposed methods is demonstrated in Section 4. We discuss potential extensions in Section 5.

## 2. Methodology

Shifting loss is beneficial for robustness when shift parameters are properly given, as discussed in the previous section. The use of alternating estimation is a reasonable choice for the minimization of (2) over $f$ and $\boldsymbol{\gamma}$. Given $f$ in the alternating estimation procedure, shift parameter $\boldsymbol{\gamma}$ is estimated under penalized loss minimization. In this section, we provide some penalty functions on shift parameter with which its estimate is suitable for robust

estimation. Before doing it, we discuss some desirable properties for shift parameter in logistic regression.

### 2.1. Thresholding on margin

Once the classifier $f$ is obtained at the current step of iterative procedure, the shift parameters are updated by minimizing (2) based on the margins $u_i = y_i f(\mathbf{x}_i)$. Note that since shift parameter estimation is separable, they are individually optimized by minimizing

$$\hat{\gamma}_i = \arg\min_r \log(1 + e^{-u_i + r}) + p_\lambda(r)$$

for each $i = 1, \ldots, n$. A desirable $\hat{\gamma}_i$ satisfies

- $\hat{\gamma}_i = 0$ for a positive margin or small negative margin. This implies that observations classified correctly or misclassified mildly (misclassified but located near the decision boundary) do not necessarily require adjustment for their margins and
- $u_i - \hat{\gamma}_i > 0$ for a gross negative margin. This implies that the adjusted margin becomes positive, or can be $+\infty$ if necessary, so that the misclassified observations under the current classifier would have the reduced impact on the classifier estimation in the next round.

With these considerations, we develop a penalty function $p_\lambda(r)$ that produces the associated estimate $\hat{\gamma}_i$ satisfying $\hat{\gamma}_i = 0$ for $u_i > -\lambda$ and $\hat{\gamma}_i < 0$ for $u_i \leq -\lambda$ for a given $\lambda > 0$. Recall the robust regression of (3), where the individual shift parameters are given in the form of thresholding function of residuals. Motivated from robust regression, we propose margin thresholding in robust logistic classification. Contrast to residual thresholding, thresholding of margin $u_i = y_i f(\mathbf{x}_i)$ must be active (i.e. $\hat{\gamma}_i < 0$) when $u_i \leq -\lambda$ and inactive (i.e. $\hat{\gamma}_i = 0$) when $u_i > -\lambda$. Moreover, we want to make $u_i - \hat{\gamma}_i$ larger than 0 toward $+\infty$. To meet these requirements, our suggestion is of the form $\hat{\gamma}_i = a \min\{\Theta(u; \lambda), 0\}$ with $a > 1$. Here, $\Theta(u; \lambda)$ is any thresholding function on $u$ with a threshold $\lambda$, that is used in wavelet thresholding. And $a$ is a multiplicative factor which pushes the adjusted margin toward $+\infty$.

The following theorems introduce special forms of penalty function to produce shift parameter estimates associated with soft and hard thresholdings.

**Theorem 2.1 (Soft thresholding for margin):** *Let* $g_u(r) = \log(1 + e^{-u+r}) + p_\lambda(r)$ *with*

$$p_\lambda(r) = \frac{a}{a-1}\log(1 + e^{\lambda}) - \frac{a}{a-1}\log\left(1 + e^{-((a-1)/a)|r|+\lambda}\right),$$

*where* $\lambda > 0$ *and* $1 < a < \infty$. *Then, for any given u, the minimizer of* $g_u(r)$, *say* $\hat{r}$, *satisfies*

(a) $\hat{r} \leq 0$ *for any u.*
(b) $\hat{r}$ *is given as*

$$\hat{r} = a \min\{\Theta_{\text{soft}}(u; \lambda), 0\} = \min\{a(u + \lambda), 0\} = \begin{cases} a(u + \lambda), & u \leq -\lambda, \\ 0, & u > -\lambda. \end{cases}$$

***Proof:*** $g_u(r)$ is a continuous function defined on $r \in \mathbb{R}$ since $p_\lambda(r)$ is continuous. Note that for $r > 0$, its derivative is $g_u'(r) = (1/(1+e^{u-r})) + (1/(1+e^{((a-1)/a)r-\lambda})) > 0$ , implying that $g_u(r)$ is strictly increasing over $r \in (0,\infty)$. Thus, a minimizer of $g_u(r)$ cannot be positive. For $r \le 0$, the stationary point satisfying $g_u'(r^*) = (1/(1+e^{u-r^*})) - (1/(1+e^{-((a-1)/a)r^*-\lambda})) = 0$ is $r^* = a(u+\lambda)$. If $u > -\lambda$, then $r^* > 0$ so that $g_u'(r) < 0$ for all $r < 0$. Since $g_u(r)$ is strictly decreasing in $r \le 0$ and strictly increasing in $r > 0$, we get $\hat{r} = 0$ for $u > -\lambda$. On the other hand, if $u \le -\lambda$, then $g_u(r)$ is decreasing in $-\infty < r < r^*$ and increasing in $r^* < r < 0$. Thus $\hat{r} = r^* = a(u+\lambda)$. This completes the proof. ■

**Corollary 2.2:** *The soft thresholding for margin in Theorem* 2.1 *has the below special cases:*

(1) *If* $a \to +\infty$, *then*

$$\lim_{a\to+\infty} p_\lambda(r) = \log(1+e^{\lambda}) - \log(1+e^{-|r|+\lambda}), \quad \hat{r} = \begin{cases} -\infty, & u \le -\lambda, \\ 0, & u > -\lambda. \end{cases}$$

(2) *If* $a \to 1$, *then*

$$\lim_{a\to 1} p_\lambda(r) = \frac{e^{\lambda}}{1+e^{\lambda}}|r|, \quad \hat{r} = \begin{cases} u+\lambda, & u \le -\lambda, \\ 0, & u > -\lambda. \end{cases}$$

**Theorem 2.3 (Hard thresholding for margin):** *Let* $g_u(r) = \log(1+e^{-u+r}) + p_\lambda(r)$ *with*

$$p_\lambda(r) = \begin{cases} \log(1+e^{\lambda}) - \log\left(1+e^{-|r|+\lambda}\right), & 0 < |r| < a\lambda, \\ \log(1+e^{\lambda}) + \frac{1}{a-1}\log(1+e^{-(a-1)\lambda}) - \frac{a}{a-1} & \\ \quad \times \log\left(1+e^{-((a-1)/a)|r|}\right), & |r| \ge a\lambda, \end{cases}$$

*where* $\lambda > 0$ *and* $1 < a < \infty$. *Then, for any given u, the minimizer of* $g_u(r)$, *say* $\hat{r}$, *satisfies*

(a) $\hat{r} \le 0$ *for any u.*
(b) $\hat{r}$ *is given as*

$$\hat{r} = a\min\{\Theta_{\text{hard}}(u;\lambda), 0\} = auI(u \le -\lambda) = \begin{cases} au, & u \le -\lambda, \\ 0, & u > -\lambda. \end{cases}$$

***Proof:*** The proof is similar as Theorem 1. Note that $g_u'(r) = (1/(1+e^{u-r})) + (1/(1+e^{r-\lambda})) > 0$ for $0 < r < a\lambda$ and $g_u'(r) = (1/(1+e^{u-r})) + 1/(1+e^{((a-1)/a)r}) > 0$ for $r > a\lambda$. Since $g_u'(r) > 0$ for $r > 0$, $g_u(r)$ is strictly increasing, implying that $g_u(r)$ does not have a minimum at positive domain. Now, consider $r \le 0$. Note that

$$g_u'(r) = \begin{cases} \frac{1}{1+e^{u-r}} - \frac{1}{1+e^{-r-\lambda}}, & -a\lambda < r < 0, \\ \frac{1}{1+e^{u-r}} - \frac{1}{1+e^{-\frac{a-1}{a}r}}, & r \le -a\lambda. \end{cases}$$

Consider $u > -\lambda$. Since $u - r > -r - \lambda$ for a negative $r$, we get $(1/(1+e^{u-r})) < (1/(1+e^{-r-\lambda}))$ or $g_u'(r) < 0$ for $-a\lambda < r < 0$. For $r \le -a\lambda$, the fact $u - r > -r - \lambda \ge$

$-r + (r/a) = -((a-1)/a)r$ leads to $(1/(1+e^{u-r})) < (1/(1+e^{-((a-1)/a)r}))$ , equivalently, $g_u'(r) < 0$. Thus, $g_u(r)$ is strictly decreasing in $r < 0$. Since $g_u(r)$ is strictly increasing in $r > 0$, we get $\hat{r} = 0$ for $u > -\lambda$. Next, consider the case of $u \le -\lambda$. Since $u - r \le -r - \lambda$ so that $1/(1+e^{u-r}) \ge 1/(1+e^{-r-\lambda})$ , $g_u'(r) \ge 0$ and $g_u(r)$ is increasing in $-a\lambda < r < 0$. In $r \le -a\lambda$, the stationary point satisfying $g_u'(r^*) = 0$ is $r^* = au$. Since $r^* \le -a\lambda$ from $u \le -\lambda$, $g_u(r)$ is decreasing in $-\infty < r \le r^*$ and increasing in $r^* \le r \le -a\lambda$. Thus, $\hat{r} = r^* = au$. This completes the proof. ■

**Corollary 2.4:** *The hard thresholding for margin in Theorem* 2.3 *has the below special cases:*

(1) *If* $a \to +\infty$, *then*

$$\lim_{a\to+\infty} p_\lambda(r) = \log(1+e^{\lambda}) - \log(1+e^{-|r|+\lambda}), \quad \hat{r} = \begin{cases} -\infty, & u \le -\lambda, \\ 0, & u > -\lambda. \end{cases}$$

(2) *If* $a \to 1$, *then*

$$\lim_{a\to 1} p_\lambda(r) = \begin{cases} \log(1+e^{\lambda}) - \log(1+e^{-|r|+\lambda}), & 0 < |r| < \lambda \\ \log(1+e^{\lambda}) - \log 2 + \frac{|r|-\lambda}{2}, & |r| \ge -\lambda \end{cases}, \quad \hat{r} = \begin{cases} u, & u \le -\lambda, \\ 0, & u > \lambda. \end{cases}$$

The penalty functions, $p_\lambda(r)$, constructed in the above theorems and corollaries are nonnegative, continuous, nondecreasing in $r > 0$, and symmetric about 0. All types of $p_\lambda(r)$ are singular at $r = 0$, resulting in a thresholding rule that encourages sparsity of the solution [11]. Both thresholdings on margin presented in Theorems 1 and 2 are motivated from soft and hard thresholdings on residual, suggested in Ref. [12]. When $a = 1$, both thresholding rules on margin lead to $u - \hat{r} \le 0$ for all $u$ so that desirable robustness is not expected. When $a = +\infty$, soft and hard thresholdings produce the same $\hat{r}$ and $u - \hat{r} = +\infty$ for an gross negative margin, implying that maximal robustness is expected.

### 2.2. Effective loss and robust properties

The ultimate purpose of simultaneous minimization of (2) over $f$ and $\boldsymbol{\gamma}$ is to get a robust classifier estimate $\hat{f}$ under existence of outliers. Note that $\hat{\gamma}_i$ is a function of $u_i = y_i f(\mathbf{x}_i)$ so that minimization procedure with (2) eventually provides the minimization of

$$R(f) = \sum_{i=1}^{n} g(u_i) \tag{4}$$

with $g(u) = \ell(u - r(u)) + p_\lambda(r(u))$ and $\hat{\gamma} = r(u)$. The function $g(u)$ is called effective loss that is obtained by profiling out $\hat{\gamma}$ [7]. The effective loss $g(u)$ actually plays a role of loss function in the risk minimization procedure for classifier estimation. Thus, performance of the procedure highly depends on the properties of the effective loss $g(u)$ in margin-based classification problem. Before we investigate it, the effective loss should be derived in the association with $p_\lambda(r)$ or, equivalently, $r(u)$ used in the procedure.

The effective loss function can be easily obtained by simply plugging $r(u)$ in the shifted loss with penalty. The specific form of $g(u)$ using thresholding types for margin in the previous theorems and corollaries leads to the following $g(u)$.

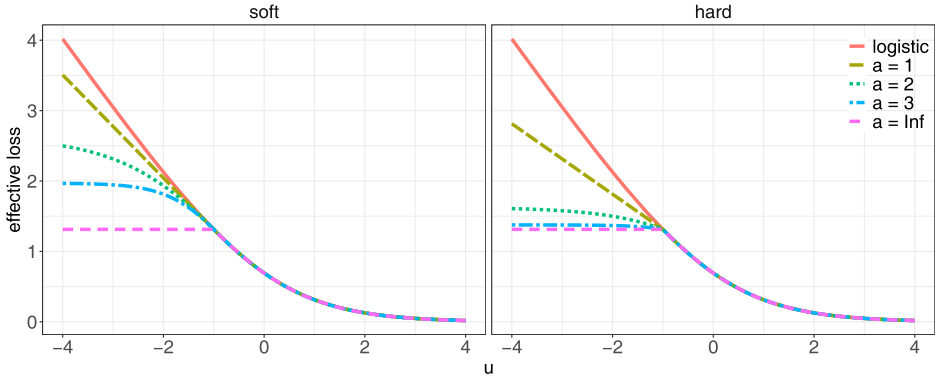


**Figure 1.** Effective loss using soft (left) and hard (right) thresholding with $\lambda = 1$.

- For $u > -\lambda$,

$$g(u) = \ell(u).$$

- For $u \leq -\lambda$,
  ∘$1 < a < +\infty$ case:

$$g(u) = \begin{cases} -\dfrac{1}{a-1}\log(1+e^{-(1-a)u+a\lambda}) + \dfrac{a}{a-1}\log(1+e^{\lambda}) & \text{if soft,} \\ -\dfrac{1}{a-1}\log(1+e^{-(1-a)u}) + \log(1+e^{\lambda}) & \\ \quad +\dfrac{1}{a-1}\log(1+e^{-(a-1)\lambda}) & \text{if hard.} \end{cases}$$

  ∘$a = 1$ case:

$$g(u) = \begin{cases} -\dfrac{e^{\lambda}}{1+e^{\lambda}}(u+\lambda) + \log(1+e^{\lambda}) & \text{if soft,} \\ -\dfrac{1}{2}(u+\lambda) + \log(1+e^{\lambda}) & \text{if hard.} \end{cases}$$

  ∘$a = +\infty$ case:

$$g(u) = \log(1+e^{\lambda}).$$

Functional shapes of effective loss functions are displayed in Figure 1. The effective loss $g(u)$ is continuous on $u \in \mathbb{R}$, regardless of thresholding types and $a$ values. Note that the effective loss becomes $g(u) = \min\{\ell(u), \ell(-\lambda)\}$ with $a = +\infty$ for both soft and hard thresholding cases. This is the truncated logistic loss, studied in Ref. [2]. And $g(u)$ becomes convex and continuously differentiable only for the case of soft thresholding with $a = 1$, which is studied in Refs. [6–8].

$g(u)$ is non-increasing in $u$ so that classification procedure with effective loss promotes correct classification and prevents misclassification as standard logistic regression does. And the classification procedure using $g(u)$ as a loss function satisfies Fisher consistency. This guarantees that Bayes classifier can be estimated under this procedure [1,13].

**Theorem 2.5 (Fisher consistency):** *The classification procedure with a margin-based loss $\ell(u)$ satisfies Fisher consistency if $\ell(u)$ satisfies the below conditions.*

(1) $\ell(u)$ *is non-increasing in* $u$.
(2) $\ell'(0)$ *exists and* $\ell'(0) < 0$.

The proof of Theorem 2.5 is given in Theorem 3.1 of Ref. [1], so omitted here. From the properties of effective loss and Theorem 2.5, the following corollary holds.

**Corollary 2.6:** *The classification procedure with an effective loss* $g(u)$ *satisfies Fisher consistency.*

***Proof:*** Note that $g(u) = \ell(u)$ for $u \geq -\lambda$ with $\lambda > 0$. Therefore, $g'(0)$ exists and $g'(0) = \ell'(0) = -(1/2) < 0$ . Since $g(u)$ is non-increasing, requirements for Fisher consistency in Theorem 2.5 are satisfied. ■

Unlike logistic loss $\ell(u)$ in standard logistic regression, we can see that $g(u)$ is bounded when $a > 1$. Earlier works with $L_1$ penalization [6–8] are equivalent to logistic regression with shift parameters of soft thresholding with $a = 1$, and their approaches may not be robust enough to outliers because its effective loss is unbounded. This implies that standard logistic regression can be further improved in robustness with shift parameter approach using $a > 1$. Robustness property of shift parameter approach can be viewed from weighted logistic regression as well. As we will see in Section 3, classifier $f$ and individual shift parameters $\gamma_i$ are estimated alternately. Suppose $\hat{f}$ and $\hat{\gamma}_i$ are solutions from the previous iteration. The classifier $f$ is updated in the next round by minimizing (2) over $f$ given $\hat{\gamma}_i$, which amounts to the weighted logistic regression problem

$$\min_f \sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i) + \hat{\gamma}_i}) = \min_f \sum_{i=1}^{n} w_i \log(1 + e^{-y_i f(\mathbf{x}_i)})$$

with weights

$$w_i = \frac{\log(1 + e^{-y_i \hat{f}(\mathbf{x}_i) + \hat{\gamma}_i})}{\log(1 + e^{-y_i \hat{f}(\mathbf{x}_i)})}.$$

Note that if $\hat{\gamma}_i = 0$ for all $i = 1, 2, \ldots, n$, then $w_i = 1$ hold. Thus, the procedure with the common weight becomes the standard logistic regression. If $\hat{\gamma}_i < 0$, then $w_i < 1$ so that the observations with $\hat{\gamma}_i < 0$ will have smaller weights in the next iteration step. Since $\hat{\gamma}_i = r(u_i)$ and $r(u)$ is nondecreasing in $u$, severe outliers more likely have small weights. Thus, the classifier estimate becomes robust in the presence of outliers. Moreover, $w_i$ will have the smallest value $w_i = \log 2/\log(1 + e^{-u_i})$ when $r(u_i) = -\infty$. Thus, thresholding with $a = +\infty$ may achieve maximal robustness in shift parameter approach.

## 3. Implementation

The effective loss $g(u)$ in (4) is differentiable for soft thresholding with $1 \leq a < +\infty$, but is not differentiable at $u = -\lambda$ otherwise. Convexity holds only for soft thresholding with $a = 1$. These functional properties introduce computational difficulties for optimizing (4) with $g(u)$ directly. Instead, $g(u, r) = \ell(u - r) + p_\lambda(r)$ relaxes this difficulty because

- $g(u, r)$ is convex in $u$ given $r$, which means that minimizing $\sum_{i=1}^{n} g(y_i f(\mathbf{x}_i), \hat{\gamma}_i)$ over $f$ is a convex problem and
- the minimizer, $\hat{r} = \arg\min_r g(u, r)$, is obtained as a closed-form solution, as in Theorems 1 and 2 when $u = yf(\mathbf{x})$ is given.

These facts motivate us to estimate $f$ and $\boldsymbol{\gamma}$ from (2) in an alternating way, rather to find $f$ from (4) directly. Since $f$ and $\boldsymbol{\gamma}$ are optimized in alternating fashion, one of them is initialized and then alternation can proceed. We initialize $\hat{f}$ obtained from the standard logistic regression procedure without shift parameters and, then, alternation proceeds until convergence is met. This algorithm is summarized in Algorithm 3.

**Algorithm 1.** ( Fitting algorithm for robust logistic regression)

(1) (Initialization) Fit standard logistic regression to obtain $\hat{f}^{(0)}$. Set $\hat{\gamma}_i^{(0)} = 0$. Compute $u_i = y_i \hat{f}^{(0)}(\mathbf{x}_i)$ and set $\lambda_{\max} = 2 \times \max\{-u_i\}$.
(2) Repeat
   (a) Fit $\hat{f}^{(m)}$ using standard logistic regression with an offset variable $-y_i \hat{\gamma}_i^{(m-1)}$.
   (b) Update $\hat{\gamma}_i^{(m)}$ for $i = 1, 2, \ldots, n$ individually as given in Theorems 2.1, 2.3 or Corollaries 2.2, 2.4.
   (c) Stop the loop if convergence is met. Report the current $\hat{f}^{(m)}$ as a final classifier estimate. Otherwise, go to (a) with $m \leftarrow m + 1$.

Learning under (2) with penalty function associated with thresholding on margin presented in Section 2.1 involves two additional parameters $\lambda$ and $a$. Both of them jointly affect the robustness performance of the procedure. Specifically, $\lambda$ determines the number of observations to be adjusted in margin and $a$ determines the amount of adjustment influencing on how far their margins are pushed toward infinity. Smaller $\lambda$ and larger $a$ enhance robustness but reduce efficiency due to potential decrease in available normal observations. Thus, an appropriate choice of them in balance between robustness and efficiency is crucial in prediction perspective. These parameters are outside the learning process and, thus, must be specified in advance before learning starts. To do this, we first set $a$ from a candidate set $\mathcal{A}$ and find $\lambda$ to show the best performance in prediction through cross validation.

Cross validation for choosing $\lambda$ requires a grid set of $\lambda$. Note that, from Theorems 1 and 2, $\hat{\gamma}_i = 0$ holds if $\lambda > -u_i$ for all $i = 1, \ldots, n$. This implies that $\lambda$ larger than $\max\{-u_i\}$ is not necessary for performance comparison. Thus, using the margins $u_i$ from the initial fit $\hat{f}$, we make a grid set on the range $0 < \lambda \leq \lambda_{\max}$ by setting $\lambda_{\max} = 2 \times \max\{-u_i\}$, since margins keep being re-evaluated at every iteration so that their magnitude keeps changing. For each $a \in \mathcal{A}$, let $\lambda_a$ be the optimal $\lambda$ that generates the best predictive performance by achieving the smallest validation error rate, say $\mathrm{CV}(\lambda_a)$. Then we choose $a^{opt} = \arg\min_a \{\mathrm{CV}(\lambda_a) : a \in \mathcal{A}\}$ and $\lambda^{opt} = \lambda_{a^{opt}}$. The final solution is obtained by applying robust logistic regression to the whole data with these optimal parameters $a = a^{opt}$ and $\lambda = \lambda^{opt}$.

Let us show the convergence analysis of Algorithm 1, where $R(f, \boldsymbol{\gamma}) = \sum_{i=1}^{n} g(u_i, \gamma_i)$ with $u_i = y_i f(\mathbf{x}_i)$ and $g(u, r) = \ell(u - r) + p_\lambda(r)$ is minimized alternately over $f$ and $\boldsymbol{\gamma}$. Note that $g(u, r)$ becomes a majorizing function of $g(u)$ at $r = r(u)$, which means that $g(u, r) \geq g(u)$ for all $r$ and the equality $g(u, r(u)) = g(u)$ holds only when $r = r(u)$. This MM (Majorization-minimization) property [14,15] ensures that the updated $\hat{f}$ keeps lowering $R(f)$ in (4) at every iteration and, finally, converges to a local minimum of $R(f)$.

**Theorem 3.1:** *Let $\hat{f}^{(m)}$ be the estimate of $f$ at the mth step under Algorithm* 3. *The sequence of $\{\hat{f}^{(m)} : m = 1, 2, \ldots\}$ converges to a local minimum of the objective function $R(f) = \sum_{i=1}^{n} g(y_i f(\mathbf{x}_i))$.*

***Proof:*** Denote $\hat{\gamma}_i^{(m)} = r(y_i \hat{f}^{(m)}(\mathbf{x}_i))$ with the margin thresholding $r(u)$ associated with $p_\lambda(r)$. Observe that

$$\begin{aligned} R(\hat{f}^{(m)}) &= \sum_{i=1}^{n} g(y_i \hat{f}^{(m)}(\mathbf{x}_i)) \;=\; \sum_{i=1}^{n} g(y_i \hat{f}^{(m)}(\mathbf{x}_i), \hat{\gamma}_i^{(m)}) \\ &\geq \sum_{i=1}^{n} g(y_i \hat{f}^{(m+1)}(\mathbf{x}_i), \hat{\gamma}_i^{(m)}) \;\geq\; \sum_{i=1}^{n} g(y_i \hat{f}^{(m+1)}(\mathbf{x}_i), \hat{\gamma}_i^{(m+1)}) \\ &= \sum_{i=1}^{n} g(y_i \hat{f}^{(m+1)}(\mathbf{x}_i)) \;=\; R(\hat{f}^{(m+1)}). \end{aligned}$$

The first inequality holds since $\hat{f}^{(m+1)}$ is the solution of logistic regression, which minimizes the negative log likelihood given the offset $-y_i \hat{\gamma}_i^{(m)}$. The second inequality comes from the fact that thresholding on the current margin, $\hat{\gamma}_i^{(m+1)} = r(y_i \hat{f}^{(m+1)}(\mathbf{x}_i))$, is the unique minimizer of $g(y_i \hat{f}^{(m+1)}(\mathbf{x}_i), r)$ for each $i = 1, \ldots, n$. Since $R(f)$ is bounded below and keeps decreasing at every iteration step, this algorithm eventually stops at a local minimum. ■

## 4. Numerical studies

### 4.1. Simulation 1: linear case

Two-class data were generated from normal distribution $\mathbf{x} \sim N_p((\pm c_\alpha/2)\mathbf{1}_p, \mathbf{I}_p)$ with a positive constant $c_\alpha$. Class label $y = \pm 1$ is taken according to its mean. The common variance is assumed since we consider the linear classification in this scenario. The distance between means of two classes is $c_\alpha \sqrt{p}$. The positive constant $c_\alpha$ is defined as $c_\alpha = 2z_\alpha/\sqrt{p}$ with the upper standard normal quantile $z_\alpha$. Bayes decision boundary is $f^\star(\mathbf{x}) = x_1 + \cdots + x_p = 0$. This $c_\alpha$ is tailored to have Bayes error rate of $\alpha$. We assume the linear classifier model $f(\mathbf{x}) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta}$ for linear classification.

We generate $n/2$ observations for each class and this dataset free from outliers is considered as a 'clean' data. In this simulation, we consider two scenarios for noise generation:

(S1) Noise in $y$: we randomly select 0%, 5%, 10%, 15% of observations from $-1$ class, and flip their class label over by assigning $+1$.

**Table 1.** Noise scenario (S1) for linear classification: average of test misclassification rates from 100 repetitions is presented. 'true' and 'noisy' are for logistic regression with true labels and flipped labels, respectively. '$\text{soft}_a$' and '$\text{hard}_a$' are robust logistic regression with soft and hard thresholding adjustment. $a$ is a multiplicative factor used. $a =$ 'cv' means $a$ is chosen by cross-validation procedure.

| $n$ | $p$ | rate | true | noisy | $\text{soft}_1$ | $\text{soft}_2$ | $\text{soft}_{\text{inf}}$ | $\text{soft}_{\text{cv}}$ | $\text{hard}_1$ | $\text{hard}_2$ | $\text{hard}_{\text{inf}}$ | $\text{hard}_{\text{cv}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 200 | 2 | 0 | 0.102 | 0.102 | 0.102 | 0.102 | 0.102 | 0.102 | 0.102 | 0.102 | 0.102 | 0.102 |
| | | 0.05 | 0.102 | 0.111 | 0.111 | 0.109 | 0.106 | 0.107 | 0.110 | 0.108 | 0.106 | 0.107 |
| | | 0.10 | 0.102 | 0.143 | 0.141 | 0.133 | 0.121 | 0.123 | 0.138 | 0.129 | 0.121 | 0.123 |
| | | 0.15 | 0.102 | 0.199 | 0.195 | 0.184 | 0.157 | 0.159 | 0.190 | 0.174 | 0.158 | 0.158 |
| | 10 | 0 | 0.108 | 0.108 | 0.108 | 0.108 | 0.108 | 0.108 | 0.108 | 0.108 | 0.109 | 0.108 |
| | | 0.05 | 0.108 | 0.121 | 0.121 | 0.119 | 0.115 | 0.115 | 0.120 | 0.117 | 0.115 | 0.115 |
| | | 010 | 0.108 | 0.155 | 0.152 | 0.146 | 0.135 | 0.136 | 0.150 | 0.141 | 0.134 | 0.136 |
| | | 0.15 | 0.108 | 0.216 | 0.213 | 0.208 | 0.193 | 0.194 | 0.211 | 0.203 | 0.198 | 0.199 |
| | 20 | 0 | 0.116 | 0.116 | 0.116 | 0.116 | 0.117 | 0.117 | 0.116 | 0.116 | 0.117 | 0.117 |
| | | 0.05 | 0.116 | 0.132 | 0.131 | 0.128 | 0.124 | 0.124 | 0.130 | 0.126 | 0.123 | 0.124 |
| | | 0.10 | 0.116 | 0.167 | 0.165 | 0.159 | 0.153 | 0.152 | 0.162 | 0.154 | 0.150 | 0.150 |
| | | 0.20 | 0.116 | 0.229 | 0.227 | 0.223 | 0.219 | 0.219 | 0.227 | 0.224 | 0.216 | 0.220 |
| 400 | 2 | 0 | 0.101 | 0.101 | 0.101 | 0.101 | 0.101 | 0.101 | 0.101 | 0.101 | 0.101 | 0.101 |
| | | 0.05 | 0.101 | 0.110 | 0.110 | 0.108 | 0.105 | 0.106 | 0.109 | 0.107 | 0.105 | 0.105 |
| | | 0.10 | 0.101 | 0.142 | 0.139 | 0.131 | 0.118 | 0.119 | 0.136 | 0.128 | 0.119 | 0.122 |
| | | 0.15 | 0.101 | 0.199 | 0.193 | 0.181 | 0.151 | 0.154 | 0.189 | 0.172 | 0.150 | 0.151 |
| | 10 | 0 | 0.104 | 0.104 | 0.104 | 0.104 | 0.104 | 0.104 | 0.104 | 0.104 | 0.105 | 0.104 |
| | | 0.05 | 0.104 | 0.115 | 0.115 | 0.113 | 0.110 | 0.110 | 0.114 | 0.112 | 0.110 | 0.110 |
| | | 010 | 0.104 | 0.146 | 0.143 | 0.137 | 0.124 | 0.125 | 0.142 | 0.134 | 0.125 | 0.126 |
| | | 0.15 | 0.104 | 0.202 | 0.197 | 0.187 | 0.165 | 0.167 | 0.193 | 0.181 | 0.163 | 0.164 |
| | 20 | 0 | 0.108 | 0.108 | 0.108 | 0.108 | 0.109 | 0.108 | 0.108 | 0.108 | 0.109 | 0.108 |
| | | 0.05 | 0.108 | 0.121 | 0.120 | 0.118 | 0.114 | 0.114 | 0.119 | 0.116 | 0.115 | 0.115 |
| | | 0.10 | 0.108 | 0.153 | 0.150 | 0.144 | 0.131 | 0.132 | 0.149 | 0.140 | 0.132 | 0.133 |
| | | 0.20 | 0.108 | 0.211 | 0.208 | 0.201 | 0.186 | 0.187 | 0.206 | 0.197 | 0.187 | 0.188 |

(S2) Noise in $\mathbf{x}$: we randomly select 10% of observations from $-1$ class, and replace their by an outlying observation from $N_p(m \times (+c_\alpha/2)\mathbf{1}_p, \mathbf{I}_p)$ with $m = 2$ (mild outliers) and $m = 5$ (severe outliers).

In (S1), a portion of observations from negative class has flipped class labels so that observations in positive class of training data are not random samples from population. This kind of noise is called label noise [16]. In (S2), some observations in negative class have $\mathbf{x}$ outlying from their class toward positive class. For both noise scenarios, we consider asymmetric noise generation, in which noise in $y$ and $\mathbf{x}$ intervenes the negative class only, because symmetric noise does not seriously affect classification performance [6,16]. It is known well that outliers located inside their class have little effect on classification procedure as their margin typically becomes large positive and, thus, their loss takes lower value with a nondecreasing loss function. With this reason, (S2) considers outliers located in the opposite class.

For performance demonstration, we generated training data of size $n$ with Bayes error $\alpha = 0.1$ and made a noisy dataset from the clean training data under noise-generating schemes (S1) and (S2). These noisy data are used to estimate $f$ in robust logistic regression with soft and hard thresholding on margin with $a \in \mathcal{A} = \{1, 2, 3, 4, 5, \infty\}$. As a benchmark, logistic regression with true label (clean data) and noisy label (noisy data) are performed as well. Test data of size 10, 000 and free from noise are used for test misclassification rate. We repeated it 100 times and report the average of 100 test misclassification rates in Tables 1 and 2.

**Table 2.** Noise scenario (S2) for linear classification: average of test misclassification rates from 100 repetitions is presented. 'true' and 'noisy' are for logistic regression with true labels and flipped labels, respectively. '$\text{soft}_a$' and '$\text{hard}_a$' are robust logistic regression with soft and hard thresholding adjustment. $a$ is a multiplicative factor used. $a =$ 'cv' means $a$ is chosen by cross-validation procedure.

| $n$ | $p$ | $m$ | true | noisy | $\text{soft}_1$ | $\text{soft}_2$ | $\text{soft}_{\text{inf}}$ | $\text{soft}_{\text{cv}}$ | $\text{hard}_1$ | $\text{hard}_2$ | $\text{hard}_{\text{inf}}$ | $\text{hard}_{\text{cv}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 200 | 2 | 2 | 0.102 | 0.122 | 0.114 | 0.106 | 0.105 | 0.106 | 0.111 | 0.105 | 0.105 | 0.106 |
| | | 5 | 0.102 | 0.170 | 0.118 | 0.114 | 0.116 | 0.118 | 0.110 | 0.122 | 0.116 | 0.118 |
| | 10 | 2 | 0.108 | 0.140 | 0.128 | 0.112 | 0.109 | 0.111 | 0.122 | 0.111 | 0.110 | 0.110 |
| | | 5 | 0.108 | 0.209 | 0.152 | 0.129 | 0.141 | 0.137 | 0.140 | 0.127 | 0.133 | 0.132 |
| | 20 | 2 | 0.116 | 0.156 | 0.142 | 0.121 | 0.118 | 0.120 | 0.137 | 0.119 | 0.117 | 0.119 |
| | | 5 | 0.116 | 0.235 | 0.186 | 0.140 | 0.158 | 0.155 | 0.169 | 0.140 | 0.152 | 0.144 |
| 400 | 2 | 2 | 0.101 | 0.120 | 0.112 | 0.103 | 0.102 | 0.104 | 0.109 | 0.105 | 0.102 | 0.104 |
| | | 5 | 0.101 | 0.167 | 0.113 | 0.110 | 0.106 | 0.107 | 0.108 | 0.111 | 0.106 | 0.109 |
| | 10 | 2 | 0.104 | 0.132 | 0.120 | 0.108 | 0.106 | 0.106 | 0.115 | 0.108 | 0.107 | 0.107 |
| | | 5 | 0.104 | 0.197 | 0.133 | 0.124 | 0.119 | 0.119 | 0.124 | 0.122 | 0.116 | 0.121 |
| | 20 | 2 | 0.108 | 0.140 | 0.127 | 0.111 | 0.109 | 0.110 | 0.122 | 0.113 | 0.109 | 0.110 |
| | | 5 | 0.108 | 0.209 | 0.148 | 0.125 | 0.125 | 0.124 | 0.138 | 0.127 | 0.126 | 0.129 |

Table 1 presents the results under (S1). Logistic regression with clean data ('true') shows test misclassification rate close to Bayes error $\alpha = 0.1$. However, its performance with noisy data ('noisy') quickly deteriorates as label noise rate ('rate') increases. Applying shift parameter approach with soft and hard thresholding on margin to logistic regression improves prediction performance when $a > 1$. With $a = 1$, misclassification error rates are little improved as we expect, since the corresponding effective loss is unbounded. Comparing to $a = 1$, any choice of $a > 1$ is preferred. The choice of $a = \infty$ shows the almost best performance over $\mathcal{A}$ in Table 1 (we do not report $a = 3, 4, 5$ here due to space limit). Moreover, data-driven choice of $a$ by cross validation performs similarly as $a = \infty$. Thus, in the case that label noise is suspected, we suggest to use $a = \infty$ and find $\lambda$ by cross validation. It is more efficient in computation than two-dimensional grid search for selecting both $a$ and $\lambda$. We observe that two types of thresholding, soft and hard, do not show significant difference in prediction performance, while hard thresholding seems a bit better than soft thresholding when $a$ is small. This difference can be easily overcome by selecting $a$ wisely.

The results in Table 1 may not be satisfactory enough because amount of improvement seems marginal comparing to 'noisy.' Note that label noise comes from flipping their labels, so that noisy observations are not far from their own class in $\mathbf{x}$ space. In the case that two classes are overlapped substantially, flipped observations are indistinguishable with non-flipped ones from the opposite class so that they are not clearly distinct as outliers. This is demonstrated in the results in Table 1.

Different from (S1), the scenario (S2) generates distinct outliers by locating a portion of observations toward the opposite side, far away from their own class. The larger the factor $m$ is, the farther the outliers are located away from their class. Table 2 presents the results under (S2) with 10% noise rate. Note that in the case of $m = 5$, logistic regression with noisy data provides prediction performance worse than that with the same rate (10%) of label noise in (S1). Table 2 shows that robust logistic regression under shift parameter approach considerably improves the naive use of logistic regression with noisy data. We observe that contrast to (S1), thresholding with any $a > 1$ performs comparable to that with the choice from cross validation. This indicates that moderate size of $a$ can effectively remove the effect of severe outliers, while moderate outliers, e.g. label noise in (S1), require a large $a$ to lessen their effect on the performance.

### 4.2. Nonlinear case

To obtain flexible nonlinear decision boundary, we consider a classifier model $f(\mathbf{x}) = \beta_0 + h(\mathbf{x}) \in \overline{\mathcal{H}}_K = \mathbb{R} + \mathcal{H}_K$, where $\mathcal{H}_K$ is a reproducing kernel Hilbert space induced by a kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Thus, in alternating fitting procedure, $f$ is estimated by

$$\hat{f} = \underset{f \in \overline{\mathcal{H}}_K}{\arg\min} \sum_{i=1}^{n} \ell(y_i f(\mathbf{x}_i) - \hat{\gamma}_i) + \kappa \|h\|^2_{\mathcal{H}_K}. \tag{5}$$

By the representer theorem and the reproducing property of $\mathcal{H}_K$, the infinite-dimensional optimization problem of (5) becomes a finite dimensional problem where $\hat{f}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$ and $\|h\|^2_{\mathcal{H}_K} = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$, resulting in

$$\hat{f}(\mathbf{x}) = \underset{\beta_0 \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}^n}{\arg\min} \ \ell(y_i(\beta_0 + \alpha_i K(\mathbf{x}, \mathbf{x}_i)) - \hat{\gamma}_i) + \kappa \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \tag{6}$$

with a matrix $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\ldots,n}$. The optimization problem (6) can be performed under penalized logistic regression with the model matrix $\mathbf{K}$ and the offset $-y_i\hat{\gamma}_i$. The detailed properties and implementations related to kernel method are referred to Refs. [17,18].

Regarding the flexible nonlinear learning under the presence of outliers, there is a potential concern that outlying patterns may not be distinguished from main features underlying the data-generating process. This implies that highly flexible nonlinear model may learn outlying patterns in learning process and, thus, cannot be generalized well to new observations. Shift parameter approach is applied to nonlinear classification by learning $f(\mathbf{x}_i) - y_i\gamma_i$. We expect that outlying patterns are absorbed into $\hat{\gamma}_i$, and well separated from $\hat{f}$.

For nonlinear classification, we set $\mathbf{x}_i = (x_{i1}, x_{i2})^T$ where $x_{ij}$ are independently sampled from uniform distribution over $(-1, 1)$. And we set their labels by $y_i = 2I(x_{i1}x_{i2} \geq 0) - 1$. To introduce noise in the data, we randomly choose 5%, 10%, 15% of data from class of $y_i = 1$ and relocate a half of them into the upper-left quadrant and the remaining half into the lower-right quadrant. Figure 2 shows an example of size $n = 200$ and 10% outliers. Radial-basis gaussian kernel $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \ ^2/2\sigma^2)$ is used for nonlinear classifier, where $\sigma$ is tuned among $\{0.1, \ldots, 0.5\}$ quantiles of pairwise distances of training data. In Figure 2, both decision boundaries from support vector machines (SVMs) and logistic regression with noisy training data contain islands around outlying observations, implying mis-located observations heavily influence on classification results. Contrast to them, such islands disappear in shift parameter approaches. In panels of shift parameter approach, nonzero estimates of $\gamma_i$ are highlighted by filled points. This demonstrates that individual shift parameter is able to separate outlier's effect from classifier estimation. To evaluate and compare their prediction performance, we repeated it 100 times and obtained test misclassification rates, whose averages are provided in Table 3. When the proportion of outliers is small (5% rate), we observed that SVM and naive logistic regression estimate Bayes decision boundary well without islands around outliers and perform even better than shift parameter approaches in prediction. However, as the number of outliers increases, they often create holes near the area with many outliers, making wrong decision to test prediction. With moderate rate of outliers (10%), shift parameter approaches outperform SVM

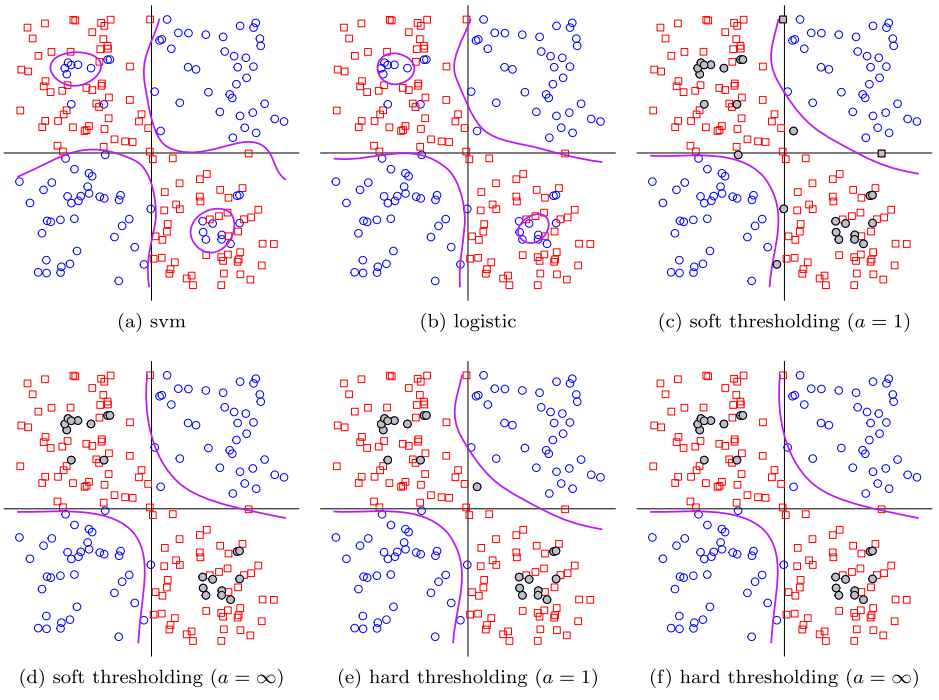


**Figure 2.** An example of 10% noise rate is presented. Solid curve represents decision boundary estimated from each classification procedure. Filled points are the observations of nonzero shift parameter estimate.

**Table 3.** Nonlinear classification: average of test misclassification rates from 100 repetitions is presented. 'svm' and 'noisy' are for support vector machine and logistic regression, respectively, with flipped labels. '$\text{soft}_a$' and '$\text{hard}_a$' are robust logistic regression with soft and hard thresholding adjustment. $a$ is a multiplicative factor used. $a =$ 'cv' means $a$ is chosen by cross-validation procedure.

| $n$ | rate | svm | noisy | $\text{soft}_1$ | $\text{soft}_2$ | $\text{soft}_{\text{inf}}$ | $\text{soft}_{\text{cv}}$ | $\text{hard}_1$ | $\text{hard}_2$ | $\text{hard}_{\text{inf}}$ | $\text{hard}_{\text{cv}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 200 | 5% | 0.052 | 0.052 | 0.054 | 0.053 | 0.067 | 0.066 | 0.054 | 0.053 | 0.067 | 0.066 |
| | 10% | 0.082 | 0.080 | 0.069 | 0.067 | 0.072 | 0.073 | 0.069 | 0.067 | 0.072 | 0.073 |
| | 15% | 0.111 | 0.098 | 0.092 | 0.090 | 0.084 | 0.088 | 0.092 | 0.09 | 0.084 | 0.087 |
| 400 | 5% | 0.039 | 0.040 | 0.037 | 0.037 | 0.046 | 0.045 | 0.037 | 0.037 | 0.046 | 0.045 |
| | 10% | 0.070 | 0.064 | 0.048 | 0.047 | 0.053 | 0.053 | 0.048 | 0.047 | 0.053 | 0.053 |
| | 15% | 0.095 | 0.081 | 0.072 | 0.070 | 0.061 | 0.063 | 0.071 | 0.070 | 0.061 | 0.063 |

and naive logistic regression. But, we observed that when there are sizable amount of outliers in training set, shift parameter approaches tend to learn outlying pattern as well. So their prediction performance deteriorates quickly as the rate of outliers increases, implying that outliers are not ignorable in learning process.

### 4.3. Real examples

We applied the shift parameter approach to some real-world datasets having two-class labels from UCI Machine Learning Repository [19]. Three datasets (`ionosphere`, `parkinsons` and `spambase`) are used for linear classification, and two datasets

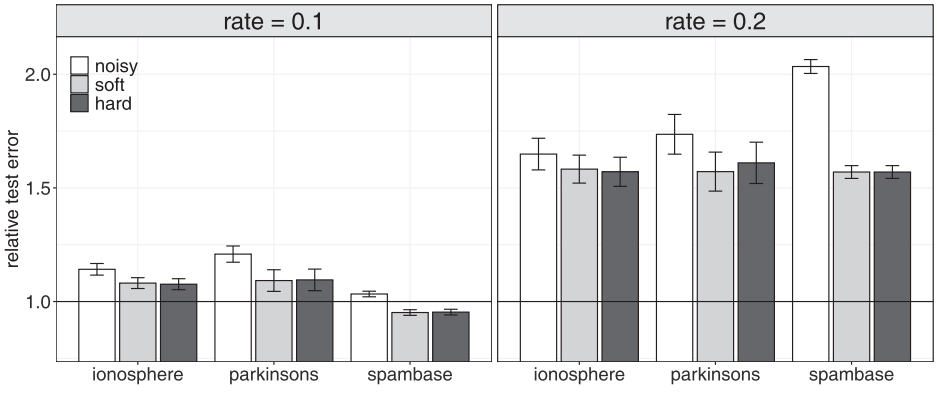


**Figure 3.** Real data analysis: linear classification.

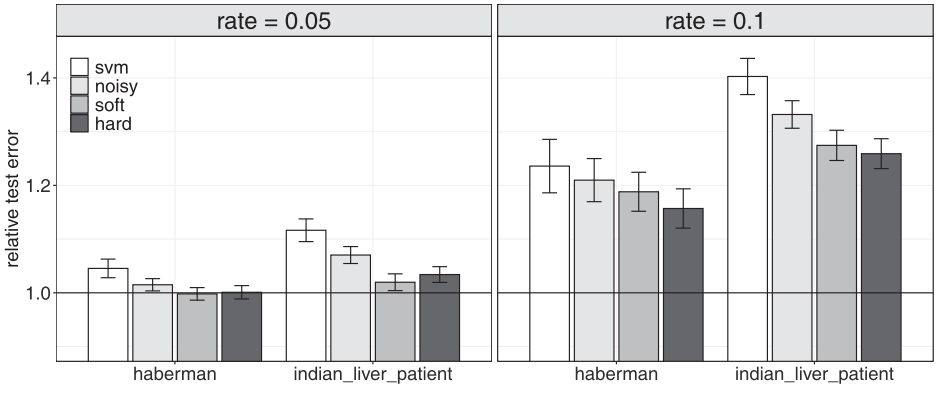


**Figure 4.** Real data analysis: nonlinear classification.

(`haberman` and `indian liver patient`) are for nonlinear classification. These datasets are assumed correctly labelled since we do not know their real mechanism on label noise and there is no reason to doubt the existence of label noise in datasets. For each dataset, we randomly split it into 70% and 30% to use training dataset and test dataset, respectively. The training data are intentionally corrupted in the way that a portion of observations is randomly chosen from the major class and, then, their labels are flipped over to the minor class. After fitting the methods to the noisy training dataset, misclassification rate for clean test data was computed and used for comparison. Since the datasets considered here have different scale of misclassification rate, we provide the relative misclassification rate with respect to logistic regression from the clean training data without label noise. To reduce the effect on prediction performance from random train/test splitting, we repeated the procedure 50 times and provide averaged relative test errors with an errorbar of $\pm$SE in Figures 3 and 4.

Figure 3 shows the relative test misclassification rate of linear classification using training data being added by 10% and 20% label noise. Misclassification rate increases as noise rate increases. There is no significant difference between soft and hard thresholdings for shift parameter, with $a$ chosen by cross validation, as we observed in artificial

data examples. However, they clearly outperform naive logistic regression. For nonlinear classification, we used radial-basis gaussian kernel as in Section 4.2 and present relative misclassification rates in Figure 4 including SVM applied to noisy training data having 5% and 10% noise rate. Similar results are found as linear classification. Shift parameter approaches provide smaller relative error than SVM and logistic regression applied naively to noisy labels. From our experience on linear and nonlinear classification to some real datasets, shift parameter approaches applied to noisy labelled data often show even better performance than standard logistic regression to clean data without adding noise. `spambase` and `haberman` are such a case, shown in Figures 3 and 4, where their relative error rates for both shift parameter approaches relative to standard logistic regression are smaller than 1 ('soft' and 'hard' in figures). It may be, we guess, because such datasets may not be free from noise in label/inputs and our robust methods improve prediction by adjusting margins not only for intentionally added noisy labels but also for unknown outliers that may exist in the data.

## 5. Conclusion

From this work, we investigate the shift parameter approach for robust logistic regression. While the existing methods translate margin to 0, shift parameter moves margin toward the minimum of logistic loss, $+\infty$, in our proposal. This makes logistic regression more robust. And we discuss its robustness using effective loss properties. In this work, we propose soft and hard thresholding on margins for shift parameter and devise penalty functions associated with such thresholding estimates. A simple algorithm is developed under alternating fashion. Numerical studies using synthetic and real examples demonstrate that the shift parameter approach with soft and hard thresholding for margin perform similar, but outperform naive logistic regression and support vector machines in classification with noisy training data.

Shift parameter approach is motivated from wavelet denoising and applied to robust regression. Then, this idea is extended to logistic regression, but the shape of shift parameter estimate suitable for robust logistic regression should be different from that for robust regression, as we discussed in this paper. Robust estimation using the shift parameter approach may be further extended to robust classification having any margin-based loss function, for example, hinge loss for SVMs or exponential loss for boosting, etc. Since penalty function associated with soft and hard thresholding estimate depends on loss function used in classification procedure, it is worth studying robust classification for general margin-based loss under shift parameter estimation. In addition, as we mentioned, it is not an easy task to learn a generalizable classifier from noisy data because noise pattern can be easily absorbed into the flexible model. This can be understood as a kind of overfitting phenomenon to the data, which is an important issue to be resolved in learning. These are our research directions in near future and this study will serve as a foundation for them.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2C1003956).

## References

[1] Lin Y. A note on margin-based loss functions in classification. Stat Probab Lett. 2004;68:73–82.
[2] Park SY, Liu Y. Robust penalized logistic regression with truncated loss function. Can J Stat. 2011;39:300–323.
[3] Wu Y, Liu Y. Robust truncated hinge loss support vector machines. J Am Stat Assoc. 2007;102:974–983.
[4] Bootkrajang J. A generalised label noise model for classification in the presence of annotation errors. Neurocomputing. 2016;192:61–71.
[5] Jung H, Lee S. Individual transition label noise logistic regression in binary classification for incorrectly labeled data. Technometrics. 2022;64:18–29.
[6] Lee S, Shin H, Lee SH. Label-noise resistant logistic regression for functional data classification with an application to Alzheimer's disease. Biometrics. 2016;72:1325–1335.
[7] Lee Y, MacEchern SN, Jung Y. Regularization of case-specific parameters for robustness and efficiency. Stat Sci. 2012;27:350–372.
[8] Tibshirani J, Manning DC. Robust logistic regression using shift parameters. In: Proceeding of the 52nd Annual Meeting of the Association for Computational Linguistics (Vol. 2); 2014. p. 124–129; Baltimore, Maryland.
[9] Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc. 2001;96:1348–1360.
[10] Zhang C-H. Nearly unbiased variable selection under minimax concave penalty. Ann Stat. 2010;38:894–942.
[11] Antoniadis A. Wavelet methods in statistics: some recent developments and their applications. Stat Surv. 2007;1:16–55.
[12] She Y, Owen AB. Outlier detection using nonconvex penalized regression. J Am Stat Assoc. 2011;106:626–639.
[13] Bartlett PL, Jordan MI, McAuliffe JD. Convexity, classification, and risk bounds. J Am Stat Assoc. 2006;101:138–156.
[14] Hunter DR, Lange K. A tutorial on MM algorithms. Am Stat. 2004;58:30–37.
[15] Lange K, Hunter DR, Yang I. Optimization transfer using surrogate objective function (with discussion). J Comput Graph Stat. 2000;9:1–20.
[16] Frénay B, Verleysen M. Classification in the presence of label noise: a survey. IEEE Trans Neural Netw Learn Syst. 2014;25:845–869.
[17] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2nd ed. New York: Springer; 2009.
[18] Schölkopf B, Smola AJ. Learning with kernels. Cambridge: The MIT Press; 2002.
[19] Dua D, Graff C. UCI Machine Learning Repository. Irvine, CA: School of Information and Computer Science, University of California; 2019. Available from: http://archive.ics.uci.edu/ml